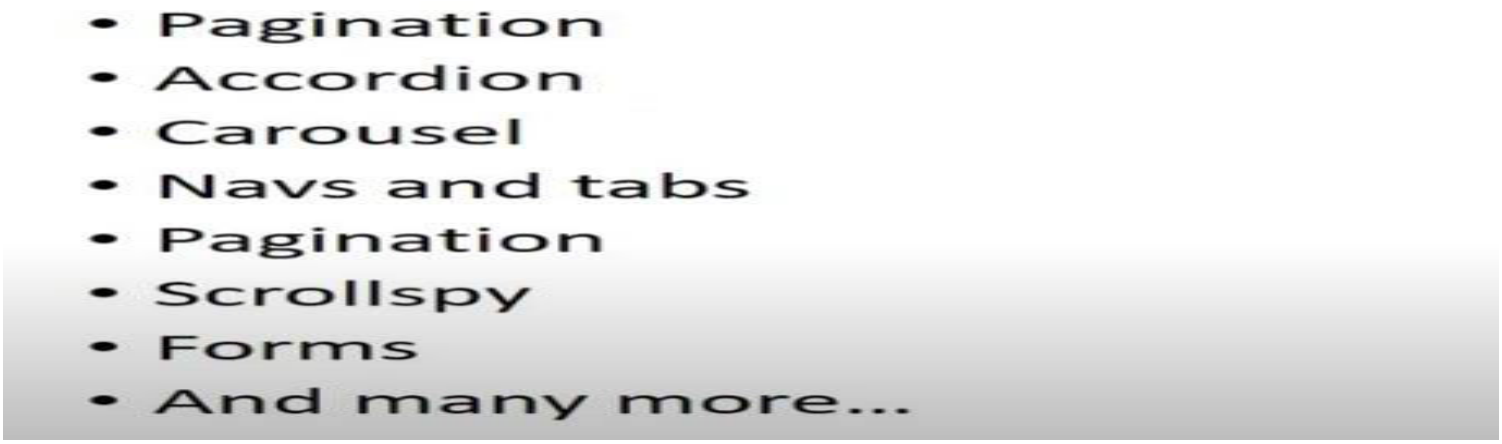
BOOTSTRAP 5

For notes refer to this link

Introduction • Bootstrap v5.0 (getbootstrap.com)
Or go the getbootstrap.com and then docs,
We have different things.
It is a font end framework in which html , css and javaScript used to create framework to the work easier of the developer so that we will not have repeat the same thing. It provides gri system, different classes and components for different work like to make the image responsive we class="img-fluid" . It is used to make the website responsive. We do have other frameworks but it is very famous.
Componets like button to create button.
We have different componts like buttons, alerts etc as given below



To use the bootstrape, we have copied the syntax and pasted in bootstrap folder,

We need to go the get started.com , docs, intoroduction to the syntax

To wrap the content to remove the horzontal line in vs code we will go to view and the select word wrap and alt+z

class=”container” will give margin from left and right and at mobile screen it will 100% width

class=”container-fluid” will give 100% width
class=”bg-danger text-white” to color the text white of inner div text and bg-danger to color background color red

Bootstrap Flow[First container then row then column then content]
container>row>column>content(text and image)

```
<div class="container">
    <div class="row">
      <div class="column">
        <p>Hello</p>
      </div>
    </div>
  </div>
```

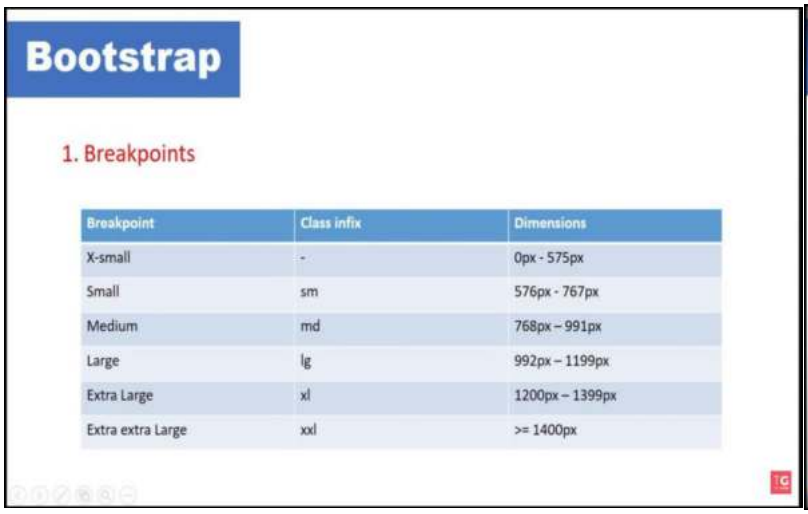
Class col-auto will take width according to the content width
Column-offset works in two columns to give space from left offset-1 means one column space form left
class text-center to center the text and text-end to align at the end

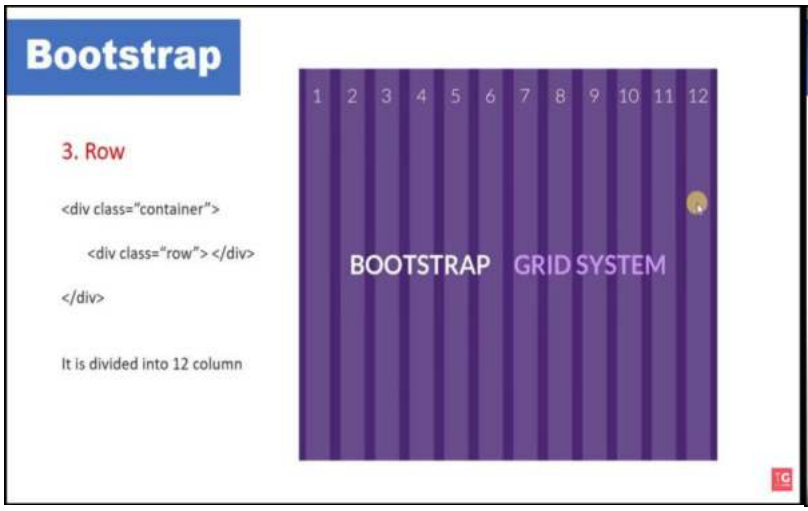
<img src="..." class="img-fluid" alt="...">

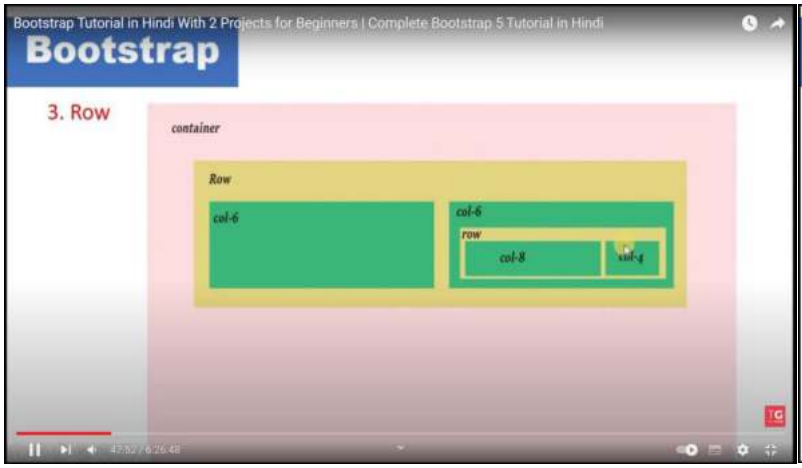
```
<img src=""  class="img-fluid"  alt="" >     to make the image responsive// we use rounded to make the border radius round
```

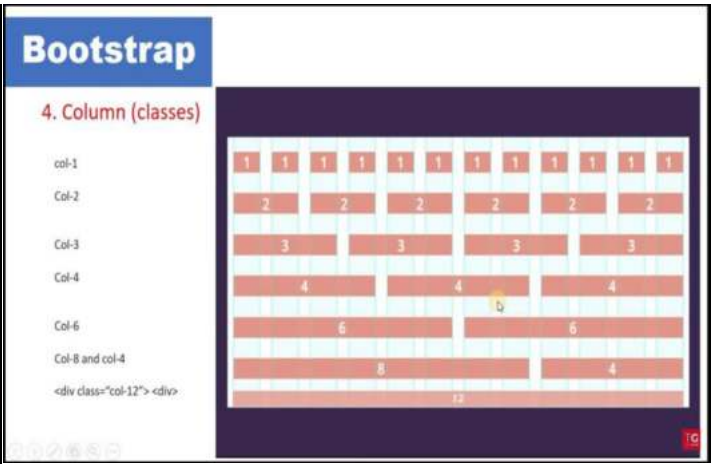
```
<img src=""  class="rounded"  alt="" >   // we use rounded to make the border radius round
```

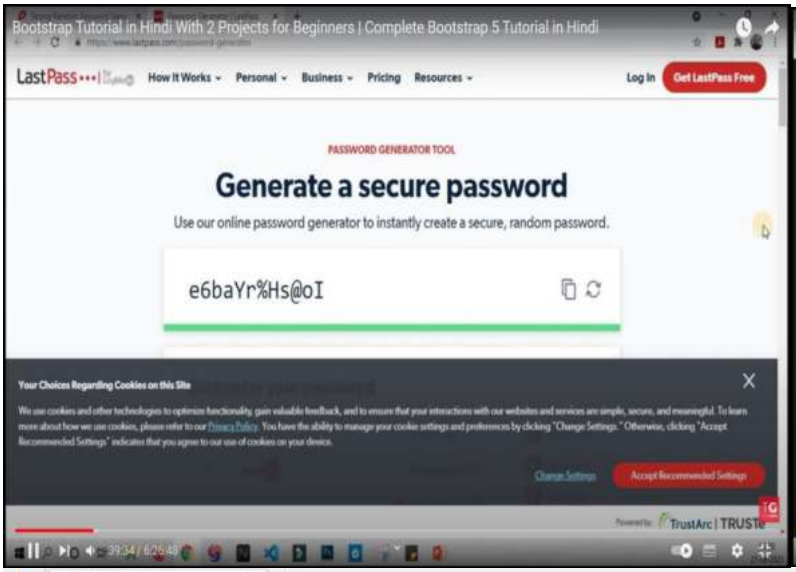
Content>table to style the table in bootstrap
We use class=”table” to use the table styling
If we have more columns and want to responsive that table then need to put the table inside a div container and Give it a class table-responsive

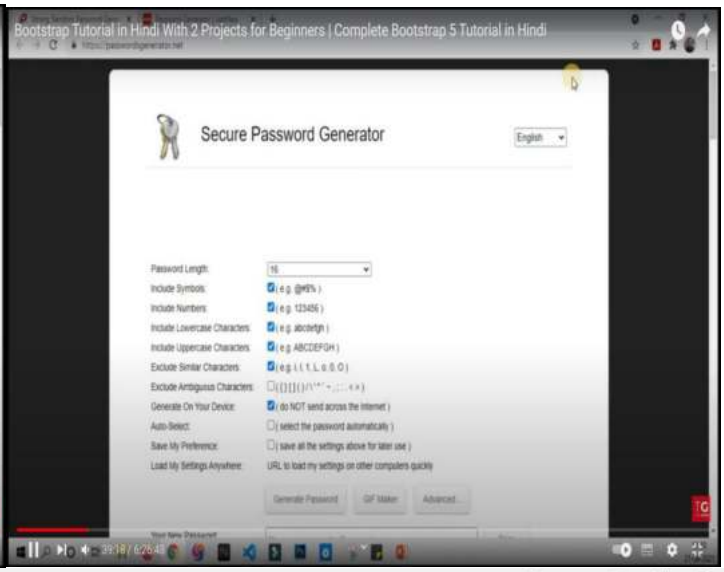


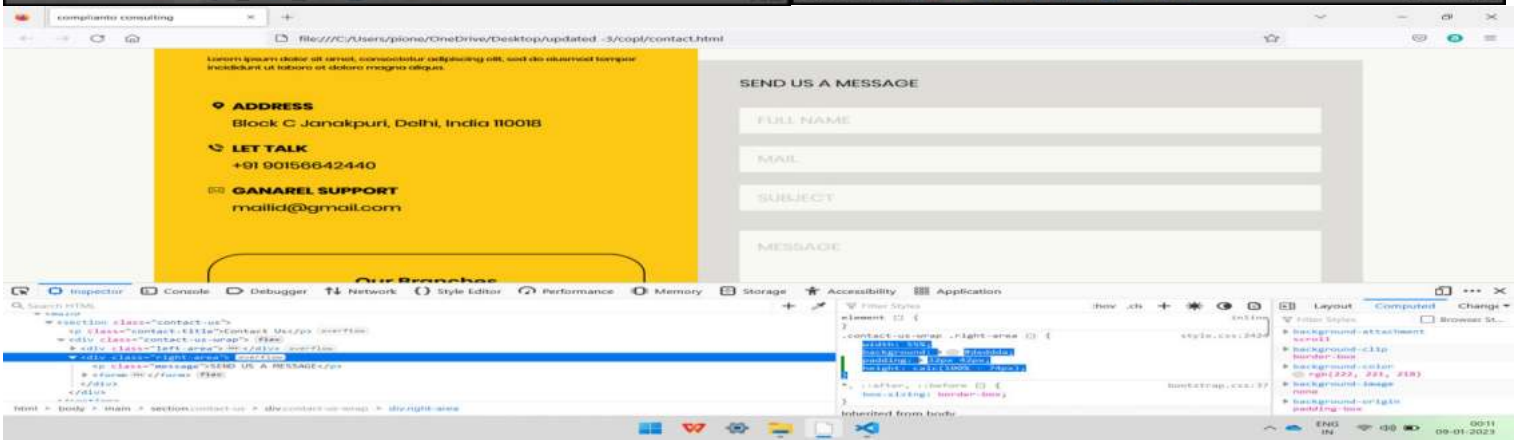


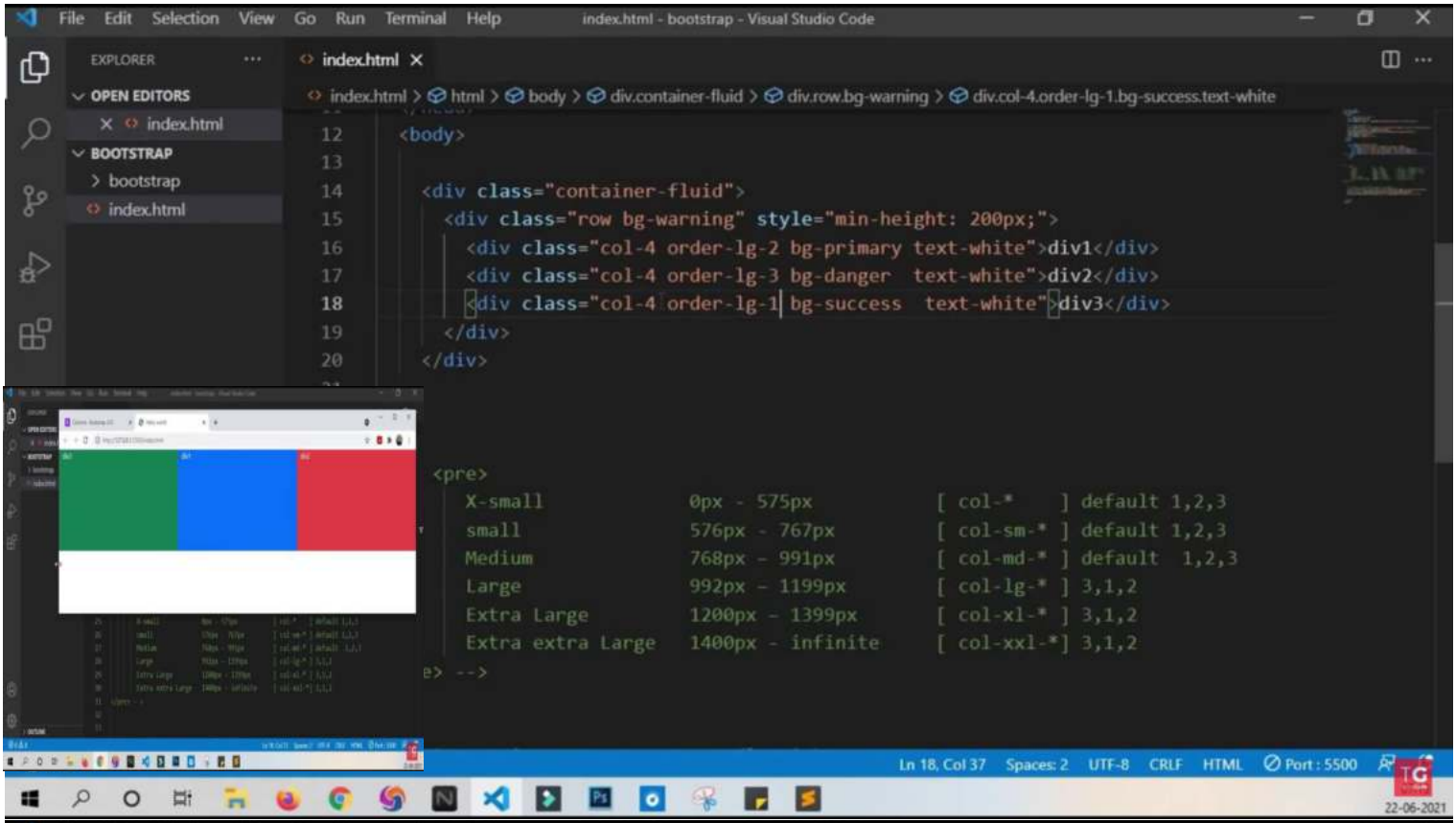





Bootstrap Tutorial in Hindi With 2 Projects for Beginners | Complete Bootstrap 5 Tutorial in Hindi
LastPass
How It Works
Personal
Business
Pricing
Resources
Log In
Get LastPass Free
PASSWORD GENERATOR TOOL
Generate a secure password
Use our online password generator to instantly create a secure, random password.
e6baYr%Hs@oI
Your Choices Regarding Cookies on this Site
Change Settings
Accept Recommended Settings

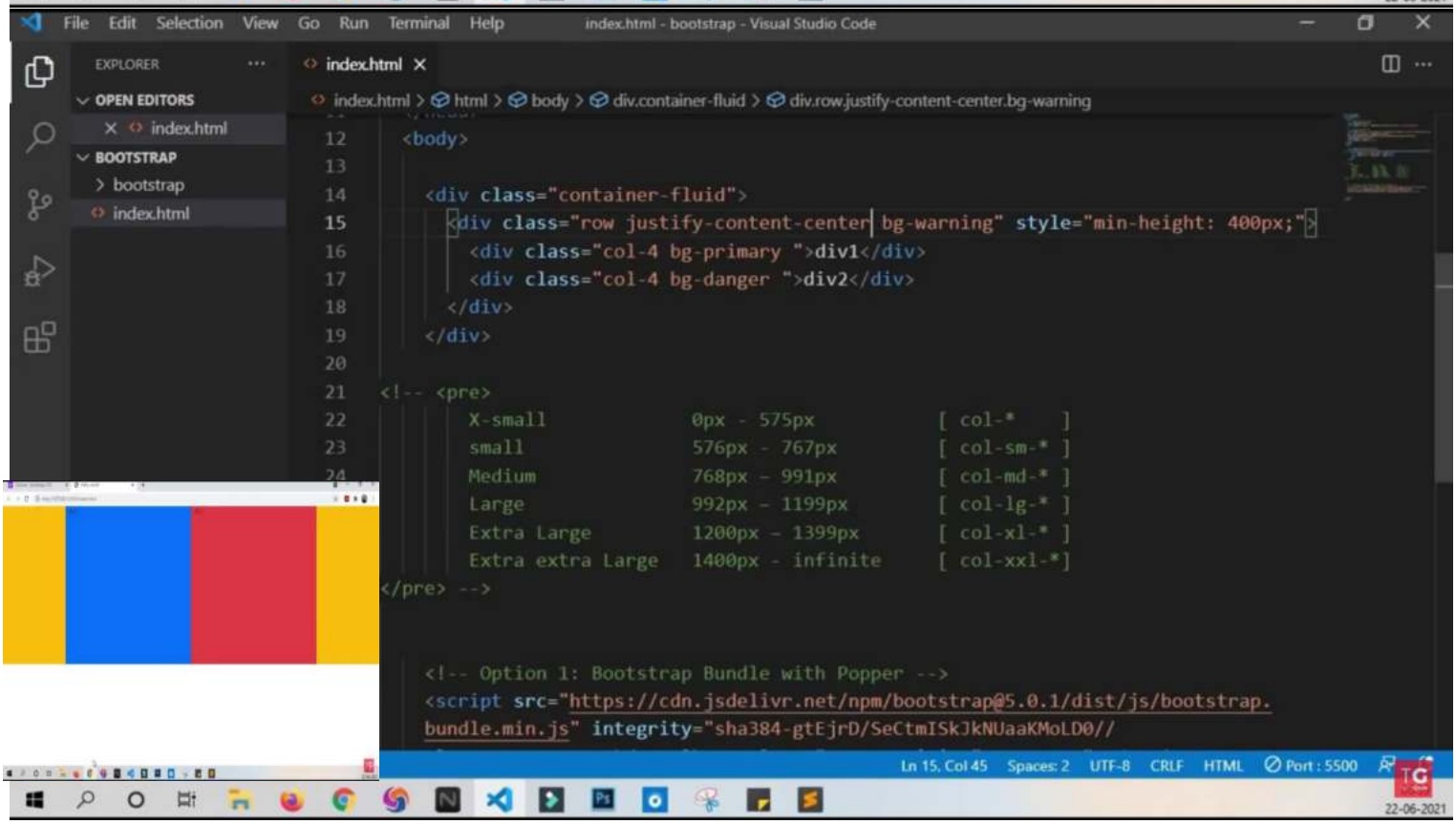

Bootstrap Tutorial in Hindi With 2 Projects for Beginners | Complete Bootstrap 5 Tutorial in Hindi
Secure Password Generator
English
Password Length:
Include Symbols:
Include Numbers:
Include Lowercase Characters:
Include Uppercase Characters:
Exclude Similar Characters:
Exclude Ambiguous Characters:
Generate On Your Device:
Auto-Select:
Save My Preference:
Load My Settings Anywhere:
Generate Password
GP Maker
Advanced...


ADDRESS
Block C Janakpuri, Delhi, India 110018
LET TALK
+91 90156642440
GANAREL SUPPORT
mailid@gmail.com
SEND US A MESSAGE
FULL NAME
MAIL
SUBJECT
MESSAGE


File Edit Selection View Go Run Terminal Help
index.html - bootstrap - Visual Studio Code
EXPLORER
OPEN EDITORS
index.html
BOOTSTRAP
bootstrap
index.html
index.html > html > body > div.container-fluid > div.row.bg-warning > div.col-4.order-lg-1.bg-success.text-white
<body>

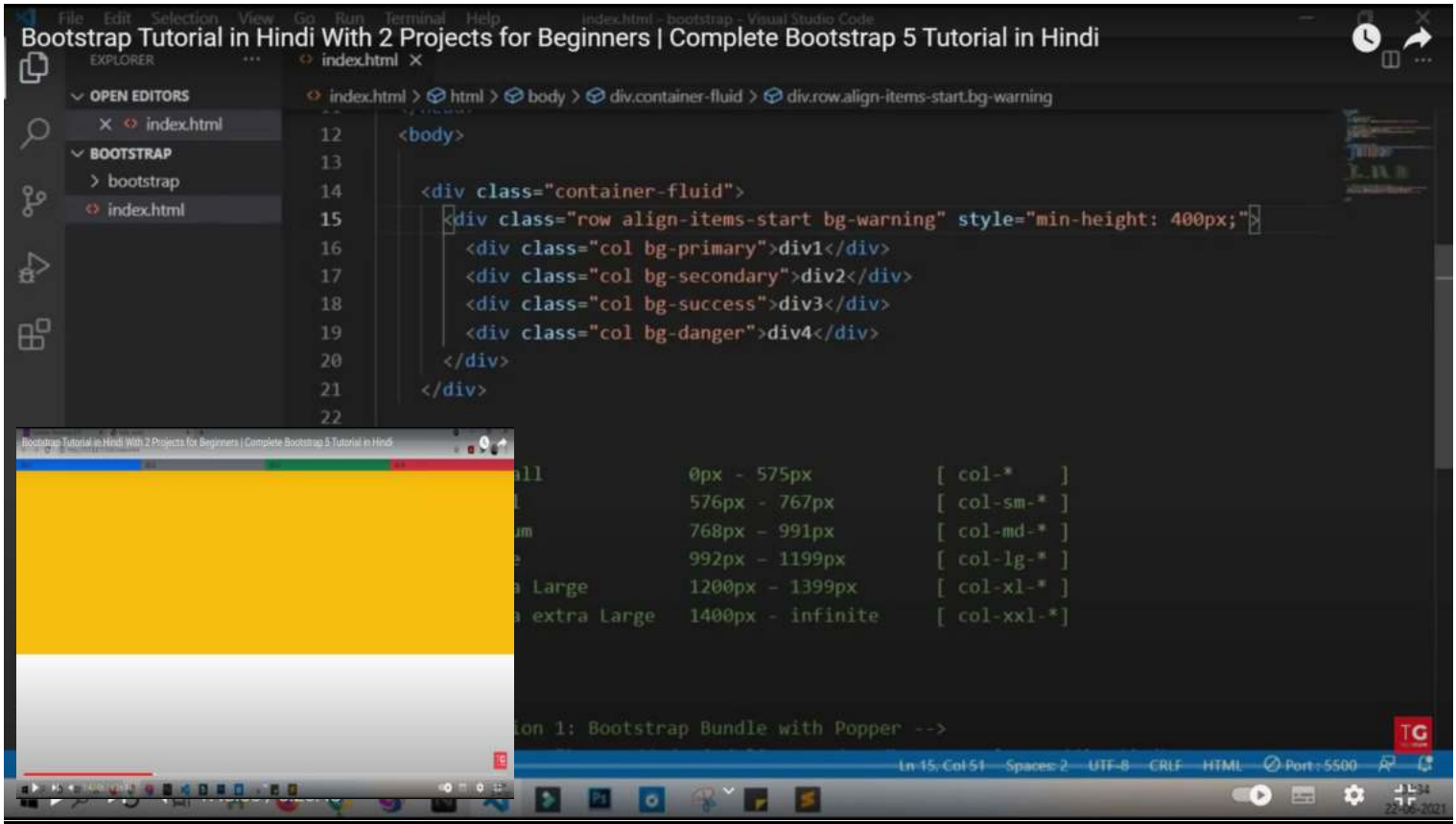
div1
div2
div3

<pre>
X-small 0px - 575px [col-*] default 1,2,3
small 576px - 767px [col-sm-*] default 1,2,3
Medium 768px - 991px [col-md-*] default 1,2,3
Large 992px - 1199px [col-lg-*] 3,1,2
Extra Large 1200px - 1399px [col-xl-*] 3,1,2
Extra extra Large 1400px - infinite [col-xxl-*] 3,1,2
e> -->
Ln 18, Col 37 Spaces: 2 UTF-8 CRLF HTML Port : 5500
22-06-2021


File Edit Selection View Go Run Terminal Help
index.html - bootstrap - Visual Studio Code
EXPLORER
OPEN EDITORS
index.html
BOOTSTRAP
bootstrap
index.html
index.html > html > body > div.container-fluid > div.row.justify-content-center.bg-warning
<body>

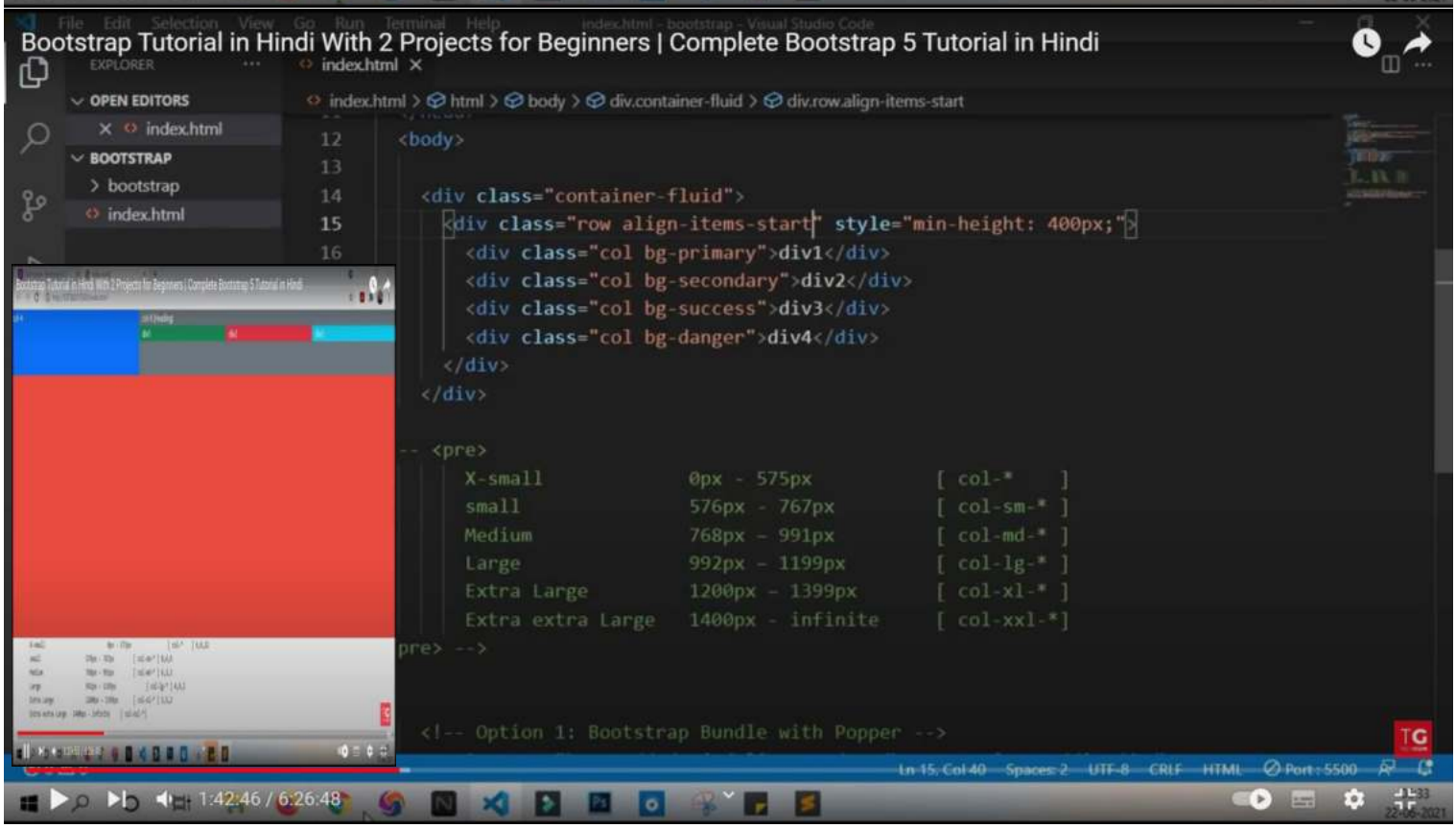
div1
div2

<!-- <pre>
X-small 0px - 575px [col-*]
small 576px - 767px [col-sm-*]
Medium 768px - 991px [col-md-*]
Large 992px - 1199px [col-lg-*]
Extra Large 1200px - 1399px [col-xl-*]
Extra extra Large 1400px - infinite [col-xxl-*]
</pre> -->
<!-- Option 1: Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.
bundle.min.js" integrity="sha384-gtEjrD/SeCtmISkJkNUaaKMoLD0//
Ln 15, Col 45 Spaces: 2 UTF-8 CRLF HTML Port : 5500
22-06-2021


Bootstrap Tutorial in Hindi With 2 Projects for Beginners | Complete Bootstrap 5 Tutorial in Hindi
EXPLORER
index.html
OPEN EDITORS
index.html
BOOTSTRAP
bootstrap
index.html
index.html > html > body > div.container-fluid > div.row.align-items-start.bg-warning
<body>

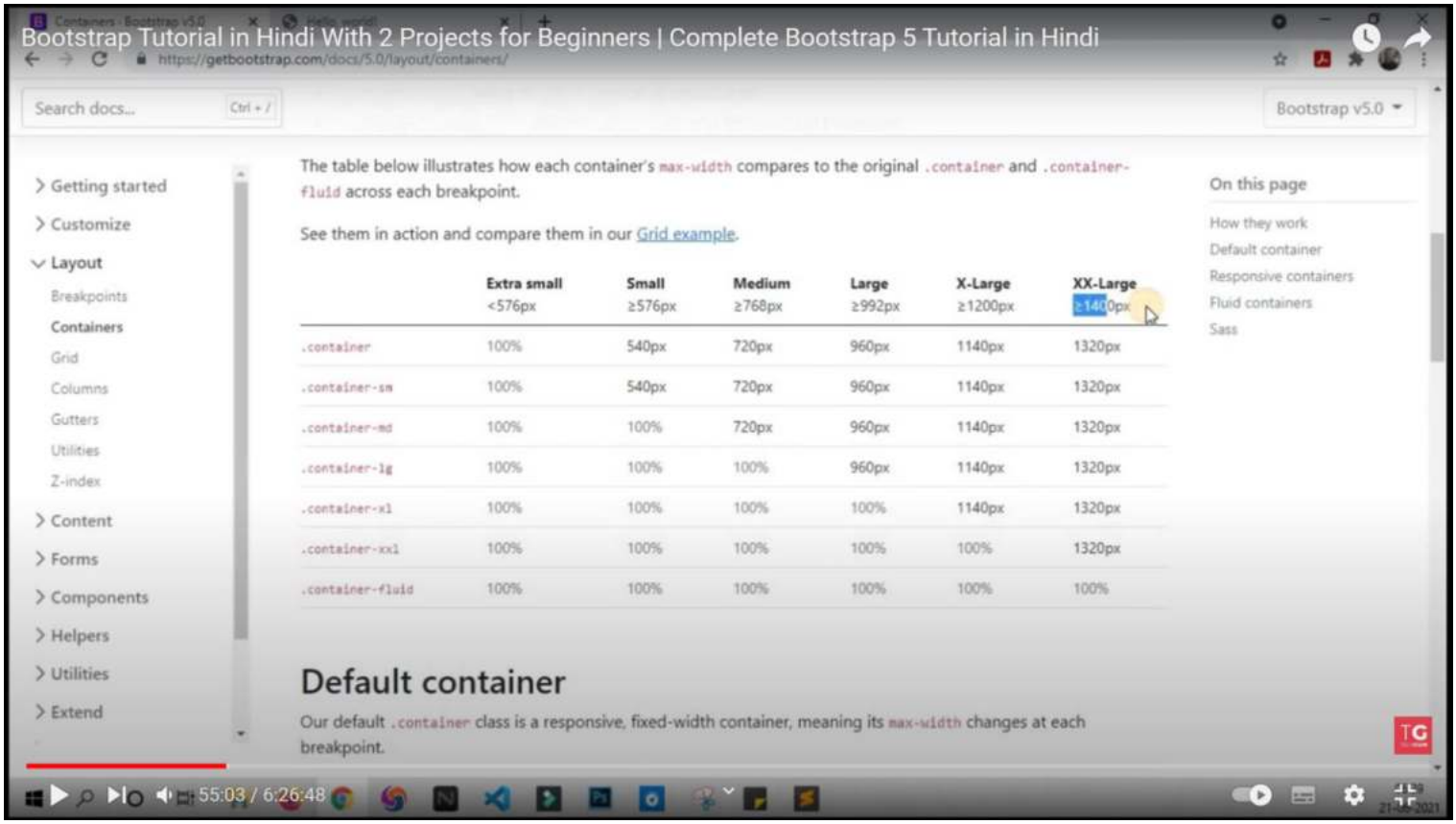
div1
div2
div3
div4

0px - 575px [col-*]
576px - 767px [col-sm-*]
768px - 991px [col-md-*]
992px - 1199px [col-lg-*]
Large 1200px - 1399px [col-xl-*]
extra Large 1400px - infinite [col-xxl-*]
on 1: Bootstrap Bundle with Popper -->
Ln 15, Col 51 Spaces: 2 UTF-8 CRLF HTML Port : 5500


Bootstrap Tutorial in Hindi With 2 Projects for Beginners | Complete Bootstrap 5 Tutorial in Hindi
EXPLORER
index.html
OPEN EDITORS
index.html
BOOTSTRAP
bootstrap
index.html
index.html > html > body > div.container-fluid > div.row.align-items-start
<body>

div1
div2
div3
div4

<pre>
X-small 0px - 575px [col-*]
small 576px - 767px [col-sm-*]
Medium 768px - 991px [col-md-*]
Large 992px - 1199px [col-lg-*]
Extra Large 1200px - 1399px [col-xl-*]
Extra extra Large 1400px - infinite [col-xxl-*]
pre> -->
<!-- Option 1: Bootstrap Bundle with Popper -->
Ln 15, Col 40 Spaces: 2 UTF-8 CRLF HTML Port : 5500
1:42:46 / 6:26:48



Here, >=1400px or XX-large means , when screen size is XX OR 1400px or more than 1400px then container width will be 1320px and less than or equal to 576px screen size , container will be 100% .

There are three types of container in bootstrap
Container-fluid class gives 100% width of the container in each device.
Container class gives margin from left and right.
Note:- row class means it has 12 columns. So, 12 columns can be taken inside it.

GITHUB

How to create repository and put big file online
Go to https://github.com/Srz20/
Then create new repositary. Then upload file then open git hub installed in windows as it will allow to upload big project then click on Add drop down and clone repository and then find the repository you have created or click on URL tab and to go the github.com and copy the path of https that you will find under code tab and past in the desktop version and paste the copied url and then clone, it will start cloning. Once it is cloned, it will open a new page NO LOCAL CHANGES
Click on show on explorer. It will open a new page, like .git file will be there, paste all the files of your projects here. Once it is done, go to the desktop version git hub, here you will see green page will files.filll section before description about your web and then click on commit to main. Now, click on push origin, it will download like 100%

Now go to the browser git hub and click on your repository, you will see your all files are downloaded here.
Now go the settings
You will see your repository name that you can rename if you want.
Click on pages, you will see GitHub Pages is currently disabled under BRANCH , change it to main.
Wait for a while and go the settings again and then you will see the link of your website

HOW TO COPY THE WEBSITE

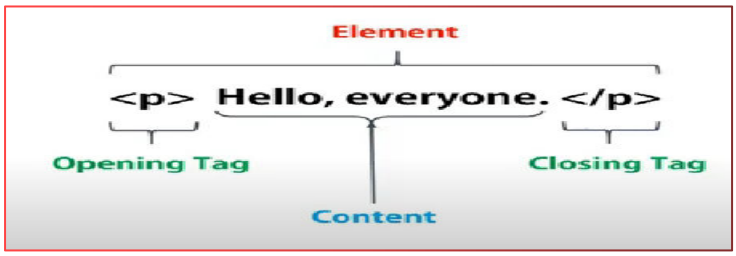
KTGB12621
How to Download source code of any website | Latest Trick (2020) - YouTube

1. What is HTML(v)

HTML is hypertext markup language for creating web-pages. It describes the structure of the web-page. It consists of series of elements. HTML elements tell the browser how to display the content. We need HTML editors to create and modify web-pages such as notepad, notepad++, sublime text, visual studio, atoms etc. The purpose of browser is to read HTML documents and display them

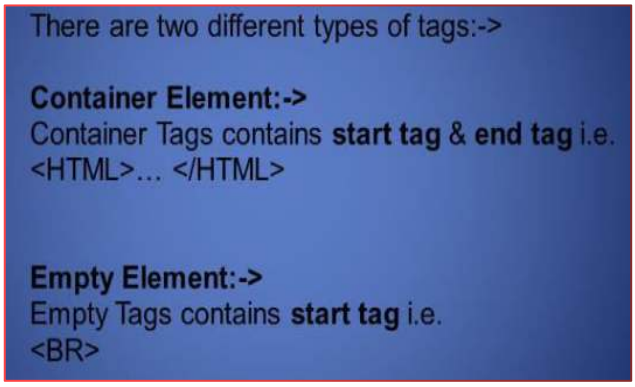
2.What is TAG(v)

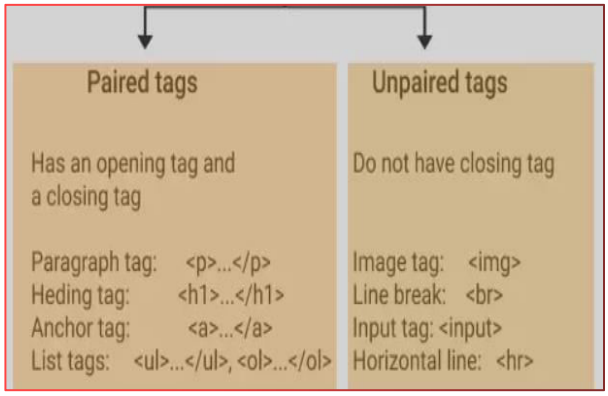
HTML TAG is character or word which is placed between Less than symbol"<" and greater than symbol ">". It decides which formatting will be applied in any content or text. For example, - p is a word which is placed between less and greater symbol, this p tag will be used for formatting the content “Sarfraz” if Sarfraz taken. The combination of opening tag, content or text and closing tag is called HTML element.

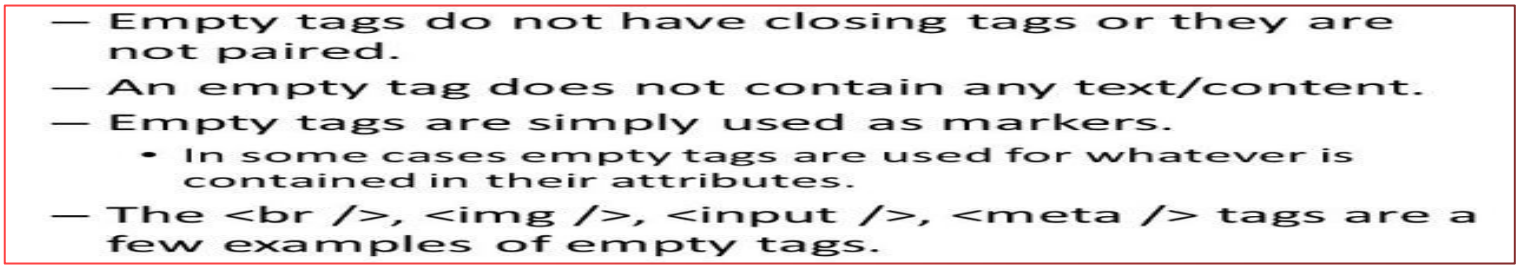


In HTML file, content/text is surrounded by Tags.

There are two types of tags.








```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>HTML HEADINGS</title>
</head>
<body>
<!--HTML_HEADINGS h1 -->
<h1>This is a Heading</h1>
```

```
<p>This is a paragraph.</p>
<div>
    <img src="H:\UI DEVELOPMENT\Project 1 Pics\1630426673318.jpg"
width="220" height="220"> <br> <h4>hdkjkj</h4>

<script>  </script>

</div>
</body>
</html>
```

3. Attribues

HTML Attributes provide additional information about HTML elements. It is always specified in the start tag and come in name/value pairs like name="value".

Attribute :-
यह किसी भी Tag की Property को Decide करता है या Additional Information के बारे मे बताता है। कि Content को और अच्छी तरह से कैसे present कर सकते है ।
It defines additional information about That Element.
Attribute दो Parts मे होता है।
Attribute divides in two part .
Name & Value |
जैसे :- <hr Color="Red">
यहां पर Color Name और Red Value है ।

```
<h2>ATTRIBUTES</h2>

<p style="color:red;">The text written in the paragraph will be red</p>

<p style="background-color:pink;">The background color of the text written in the paragraph will be pink</p>



<!--We use font-family attribute in style to use the font type for the HTML element-->
<p style="font-family:cursive;">The text will font type accordingly </p>


<!--We use font-size attribute in style to change the font size of the text-->
<p style="font-size:100%;">The text written in the paragraph will have changed font size</p>


<!--We use text-alignment attribute in style to use the text center, left or right-->
<p style="text-align:center;">The text written in the paragraph will be in center. </p>

<lang> is an attribute. The attribute <lang> is used to declare the language of the web page. It is written inside the <html>

<title> is an attribute. The attribute <title > is used to give extra information about the element
```

4. IMAGE TAG

```
<h2> IMAGE TAGE <h2>
<!-- <img> tag is used to embed an image. The attribute of <img> tag is src(to give the address of the image you want to
display), width & height to give width & height of the image and alt if the image not displayed message will be displayed. It is
also said like an alternate text will be displayed-->

<img src="voterid.jpg" width="100" height="200" alt="voter id not displayed" align="right" id="any">
```

5. ANCHOR TAG (v)

HTML provide a hyperlink called anchor tag which specify the link in the webpage.

An anchor tag is used to link two sections, web pages, or website templates in HTML.

Anchor tag defines a hyperlink. The attribute of Anchor tag is "href" used to link the page or href represents hyper text reference.

Its format is:

****

Where 'href' is an attribute of the anchor tag used to identify the sections in a document, the 'link' is defined in the target attribute, which is to be linked.

_self will open the targeted link in the same window or tab.(default)

_blank will open the targeted link in the new window or tab.

_top will open the targeted link in the full window body.

```
<h2>Anchor Tag </h2>
<p> Simple Link/External Link </p>

<a href="https://www.google.co.in/webhp"> It will give a link and clicking on that you will visit google page</a>
<p> local links used to link the page available in the same folder or same web or which in inside you are working on <p>
<a href="HTML_HEADING - DIV TAG.html"> HTML_HEADING - DIV TAG will open</a>
<p> Link with target </p>
<!--Link with target will help to open the linked webpage or page in next tab not in the same tab-->
<a href="https://www.google.co.in/webhp" target="_blank"> google page will open in another tab</a>
<p> Link with image </p>
<!--Link with image will help to open the linked webpage or page by clicking on an image not in the text-->
<a href="https://www.google.co.in/webhp" target="_blank"> <img src="C:\Users\sarfaraz\Desktop\voterid.JPG" width="100"
height="100"> When you click on the image, google website will open as href has google address</a>
<p> To open the video or audio with with the equeal size of browser</p>
<a href="https://www.google.co.in/webhp" target="_top"></a>
<p> bookmark with id </p>
<!--bookmark with id will bring the page from down to up like homepage when you click on the link-->
<a href=#home> GO TO HOME PAGE</a>
```

6) Does a hyperlink only apply to text?

NO, you can use hyperlinks on text and images both. Here, you can see both text and image are being used for hyperlink

```
<a href="https://www.google.co.in/webhp" target="_blank">
        <img src="C:\Users\sarfaraz\Desktop\voterid.JPG" width="100" height="100">    // image used with hyperlink
        When you click on the image, google website will open as href has google address  // text used with hyperlink
</a>
```

7) How to create a hyperlink in HTML?

The HTML provides an anchor tag to create a hyperlink that links one page to another page. These tags can appear in any of the following ways:

```
<a href="" target="_self">Hello Sarfraz</a>
```

- Unvisited link - It is displayed, underlined and blue. (Initially it will be blue in color)
- Visited link - It is displayed, underlined and purple. (When you click on it, it will be visited and turn purple)
- Active link - It is displayed, underlined and red. (As soon as it is clicked it will turn red which means it is active because it will open the targeted link)

Hello Sarfraz

8) LIST TAG

```
    <!--HTML_LIST TAG-->
    <!--IN HTML, list tag has two lists one is ordered list(ol) and another is unordered list(ul) -->
    <!--Un ordered list with some list items-->
    <h2> UN ORDERED LIST </h2>
    <ul>
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ul>

    <!--Various list style types (disc, square, none) -->
    <ul type="square">
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ul>
   <ul type="none">
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ul>
```

UN ORDERED LIST

- Sarfraz
- Sarf
- Saraz

- Sarfraz
- Sarf
- Saraz

Sarfraz
Sarf
Saraz

```
    <!--Ordered list with some list items. It has default 1,2,3 and so on-->

<h2> ORDERED LIST </h2>
    <ol>
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ol>
 <!--Ordered list with a. A, i, I and so on-->
    <ol type="a">
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ol>
    <ol type="A">
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ol>
    <ol type="i">
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ol>
    <ol type="I">
        <li> Sarfraz </li>
        <li> Sarf </li>
        <li> Saraz </li>
    </ol>
</body>
```

ORDERED LIST

1. Sarfraz
2. Sarf
3. Saraz

a. Sarfraz
b. Sarf
c. Saraz

A. Sarfraz
B. Sarf
C. Saraz

i. Sarfraz
ii. Sarf
iii. Saraz

I. Sarfraz
II. Sarf
III. Saraz

```
</html> I am <mark> marked</mark>text
```

I am markedtext

9. SPAN TAG

```
<!--HTML_SPAN TAG within a paragraph-->
<!--HTML_SPAN TAG is used to design/color a particular word or text-->
<h1> My Intro To Explain p tag</h1>
<p>My name is <span style="background-color:darkred;">Sarfraz</span>. I am from Dhanbad. This this how you can write any
paragraph like this using p tag</p>
</body>
```

My name is Sarfraz. I am from Dhanbad. This this how you can write any paragraph like this using p tag

10. TEXT FORMATTING TAG (v)

The HTML formatting is a process of format the text for a better look and feel. It uses different tags to make text bold, italicized, underlined.

```
    <!--<b> and <strong> tag are used to make a text bold-->
        I am <b>bold</b> text
    <!--<i> tag is used to make a text Italic-->
        I am <i>Italic</i> text
    <!--<em>  tag is used to make a text emphasized-->
        I am <em> emphasized </em> text
    <!--<mark> tag is used to make a text marked-->
        I am <mark> marked</mark>text
    <!--<small> tag is used to make a text small-->
        I am <small> small </small> text
    <!--<del> tag is used to make a text deleted-->
        I am <del> deleted </del> text
     <!--<ins> tag is used to make a text inserted-->
        I am <ins>inserted </ins> text
      <!--<sub> tag is used to make a text subscript-->
        I am <sub> subsript </sub>text
    <!--<sup> tag is used to make a text superscript-->
        I am <sup> superscripted </sup> text
```

I am **bold** text
I am *Italic* text
I am *emphasized* text
I am marked text
I am small text
I am ~~deleted~~ text
I am inserted text
I am subsriped text
I am superscripted text

<i> and tags are used to make the text italic but tag has got a different importance for search engine optimization. The same is with the tag because <b> and both have the same output.

11. Differentiate between logical and physical tags.

Physical tags are used to provide the visual appearance to the text whereas Logical tags mainly focus on adding logical or semantic value to the text.
<b> and <i> tags are physical tags but and tags are logical tags.

<i> and tags are used to make the text italic but tag has got a different importance for search engine optimization. The same is with the tag because <b> and <bold> both have the same output.

12. HTML ENTITY

The HTML character entities are used as a replacement for reserved characters in HTML. You can also replace characters that are not present on your keyboard by entities. These characters are replaced because some characters are reserved in HTML.

```
<h2>HTML ENTITIES TAG</h2>

<!--HTML ENTITIES TAGS ARE USED TO DISPLAY SPECIAL SYMBOL OR CHARACTER-->
<p>This is a &lt; paragraph &gt;.</p>
<!-- &lt; and &gt; are two entities for less than symbol < and greater than symbol > -->
<!-- Registered Trade Mark Entity-->
<h4>Genpact<sup>&reg;</sup></h4>
<!-- Copy Right-->
<h4>Copyright &copy; 2021 </h4>
<!-- Rupees Symbol-->
<h4>This is &#8377;500</h4>
<!-- &#8377; is used for Rupees Symbol and &copy;  is used for Copy Right-->
```

HTML ENTITIES TAG

This is a < paragraph >.

Genpact®

Copyright © 2021

This is ₹500

13. TABLE TAG

Or

If you want to display some HTML data in a table in tabular format, which HTML tags will you use?

It is used to display table, we need to apply css in order to give border.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>HTML TABLE TAG</title>
    <style>
        table, th, td {
            border: 2px solid red;
            border-collapse: collapse;
            text-align: center;
        }
        #custom-table thead {
            background-color: yellow;
            color: blue;
        }
    </style>


       < !-- We need to apply CSS to make border on the table and border-collapse to merge all the separate tables, text-align
             to center the text under the table-->

  </head>
```

Without border-collapse: collapse /seperate

SARFRAZ	SArfraz	pr
alam	add	alam

```
<body>
    <table width="40%">                          <!-- width is used to make the table width equal to the browser width-->
        <thead>
            <tr>
                <th> SARFRAZ</th>
                <th> SARFRAZ</th>
                <th> SARFRAZ</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>alam</td>
                <td>alam</td>
                <td>alam</td>
            </tr>
         </tbody>
    </table>


 <p>Table with Col Span</p>                       <!-- used to mearge two columns, here "2" means 2 colums were merged-->
    <table>
        <thead>
            <tr>
                <th> SARFRAZ</th>
                <th colspan="2"> SARFRAZ</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>alam</td>
```

SARFRAZ	SARFRAZ	SARFRAZ
alam	alam	alam

Table with Col Span

SARFRAZ	SARFRAZ	
alam	alam	alam

```
            <td>alam</td>
            <td>alam</td>
        </tr>
    </tbody>
</table>
```

```
<p>Table with Row Span</p>                          <!-- used to mearge two rows, rows "2" means 2 rows were merged-->
  <table>
      <thead>
          <tr>
              <th> SARFRAZ</th>
              <th> SArfraz</th>
              <th> pr </th>
          </tr>
      </thead>
      <tbody>
          <tr>
              <td rowspan="2">alam</td>
          </tr>
          <tr>
              <td> add</td>
              <td>alam</td>
          </tr>
      </tbody>
  </table>
```

14. HEAD TAG AND BODY TAG AND META TAG(v)

Head tag and Meta tag provides information about the HTML document. There is only one head and body tag in the entire HTML document. Head tag is used after the opening tag of HTML and before opening tag of body. It may contain
Meta, style, title, link, and script tag etc.
Body tag contains so many tags that appear on the web-page.

<HEAD> tags

- To **give title** to the document <head> tag is used.
- <head> tag includes another tag called **<title> tag**
- A title tag is also called a **container tag.**
- Content written within title tags **does not visible on web page but it is visible on Title bar.**

```
<head>
<title> Information Studies </title>
</head>
```

<BODY> tag

- <body> tag contains the **main text** which appears on the web page.
- The <body> tag starts just below the </head> tag and **indicates the beginning of the html text.**
- </body> tag indicates the end of text.

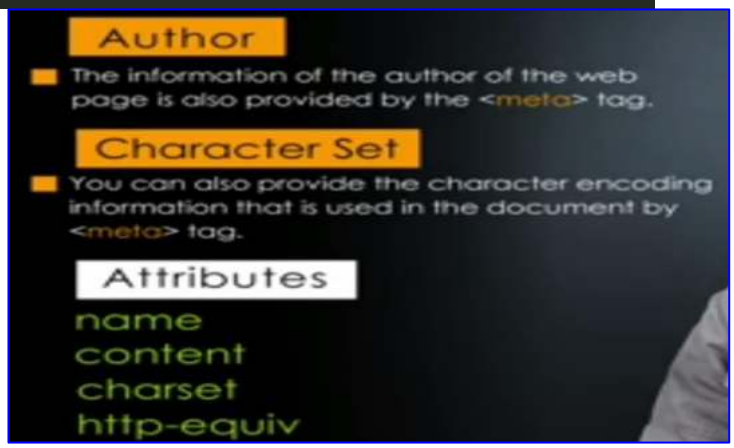
Example:

```
<body>
We are learning HTML.
</body>
```

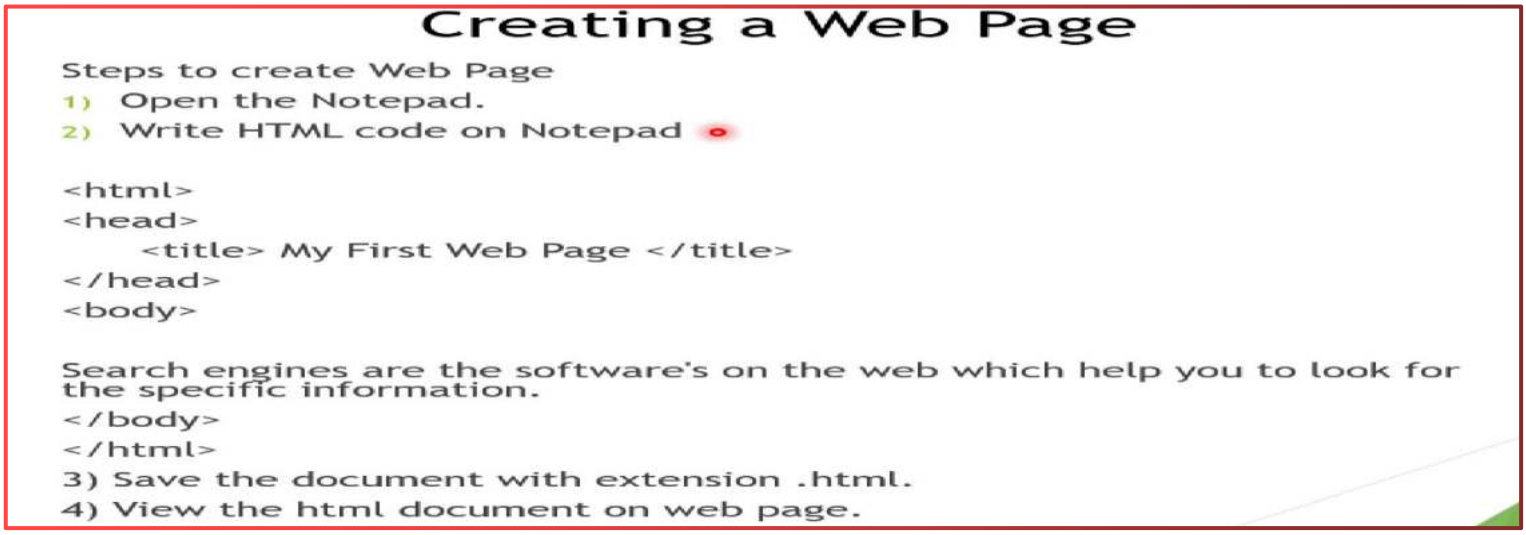
Meta tag provides information about the HTML document. It is used by the browsers and the search engine.
It is an unpaired tag and the content which is taken as an attributes in meta tag is not displayed on the web-page.
We use attributes such as name, content, charset and http-equiv, The name attribute tells about the name of the meta data(meta data provided by meta tag) like description, keywords, author, viewport, refresh etc. The content attribute gives value associated with name attribute.
Meta tag provides the author name of the web page, description(if we don't define the description then search engine will take first some lines of the web page, It also provides the keyword to the search engine so that search engine understand what all topics have been covered on the web-page and rank in the google if that keyword is searched in the google by the users and character encoding information of the document.
Initial-scale=1.0 means the content or data used in the web-page will be 100% depending on the screen the user use.

If we use attribute HTTP-quiv refresh then it will refresh the page in every 5 seconds and if we want that google.com open in 5 seconds then url will take and put the path.

```
<!doctype html>
<html>
<head><title>All Type Tutorials</title>
<meta charset="utf-8">
<meta name="description" content="All Type Tutorials are free.">
<meta name="keywords" content="Graphics,Web,Audio,Video,Vfx">
<meta name="author" content="Husain Sir">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="refresh" content="5; url=http://www.google.com">
</head>
```



14. CREATING A WEB PAGE



15. What is the key difference between HTML Elements and Tags? Also, Can you separate sections of texts in HTML? Or

How many tags can be used to separate a section of texts?

The key difference between the HTML elements and tags is that there is nothing between the opening and closing tag whereas there is something like content or text between opening and closing tag in element.

HTML Elements

The sections of the web page, such as a paragraph, an image, or a link is an element, and an element has a certain way of execution. For example, the link is used to be clicked, and the text boxes can be used to input text.

HTML Tags

HTML elements communicate with the browser how to represent the text and become HTML tags when enclosed within angular brackets <>.

Three tags are used to separate the texts.

<p> tag–Use this tag for writing a paragraph of any text.

 tag – Use this for separating a line of text. This will break a current line and also shift the flow of a text over to new line.

<blockquote> tag– It is used to take any section from another site and use in your site. This blockquote tells the browswer where the section has been taken from. We use site page from where we have taken the section and paste in the attribute of blockquote called cite.There will be no change in the output.

Blockquote:-

```
    <h3>blockquote</h3>
    <blockquote cite="https://www.flaticon.com/authors">
        <h4>Authors</h4>
        <p>
            Flaticon is home to many talented icon designers. Each works in different <br>
            and explores different concepts, so you can totally find the right icons for <br>
            your project. Do you want to join our designer community?</p>
    </blockquote>
</body>
```

blockquote

Authors

Flaticon is home to many talented icon designers. Each works in different and explores different concepts, so you can totally find the right icons for your project. Do you want to join our designer community?

16. What are Lists in HTML?

HTML lists are used to group a set of related items in lists. It is defined with an <li> tag.

Some commonly used HTML lists:

Ordered List (HTML tag: <ol>)

Unordered List (HTML tag: <ul>)

Description List (HTML tag: <dl>)

Menu List (HTML tag: <menu>)

Directory List (HTML tag: <dir>)

17. Define HTML Layout.

HTML layout specifies a way in which the web page is arranged. Every website has a specific layout to display content in a specific manner.

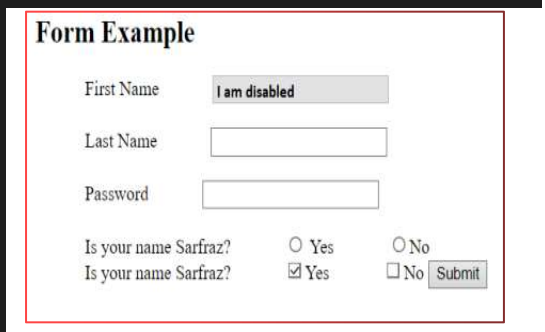
Following are different HTML5 elements which are used to define the different parts of a web-page.

- <header>: It is used to define a header for a document or a section.
- Main content where the entire web page content is included.
- <nav>: It is used to define a container for navigation links
- <section>: It is used to define a section in a document
- <article>: It is used to define an independent, self-contained article
- <aside>: It is used to define content aside from the content (like a sidebar)
- <footer>: It is used to define a footer for a document or a section

18. What are Forms in HTML?

Forms are used to collect the user information when they are filled, and details are provided to save into the database.

```
<form>
          <label for="First Name"> First Name</label>
          <input type="text" id="User Name" disabled><br><br>
          <label for="Last Name"> Last Name</label>
          <input type="text" id="Last Name"><br><br>
          <label for="Password"> Password</label>
          <input type="Password" id="Password"><br><br>
          <label>Is your name Sarfraz?</label>
          <input type="radio" value="Yes" name="Sarfraz"> <label>Yes</label>
          <input type="radio" value="No" name="Sarfraz"><label>No</label> <br>
          <label>Is your name Sarfraz?</label>
          <input type="checkbox" value="Yes" checked>Yes
          <input type="checkbox" value="No">No
           <input type="Submit" id="Submit"> <br><br>

 </form>
```



19. What is the Use of Comments in HTML?

Answer:

Comments are used in an HTML document to make important notes and help developers mention any modification to be incorporated afterward. They are not displayed in the

browser when the code is executed. A comment is always written in between the '—' symbol at the beginning and end of the angular brackets.

Syntax:

<!-- 'Safraz'--!> , // , /* statement */

<!—'Comment' !–>

20. What is HTML5?

HTML5 is the improved HTML version released in 2014 by the World Wide Web consortium.

It has set forth the following new characteristics to be learned by professionals:

DOCTYPE declaration: To declare the HTML document type to instruct the web browser about the markup language.

Main: The main tag defines the primary section in the document related to the central content of a document with a <main> tag.

Section: It is used to define specific sections in a document such as a chapter, header, footer, or any other section, and is specified with the <section> tag.

Header: The header tag defines the title or heading of a document or its section. It is specified with the <header> tag.

Footer: The footer tag defines the section of a document that contains information such as copyright or author's information. It is designated with the <footer> tag.

Article: The article tag represents an independent or self-contained part of the content of a document with the tag <article>.

21. What is Semantic HTML?

A Semantic HTML or Semantic element clearly describe its meaning to both the browser and the developer and they are as follows. Here it is clear from its name that what is inside me.

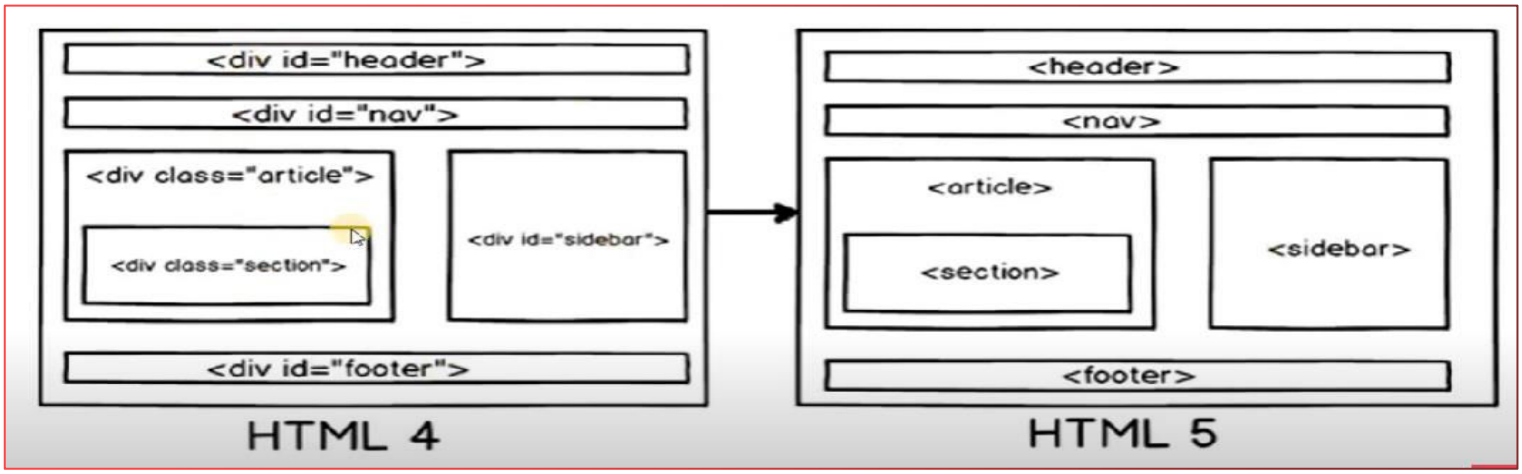
New Semantic Elements in HTML5

```
<article>
<aside>
<details>
<figcaption>
<figure>
<footer>
<header>
<main>
<mark>
<nav>
<section>
<summary>
<time>
```

In another word, It is a tag or element that can tell by its name that who I am or what my purpose is.

Non Sementic Elements do not tell anything about its content like what inside me such as div and span tag,

Both are the same but better to use the sementic element instead of div because it clearly tell the browser and developer who I am.



It is better to use writing the code like this.For Example:

```
<div id="header"></div>
<div class="section">
   <div class="article">
      <div class="figure">
         <img>
         <div class="figcaption"></div>
      </div>
   </div>
</div>
<div id="footer"></div>
```

```
<header></header>
<section>
    <article>
        <figure>
            <img>
            <figcaption></figcaption>
        </figure>
    </article>
</section>
<footer></footer>
```

  and   tags which are used to bold and italic statements in HTML are replaced with and tags in semantic HTML.

22. What is an Image Map? (v)

An Image map lets you link different web pages with a single image. It is represented with the <map> tag. Every employer expects the applicant to know about this, and this has been one of the most commonly asked **HTML interview questions**.

We use image map to hyperlink or visit the different page on clicking on the image. It will make the cursor hyperlink as soon as we hover the curson on the image we have used in the image map. We need to the same height and width of the image to use image map that we can get from image property. We also need to use an attribut called usemap and its value with prefix # should be equal to the value of map attribute name. In area tag, we will take attribut shape to display the shape of the image. Since image is of rectangle shape so rect taken as value and coords will have x and Y axis coordination, we will open the image in paint and put the cursor on left top of the image and write 15,7(as shown on the left bottom) and then right bottom cursor coordinates. The cursor point will convert into hand pointer as soon as it comes under the coordinate area and we go the href link page on clicking on that area of the image.

C: > Users > pione > OneDrive > Desktop > FED > HTML5 > imageMap.html > html > body

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>image map</title>
</head>

<body>

    <img src="imageMap/image.jpeg" alt="" usemap="#sarfraz" height="900px" width="600px">
    <map name="sarfraz">

        <area shape="rect" coords="15, 7, 587, 889" href="https://www.youtube.com/watch?v=PxL--YcYdUI">

    </map>

</body>

</html>
```

C:\Users\pione\OneDrive\Desktop\FED\HTML5\imageMap

23. Why is the Embed/Object/iframe Tag Used in HTML?

These three are used to add external or internal files/videos/audios on the webpage.

Embed and Object are static but iFrame is dynamic because it can adjust itself depending on the updated plugins and gives guidelines to the users if internet is not connected or unavailable. It gives error message if anything is wrong but not by embed or object.

An Embed Tag/object/iframe is used for including a Video or Audio in an HTML Document. A source of audio or video file to be displayed on the web-page is defined within an Embed tag as:

```
<object data="picture.jpg" width="159" height="194"></object>
<embed src="picture.jpg" width="159" height="194"></embed>
<iframe src="picture.jpg" width="159" height="194" frameborder="0" scrolling="no"></iframe>
```

24. What is front-end web development?

Front-end development is the part of web development and front end developers use technologies such as HTML, CSS, and JavaScript for creating beautiful websites and increase user experience. Front-end developers use HTML, CSS and JavaScript to communicate with the user. Front-end developers are responsible for the user's first impression and experience of a website. A good front-end developer is one who is able to develop websites that are both functional and easy to use.

25. How does HTML and CSS work together?

HTML and CSS work together to build webpages. HTML (Hypertext Markup Language) is the structural code behind most webpages. HTML is what creates the layout of the webpage, and creates the various headers, footers, text, and images you see when you view a page. CSS (Cascading Style Sheets) is applied on top of the HTML and allows you to modify the look and feel of the HTML structure. It's important to note that when you edit the HTML, you can't see the changes in the CSS due to the separation, but you will see the changes if you view the page in a different browser.

26. Do all HTML tags have an end tag?

No. There are some HTML tags that don't need a closing tag. For example: <image> tag, <br> tag.

---

28. How many types of heading does an HTML contain?

The HTML contains six types of headings which are defined with the <h1> to <h6> tags. Each type of heading tag displays different text size from another. So, <h1> is the largest heading tag and <h6> is the smallest one. For example:

```
<h1>Heading no. 1</h1>
<h2>Heading no. 2</h2>
<h3>Heading no. 3</h3>
<h4>Heading no. 4</h4>
<h5>Heading no. 5</h5>
<h6>Heading no. 6</h6>
```

---

31. How to insert a copyright symbol on a browser page?

You can insert a copyright symbol by using © or © in an HTML file.

---

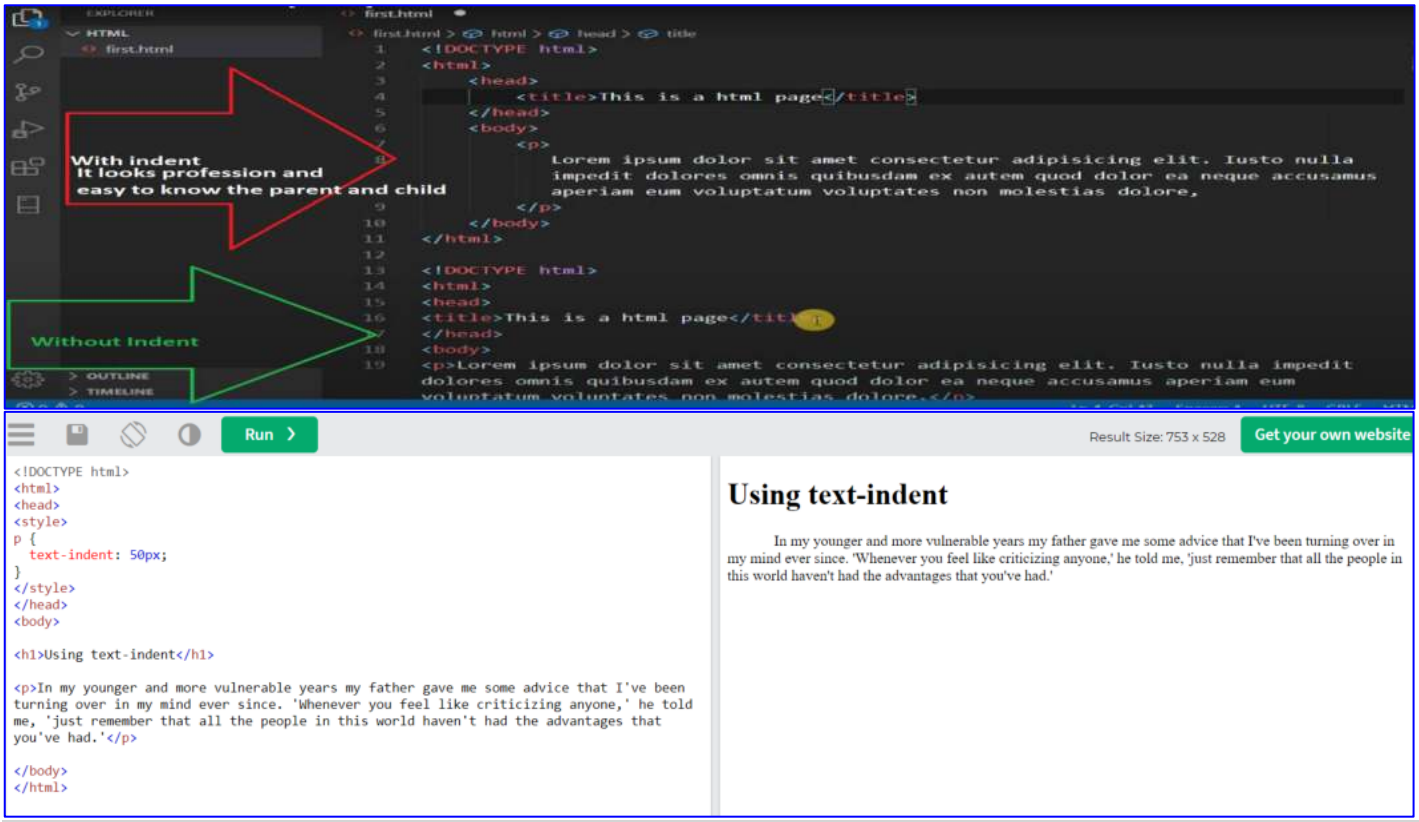
32. How to create a nested webpage in HTML?

The HTML iframe tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML <iframe> tag defines an inline frame. For example:

```
<body>
    <h2>HTML Iframes example</h2>
    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Nisi, commodi.</p>
    <iframe src="https://www.javatpoint.com/" height="300"  width="400" frameborder="0" scrolling="no" ></iframe>
</body>
```

33. How do you keep list elements straight in an HTML file? Or Indent.

You can keep the list elements straight by using indents.



34. What is a style sheet?

A style sheet is used to build a consistent, transportable, and well-designed style template. You can add these templates on several different web pages. It describes the look and formatting of a document written in markup language.

35. Can you create a multi-colored text on a web page?

Yes. To create a multicolor text on a web page you can use   for the specific texts you want to color.

```
<body>
    <font color="red">Sarfraz</font>
    <font color="blue">Sarfraz</font>
</body>
```

Sarfraz Sarfraz

36. Is it possible to change the color of the bullet?

The color of the bullet is always the color of the first text of the list. So, if you want to change the color of the bullet, you must change the color of the text.

```
    <style>
        ul li::marker {color:yellow; }
    </style>
```

```
</head>
<body>
    <ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
        <li>4</li>
        <li>5</li>
    </ul>
</body>
```

- 1
- 2
- 3
- 4
- 5

37. What is a marquee?

You can put scrolling text with a Marquee tag. With the help of this tag, an image or text can be scrolled up, down, left, or right. It is also used for animation.

It has attributes:-
scrollamount: define the speed of the scroll,
scrolldelay: define the time to complete one animation 1000milisecond=1minute,
direction: left(default, start from right & end to left), right/up/down,
Height:- to give the height of the background color,
Width:- to give the width of the background color in percentage,
hspace & vspace:- to give margin horizontally & vertically to the background, behavior: scroll is default ,
Behavior :-Alternate means takrakar phir wapas aana,
slide mein loop khatam hone pe nahi dikhega par scroll mein dikhega.
Loop means how many times you want to run it scroll.

```
    <marquee scrollamount="10" scrolldelay="10" bgcolor="yellow"
            direction="right" height="100px" width="70%"
            hspace="50px" vspace="50px" behavior="alternate" loop=="3" >

            Hello,SArfraz
    </marquee>
```

https://www.youtube.com/watch?v=3dIc6Y587J0

38. How to make a picture of a background image of a web page?

To make a picture a background image on a web page, you should put the following tag code after the </head> tag.

```
<body background = "image.gif">
```

Here, replace the "image.gif" with the name of your image file which you want to display on your web page.

39. What are empty elements?

HTML elements with no content are called empty elements. For example: <br>, <hr> etc.

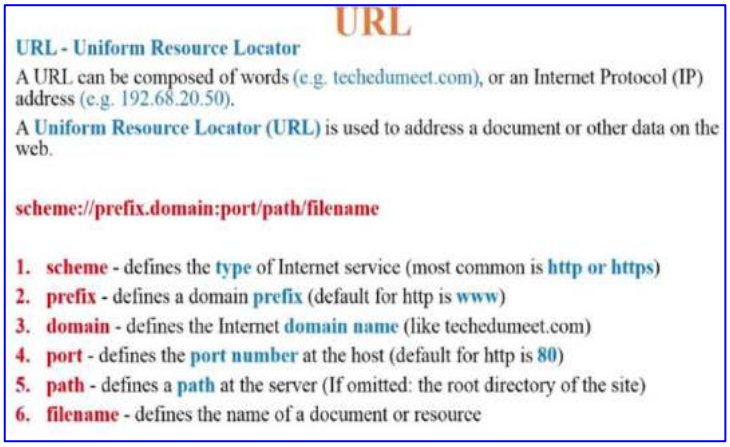
40. What is the use of a span tag? Give one example.

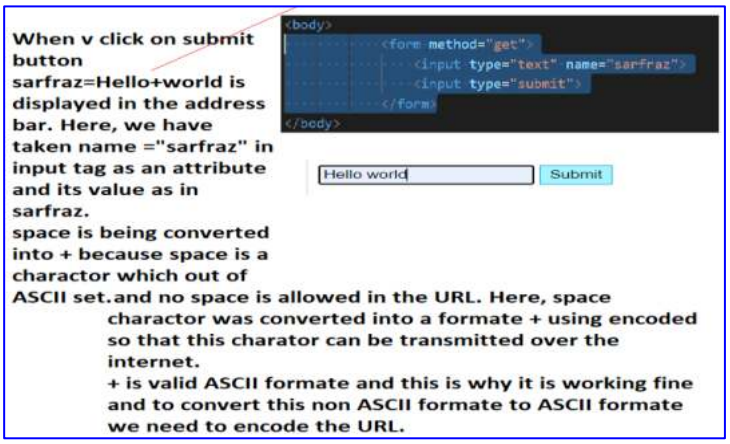
The span tag is used for following things:

- For adding color on text
- For adding background on text
- Highlight any color text

41. Why is a URL encoded in HTML?

URL encoding **converts characters into a format that can be transmitted over the Internet**. URLs can only be sent over the Internet using the ASCII character-set. Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format.





Http or https:/www.sarfraz.com:80

https://www.youtube.com/watch?v=ABlLZPlaCdg

43. Does a <!DOCTYPE html> tag is a HTML tag?

No, the <!DOCTYPE html> declaration is not an HTML tag. There are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc.

So, <!DOCTYPE html> is used to instruct the web browser about the HTML page.

44. What is the canvas element in HTML5?

The <canvas> element is used to draw graphics on the web page. The canvas is just a container for graphics so we need a language to draw the graphics, we use JavaScript to draw the graphics. We can draw line, rectangle, circle etc using canvas element in HTML5.

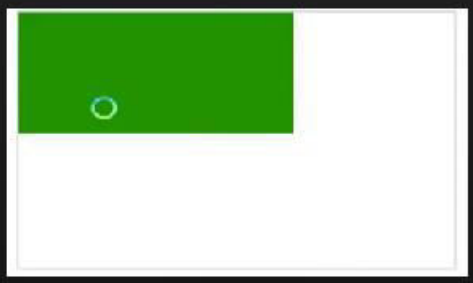
It allows to create dynamic, 2D shapes and bitmap images. There are several methods in canvas to draw paths, boxes, circles, text and add images. For Example:

Note- In canvas we take values of width, height and others in number value not pixcel or px.

We can visit https://dataveyes.com to see the beauty of canvas.

To draw rectangle:-

```
<body>
    <canvas id="myCanvas" width="300" height="200" style="border:1px solid #c3c3c3"></
    <script>
        var canvas=document.getElementById('myCanvas');
        var ctx=canvas.getContext("2d");
        ctx.fillStyle='green';
        ctx.fillRect(0,0,190,95);
        </script>
</body>
</html>
```



Fillstyle='green' to add background color

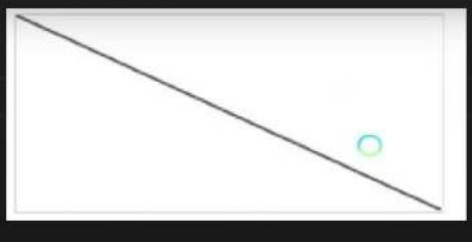
fillRect(x coordinate, y coordinate, width, height) to create rectangular.

getContext('2d') used because HTML canvas is in 2 dimensional.

```
<body>
   <canvas id = "myCanvas" style="width:300; height:300; border:2px solid red;"></canvas>
   <script>
      let canvas = document.getElementById('myCanvas');
      let can1 = canvas.getContext("2d");
      can1.fillStyle ='green';
      can1.fillRect(50,50,200,100);
   </script>
</body>
```

To draw a line

```
body>
    <canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3"></
    <script>
        var canvas=document.getElementById('myCanvas');
        var ctx=canvas.getContext("2d");
      ctx.moveTo(0,0);
      ctx.lineTo(200,100);
       ctx.stroke();
        </script>
/body>
/html>
```



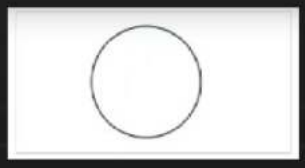
moveTo(x axis, y axis) to start the line x-0, y - 0 means start from top left corner

lineTo(200, 100) to end the line at x- 200 y-200

The above two methods will not show anything and to show we need to use a method stroke()

To draw Circle

```
body>
    <canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3"></
    <script>
        var canvas=document.getElementById('myCanvas');
        var ctx=canvas.getContext("2d");
      ctx.beginPath();
      ctx.arc(95,50,40,0,2*Math.PI);
      ctx.stroke();
        </script>
/body>
```



We need to take beginPath();

arc(x axis, y axis, redius, startAngle, endAngle), we can take arc to create circle and arc. arc is a part of circle.

x coordinate will be the x axis from center of the circle. 95 means x axis se 95px distance

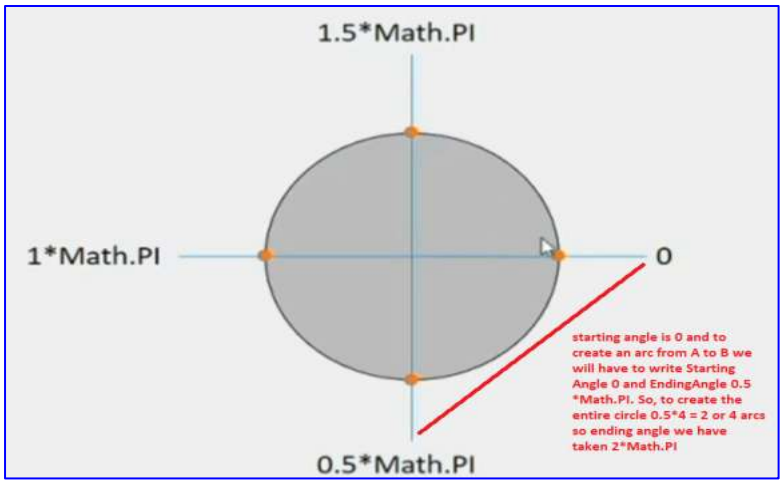
y coordinate will be the y axis from center of the circle , 50 means y axis se 50px distance

radius, how big circle you want to create

stroke() will create a circle but not inside but fill() will create a circle which will be filled with black color

strokeStyle=”red” to change the default color black to red.

fillStyle=”red” to change the default color black to red inside.



45. What is SVG?

SVG stands for Scalable Vector Graphics. It is a format of image and it is vector image which has XML format. The best part of scaleable vector image or SVG is that it is scalable so the quality or resolution will not change no matter how much you zoom in and zoom out. We do have other formats PNG, GIF(animated), JPEG available in the markets but these formats PNG, GIF, JPEG of image are bitmap image and The bitmap images are divided into pixels grade which have predefined height and width. So, image will not look good if more zoom these formats of images. Bitmap images are not scalable and file size is big.

WHY USE SVG

It can be edited by any text editor

SVGs are scalable.(increasing the size using height and width but will make no difference in the appearance)

SVSs can be printed with high quality at any resolution with loosing its quality.

It is zoomable.

It is XML format so can be used in HTML & CSS.

It can be easily searched in the google, indexed like can be indexed in the search engine index, scripted like can be used in javaScritpt easily on any event.

It can also be compressed like other formats

There are two ways to use SVG in HTML

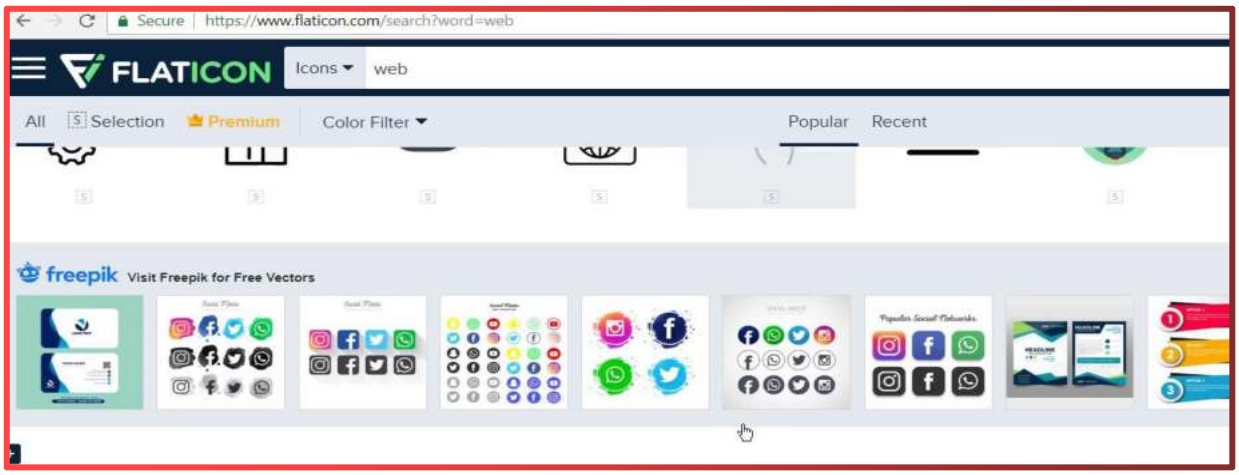
```
<svg width="100" hight="100">
   <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" />
</svg>
```

```
<img src="filename.svg" width="200px " height="200px" />

<svg width="100" height="100">
 <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4 />
</svg>
```

Limitation

We don't use photos in SVG format because it has multi shades and when we convert it into SVG, the file size will increase.
SVG images are flat color which means one color is used in one place and gradients (multi-colors) are not there. Our photos are in multi-colors and images captured from any camera are also in multi-colors or multi-shades so SVG can not be used there. We use JPEG format in photos. We use SVG for logos and graphics.

We can use flaticon.com to get SVG icons and images.



46. What are the different new form element types in HTML 5?

The details tag is used to specify some additional details on the web page. It can be viewed or hidden on demand. The summary tag is used with details tag.

Following is a list of 10 frequently used new elements in HTML 5:

Color Details (for color and details not form elements)

The below is used with input tag like <input type="date/time/emil/range/datetime-local/search/url"

Date Datetime-local Email Number Range Search Time Telephone Url

```
<details>
    <summary>My Details</summary>
    <P> I am Sarfraz</P>
</details>
```

▼ My Details

I am Sarfraz

▶ My Details

```
<p>bdo is used to change the direction of the text from right to left and left to right </p>

<div bdo dir="rtl">
    <p> I am bdo example</p>
</div>
```

I am bdo example

```
<p> Abbriviation will show that ABB is the Abbreviation of " Abbraviation" </p>

<div>
    <p> I am <abbr title="Abbriviaion">ABB</abbr> for Abbraviation</p>

</div>
```

I am ABB for Abbraviation

```
<p> Time is used in input tag that shows the current time when you refresh.</p>
<form>

    Time : <input type="time" value=""></br></br>
```

Time : --:--

```
<p> Date is used in input tag that shows the current date when you refresh.</p>

    Date : <input type="date" value=""></br></br>
```

Date : dd-mm-yyyy

```
<p> DateTime-Local is used in input tag that shows the current time & date when you refresh.</p>

    DateTime-Local : <input type="datetime-local" value=""></br></br>
```

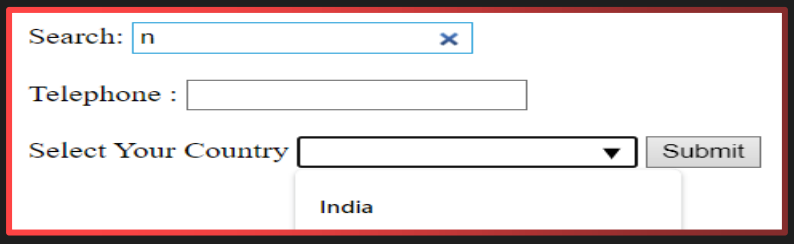
DateTime-Local : dd-mm-yyyy --:--

```
<p> Range is used in input tag that shows the range and used with number only.</p>

    Range: 10<input type="range" value="number" max="30" min="10">30</br></br>
```

Range: 10 30

```
<p> Email is used in input tag that shows the current time when you refresh.</p>

    Email : <input type="email" value=""></br></br>

<p> Time is used in input tag that shows the current time when you refresh.</p>

    Search: <input type="search"></br></br>
```

```
      <p> Time is used in input tag that shows the current time when you refresh.</p>

          Telephone : <input type="tel" value=""></br></br>

   </form>
```



Datalist:- The datalist tag is used to show drop down menu from input tag and to display drop down menu. We will take option tag and attribute value and the id attribute of data-list should be equal to the list attribute of input tag as you can see the value of list and id of input and data-list tag are same.

```
<form>
    <label for="Country">Select Your Country</label>
       <input list="Country">
       <datalist id="Country">
           <option value="India">India</option>
           <option value="Japan">Japan</option>
           <option value="China">China</option>
       </datalist>
       <input type="submit">
 </form>
```

46. Is there any need to change the web browsers to support HTML5?

No. Almost all browsers (updated versions) support HTML 5. For example Chrome, Firefox, Opera, Safari, IE.

47. Which type of video formats are supported by HTML5?

HTML 5 supports three types of video format:

- mp4
- WebM
- OGG

48. Is audio tag supported in HTML 5?

Yes. It is used to add sound or music files on the web page. There are three supported file formats for HTML 5 audio tag.

1. mp3
2. WAV
3. OGG

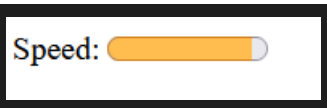
Let's see the code to play mp3 file using HTML audio tag. Without controls, it will not work.

```
   <audio controls>
   <source src="Maher_Zain_Alhamdulillah(256k).mp3" type="audio/mpeg">
   </audio>
```

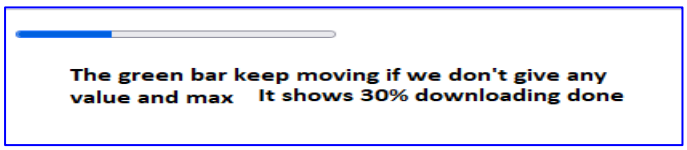
49. What is the difference between progress and meter tag?

```
<body>
   <!-- used for speed in vehicle, temperture of room , pressure
   value is between 0 to 100(can be more than 100 but usually used between 0 to 100) -->
   <meter> </meter>
</body>
```

```
   Speed: <meter min="0" max="100" value="90" low="15" high="60"> </meter>
   <!--
   meter tag is used for speed in vehicle, temperture of room , pressure
   value is between 0 to 100(can be more than 100 but usually used between 0 to 100)
   value="0" means nothing filled value="1" means fully filled, 0=0%, 1=100% if min and max not used
   It will be filled 20% filled because set meter 0 to 100
  low and high are used to change the color of the meter from green to yellow. If low="15" and high="60" then
  if tge value is less than 15 and higher than 60 than it will be yellow else green -->
```




```
   <!-- It is used to show the progress of any video, audio, website downloading.Max is 100 and value 30 means
   30% completed. Need to use JavaScript to increase the percentage as it is downloaded -->
   <progress max="100" value="30"></progress>
```



The progress tag is used to represent the progress of the task only while the meter tag is used to measure data within a given range.

50. What is the use of figure tag in HTML 5?

Figure tag is a tag introduced in HTML 5 which is a container in which we can keep images, illustrators (इलस्ट्रेट) code or diagram because it allows us to define the title of the image or code we use with the help of fig-caption tag used with figure tag so that search engine can search your image by title and display your website in the search bar drop down.

```
<body>
    <figure style="border:1px solid red ; width:fit-content;">
        <img src="My Passport-size image.png" alt="Image" width="100px" height="100px">
        <figcaption style="text-align:center ;">My Image</figcaption>
    </figure>
</body>
```

52. What is button tag?

The button tag is used in HTML 5. It is used to create a clickable button within the HTML form on the web page. It is generally used to create a "submit" or "reset" button. Let's see the code to display the button.

```
<button name="button" type="button">Click Here</button>
```

55. How are tags migrated from HTML4 to HTML5?

No.	Typical HTML4	Typical HTML5

1)		<header>
2)		<nav>
3)		<section>
4)		<article>
5)		<footer>

HTML 4 Header and Footer:

```
<div id="header">
  <h1>Monday Times</h1>
</div>
.
.
<div id="footer">
  <p>&copy; JavaTpoint. All rights reserved.</p>
</div>
```

HTML 5 Header and Footer:

```
<header>
  <h1>Monday Times</h1>
</header>
.
.
<footer>
  <p>© JavaTpoint. All rights reserved.</p>
</footer>
```

HTML 4 Menu:

```
<div id="menu">
  <ul>
    <li>News</li>
    <li>Sports</li>
    <li>Weather</li>
  </ul>
</div>
```

HTML 5 Menu:

```
<nav>
  <ul>
    <li>News</li>
    <li>Sports</li>
    <li>Weather</li>
  </ul>
</nav>
```

56. If I do not put <!DOCTYPE html> will HTML 5 work?

No, the browser will not be able to identify that it is an HTML document and HTML 5 tags do not function properly..

57. What is the use of the required attribute in HTML5?

It forces a user to fill text on the text field or text area before submitting the form. It is used for form validation.

Example:

Name: `<input type="text" name="name" required>`

58. What are the new <input> types for form validation in HTML5?

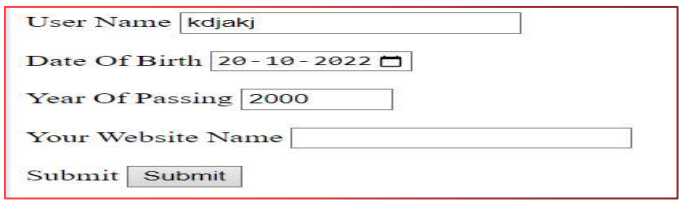
The new input types for form validation are email, URL, number, tel, and date.

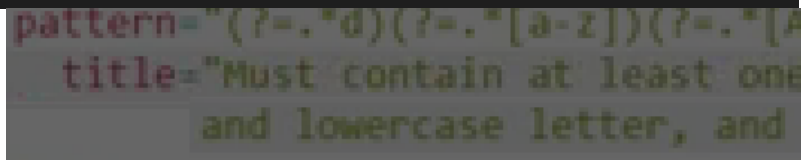
We need to validate the form before deploying or putting it online to ensure it is working fine. We have several validations.
1. User Input Length Checking (maxlegth=”8” minlength= “4”, Here user will have to enter at least 4 and maximum 8 character else throw error)

2. Required to ensure the required field is entered by the user else it will ask to fill if you submit without filling.
3. Max and min with Number and Date type
4. Email and Number to ensure use is entering correct format of email and number, we will use type = Email/ Number
5. Pattern Validation - It is used in type=text/password to set the pattern of text or password by the user.
6. URL validation using type=url to ensure user is entering the correct format of URL.

```
    <!-- max and min with date and number will show date between the given date only -->
```

```
<form action="">
      <label for="">User Name</label> <input type="text" maxlength="8" minlength="4" required> <br><br>
       <label for="">Date Of Birth</label> <input type="date" max="2022-12-01" min="2021-12-01"> <br><br>
       <label for="">Year Of Passing</label> <input type="number" max="2000" min="2010"><br><br>
       <label for="">Your Website Name</label> <input type="url"><br><br>
       <label for="">Submit </label><input type="submit">
    </form>
```





1. What is CSS?

CSS stands for Cascading Style Sheet. It's a style sheet language that determines how the elements/contents in the page are looked/shown. CSS is used to develop a consistent look and feel for all the pages.

The latest version of CSS is CSS3.

CSS was developed and is maintained by the World Wide Web Consortium(kansawtium) (W3C). It was first released on December 17, 1996.

CSS framework.

CSS is a library and its framworks are Bootstrap, Bulma, Skelton(isskeltn), Foundation.

A style sheet is a file on which codes like padding, margin, page size are written in form of code and that help to make the layout beautiful.

2. Use of Rule-set

It is used for the identification of the selectors. There are two types of ruleset

Selector used to style the HTML element

Declaration Block: it contain one or more semicolons

3. Difference between CCS and CSS3

CSS is old version CSS3 is the latest version

CSS3 supports module, media query, all the new browsers but CSS does not.

CSS3 supports gradient, RGBA, HSLA, HSL but CSS support old standard of color.

CSS3 is mobile friendly but CSS is not.

Latest property of CSS3 animation, border-box, background & its properties like background-position, background-repeat, and background-image, test-shadow,box shadow(We have box shadow generator **https://cssgenerator.org/filter-css-generator.html**), **https://www.cssportal.com/css-flexbox-generator/** border-radius, flexbox, transform, transition.

HTML RGB & RGBA COLOURS

RGB → Represents Red, Green & Blue

The intensity of the color with a value between 0 and 255

↓ Lowest ↑ highest

If rgb(0, 255, 0) then color will green as g is the highest

If rgb(0, 0, 0) then " " black

If rgb(255, 255, 255) " . . . white

RGBA

↓

A for tells about opacity about the color

↓

whether fully Transparent if value is 0.0

HEX COLOR VALUES

In HTML, a color can be specied using a hexadecimal value in the form #rrggbb

or HSL → Hue Saturation Lightness

rrggbb: rr = red, gg = green, bb = blue

are hexadecimal values between 00 & ff (Same as decimal 0-255)

Ex- #ff0000 → Will display Red

- ff → Red will be full
- 00 → green will be 0
- 00 → blue will be 0

ff = Maximum Value or Highest Value

#00ff00 → Will display green

#0000ff → " " Blue

#000000 → " " Black

#ffffff → " " White

Shades of Grey

Shddes of grey are often defined using equal values for all three pasameters

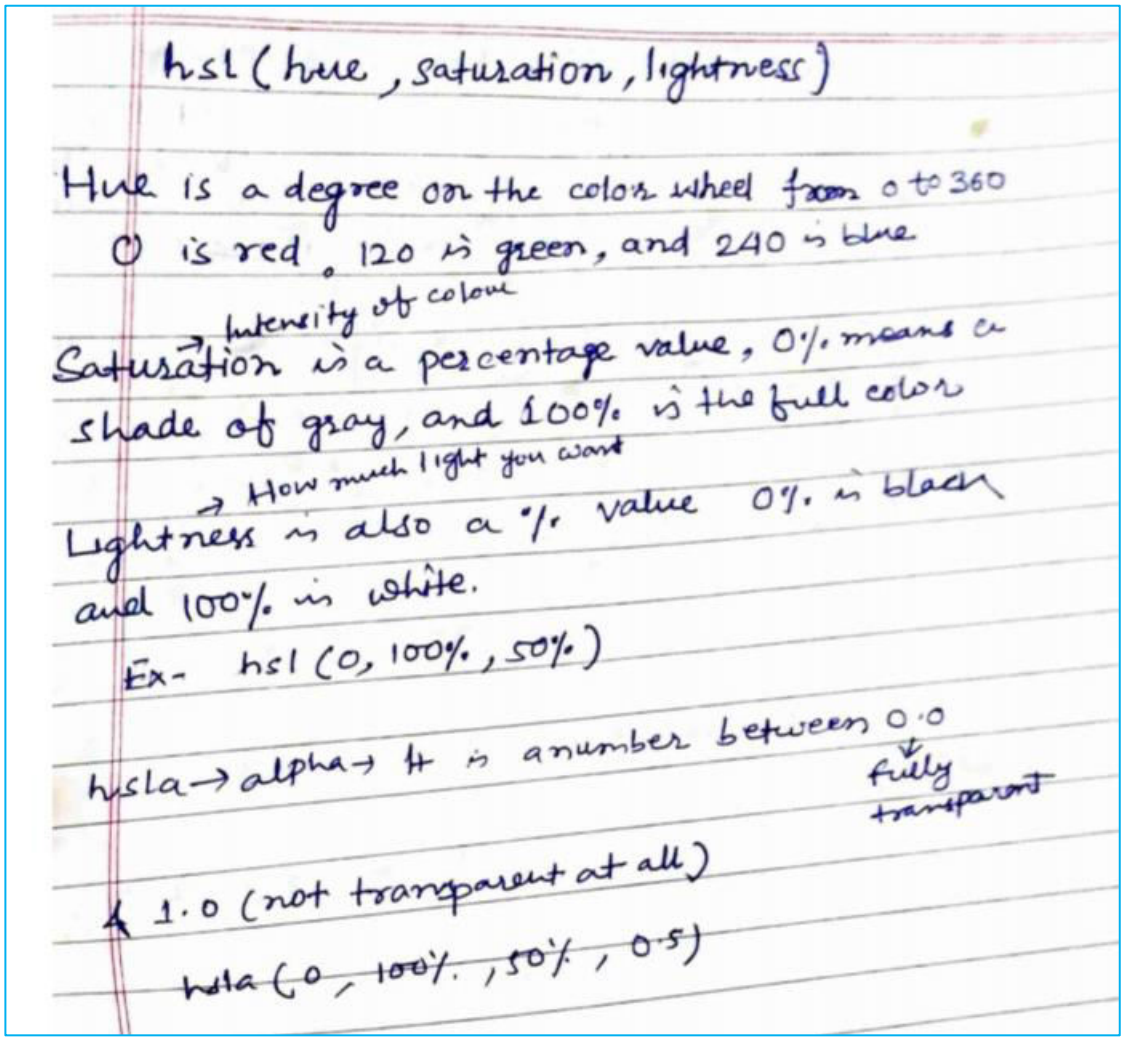
Ex- #404040

#686868

#a0a0a0

#bebebe

#dedede



4. Difference between class and id. (v)

Calss is a way of styling the HTML element. It is not unique but id is unique.

Id can be assigned to a single element but class can assigned to multiple elements.

What is common between class and ID?

Both class and ID are used in HTML **to assign a value** from CSS. The ID is used as an element, whereas the class is used as a block.

5. CSS to control background-repeat

Background-repeat:no-repeat;

6. What is meant by RGB stream?

RGB represents colors in CSS. The three streams are namely Red, Green, and Blue. The intensity of colors is represented using numbers 0 to 256. This allows CSS to have a spectrum of visible colors.

7. What was the purpose of developing CSS?

CSS was developed to define the visual appearances of websites. It allows developers to separate the structure and content of a website that was not possible before.

8. Tell us about the property used for image scroll controlling?

The background-attachment property is used to set whether the background image is fixed or it scrolls with the rest of the page. Example for a fixed background-image:

```
body {
  background-image: url('url_of_image');
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

Background Shortcut

Background: background-color background-image background-repeat/no-repeat/repeat-x/repeat-y/space/round background-positon:left-top/center/bottom or

Right-top/center/bottom or center-top/center/bottom or 40%(left to right) 10%(top to bottom)

OR

Background:blue url('sar.jpeg') no-repeat right-center

Background-size: conver/contain/auto/10px/ 10% 20%(height will be 10% and width will be 20%)

Background-attachment:fixed(It will fix the background-image and text will move)

Scroll(It will scroll both image and text)

Name some font-related CSS attributes.

FONT(svw slf)

Syntax

```
body {
  font: font-style font-variant font-weight font-size/line-heig
}
```

In Use

```
body {
  font: italic small-caps normal 13px/150% Arial, Helvetica, sa
}
```

You need to supply at least font-size and font-family for the shorthand otherwise it'll just be a syntax error and do nothing.

CSS Fonts is a module of CSS that defines font-related properties.

Font-family means which font you want to apply like arial,verdana etc.

font-weight means how bold you do to the font or text.

font-style means make the text italic. It has three values italic, oblique and normal(default or no change)

Italic and oblique both are used to make the text italic. Some of the font-family does not support italic and in that case we should use oblique to support the italic because it will be supported by all the font family

Font-variant means styling the font. It has two values normal(default) and small-caps(It will capitalize all the words and the first word of each paragraph will be bolder than the remaining words)

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. NUNC ELIT DOLOR, ORNARE IN TEMPUS VEL, CURSUS VITAE TURPIS. CURABITUR

line-height means height/space between lines/paragraph or first and second line height or height between first and second paragraph.

font-size has some predefined values like small/smaller as shown in the picture which can be used to small or big the font-size. font size value can be given in px, percentage or em.

font-family:arial; means it will set arial family on the text but it we write it as font-family:arial,Helvetica,verdana; When you open the website, it will check your system if your system has arial font or not if yes then arial family will apply or else Helvetica will if arial is not available but Helvetica is available if both are not available then verdana will apply if available or if all the three fonts are not available browsers default font will apply.

10. Define contextual selectors. /kənˈtɛkstʃʊəl/

We know that whenever we want to send any parcel to my father, Reaz, we need to give the specific or correct address so that it can reach to my father, Reaz else it will go to some other Reaz because there are many people in my state whose name is Reaz. Similarly, we have different elements in the HTML documents and some of the elements may be repeated like h1 tag might be used many times and if we style h1 tag, it will style all the h1 tag available in the document but if we want to style any particular h1 tag then we will

have to use id or class or use parent tag like nav h1(nav ke andar jo h1 hai). We tell this thing to the browser through contextual selectors.

In CSS, contextual selectors allow developers to specify styles of different parts of the document. Styles can be assigned directly to specific HTML tags or create independent classes and assign tags to them.

11. Tell us about the general CSS nomenclature.

Nomeclature (न'मे□न्क्लच(र्)) is way that tells how naming convention (do naam ko milana) is done in css correctly.

Naming things correctly in CSS will make the code easier to read and maintain.

BEM - Block Element Modifier is a methodology, that helps you to achieve **reusable components and code sharing**

Wrong Way	Write Way
.sarfrazAlam{ border:1px solid red ; }	.sarfraz-alam{ border:1px solid red ; }
.some-class{ fontWeight: 10px }	.some-class{ font-weight: 10px }

In Cascading Styling Sheets, codes are written in value and property fashion and has a terminator called semicolon. The entire style is wrapped in curly braces and attached to the selector. This creates a style sheet that can be applied to an HTML page.

12. Is it important to test the web-page in different browsers?

Yes, it is the most crucial thing or the most important trial to do when you design a web-page for the first time and make changes to it. Testing your website periodically in different browsers will help you make every web-page compatible with it as browsers have been going through many updates.

13. When should you use translate () instead of absolute positioning?

Translate is a CSS transform value. On changing opacity or transform, browser reflow or repaint is not triggered. Transform requires the browser to create a GPU layer for elements but using the CPU changes absolutes positioning properties. Translate () is more efficient and provides result quicker. On using translate (), element occupies original space, unlike in changing absolute positioning.

14. What are the advantages of using translate() instead of absolute position?

Translate() does not trigger the DOM reflow because it acts on the compositor. The absolute position triggers the repaint or DOM reflow. So, translate() gives better performance.

15. When does DOM reflow occur? (v)

Reflow is a process of a web browser for re-calculating the position/size of your DOM elements in order to display a page.

Reflow occurs when:

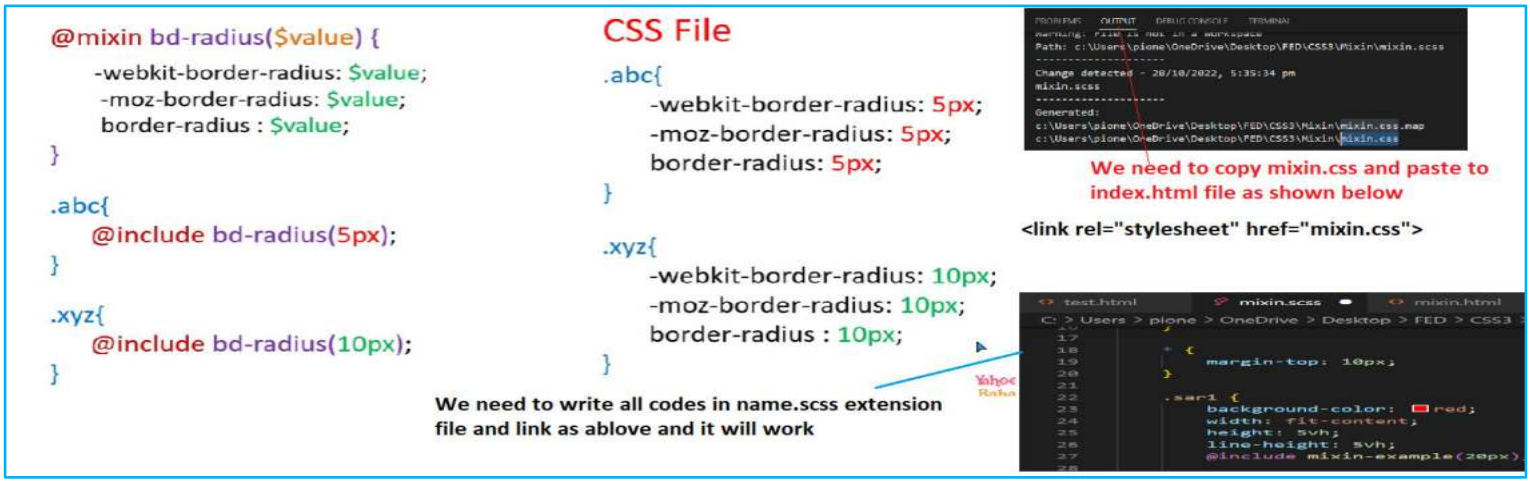
Insert, remove or update an element in the DOM.
Modify content on the page, e.g. the text in an input box.
Move a DOM element.
Animate a DOM element.
Take measurements of an element such as offsetHeight or getComputedStyle.
Change a CSS style.

16. What are mixin?

Mixins is a feature of SASS which works like a function in JavaScript. It is used to avoid the repetition of the code. We may use the same code again and again in CSS. Suppose, we have to use border: 1px solid red; font-size: 10px; border-radius:5px; several times in a web page then we will use mixins. We need to use @mixin name(){ code} and to import @import name();

Our browser can't read SASS file or file with extension .scss. So, the output of this file will not be displayed on the browser. We need to convert sccs file to css file that happens through a compiler. This compiler is available in the VS code as an extension. We need to download the extension called live sass compiler and once it is installed, go to the scss file and click on Watch Sass and copy mixin.css from live Sass compiler from output section.

https://www.youtube.com/watch?v=U8j_rk_RuXA
https://www.youtube.com/watch?v=_XU9PaUuSyo



14. How can you optimize the website to load faster.

Once you have tested the speed of your website, you can start optimizing it.

CDN, BETTER HOST, REDUCE THE SIZE OF IMAGES, REDUCE THE WEBFONTS , REDUCE REDIRECTS AND NUMBER OF PLUGINGS, NUMBER JAVASCRIPT & CSS FILES, USE WEBSITE CACHING,404 ERRORS DETECTION

1. Use a Content Delivery Network (CDN)

We know that, the load time increases when users are physically far from the server. With CDN, user requests are redirected to the nearest server.

As a result, the content is delivered to a user quicker and a website works faster. This is a rather expensive, but quite effective way to optimize the load time.

2. Use better host for the website

There are three possible types of hosting:

Sharing hosting. That's the cheapest way to get your site online in a short time and for a low fee but shared hosting isn't as fast as VPS or a dedicated server because

in sharing hosting CPU, disk space, and RAM are shared with other sites that also use

the same server.

Virtual Private Servers and dedicated servers are much faster. VPS uses multiple servers for content distribution. VPS will be the optimal solution for you If your

website has the average traffic or I have eCommerce site with traffic spikes in some periods,

The most expensive hosting option is to use a **dedicated server** which can be your own physical server. In this case, you pay a server rent and hire a system

administrator to maintain it.

3. Optimize the size of images on your website

Whenever we make any website, we may use lots images specially in commerce website that takes to much data and slower the web-page.

The best way to reduce the image size without compromising its quality is to compress images using tools such as ImageOptim, JPEGmini, or Kraken.

The procedure may take a bit of time but it's worth it. Another way to reduce the image size is to use the HTML responsive images<secret> and <size> attributes that adjust image size based on user display properties.

4. Reduce the number of plugins

We use different plugins to add specific features suggested by third parties. Unfortunately, the more plugins are installed, the more resources are needed to run them. As a result, the website works slower and also security issues can appear. Before using the plugins, we should first run the performance tests on the web-page to find out which plugins are slowing down the website and then try to avoid plugins that load a lot of scripts and styles or generate a lot of database queries. The best solution is to keep only the necessary ones and ensure that they are kept up to date.

5. Minimize the number of JavaScript and CSS files

If any website contains a lot of JavaScript and CSS files, it leads to a large number of HTTP requests when the website visitors want to access particular files. These requests are treated individually by visitor's browser and slow down the website work. If we reduce the number of JavaScript and CSS files, it will load the website quickly. We can also try to keep all JavaScript into one and also do so with all CSS files. This will reduce the overall number of HTTP requests. There are a lot of tools such **as WillPeavy, Script Minifier, or Grunt** to minify HTML, CSS, and JavaScript files quickly.

6. Use website caching (kashing)

We know that when my website is accessed by a lot of users at one time, servers work slowly and need more time to deliver the web page to each user. Caching is the process of storing the current version of the website on the hosting and presenting this version until the website is updated. This means that the web page doesn't render over and over again for each user.

Cached web page doesn't need to send database requests each time.

If the website is developed on WordPress then we can use plugins like: W3 Total Cache or W3 Super Cache.

7. Implement Gzip Compression

Gzip Compression is an effective way to reduce the size of files. It minimizes the HTTP requests and reduces the server response time. Gzip compresses the files before sending them to the browser. On the user side, a browser unzips the files and presents the contents. This method can work with all files on your website. We can enable Gzip on your website by adding some lines of the code or via a utility called gzip.

8. Database optimization in CMS

Database optimization is the an effective way to increase performance. If we use a content management system (CMS) packed with complex plugins, the database size increases and the website works slower. For instance, the WordPress CMS stores comments, blog posts, and other information that take up a lot of data storage. Each CMS requires its own optimization measures and also has a number of specific plugins. For WordPress, for example, you may consider WP-Optimize.

9. Reduce the use of web fonts

Web fonts have become very popular in website design. Unfortunately, the use of web fonts has a negative impact on the speed of page rendering. Web fonts add extra HTTP requests to external resources. The following measures will help you reduce the size of web font traffic:

- Use modern formats WOFF2 for modern browsers;
- Include only those character sets that are used on the site;
- Choose only the needed styles

10. Detect 404 errors

A 404 error means that a “Page isn’t found”. This message is provided by the hosting to browsers or search engines when the accessed content of a page no longer exists. In order to detect and correct a 404 error, you can use error detection tools and plugins. Xenu’s Link Sleuth, Google Webmaster Tools (GWT)

11. Reduce redirects

Website redirects create additional HTTP requests which negatively impact performance. We can use Screaming Frog to quickly identify redirects.

15. How can you optimize the web-page for prints?

Identify and control 'content areas' of the website. A website generally has a footer, header, sidebar, navbar, and main content area. Most of the work is done by controlling the content area. It is better to use the following tricks to optimize the print media.

1. Page breaks
2. Create a style sheet for print
3. Try to figure out the portion of the page you don’t have to print
4. Avoid unnecessary HTML tables.

Refer this link for more details:-

https://dzone.com/articles/optimizing-your-website-struct

16. What is meant by CSS working under the hood or how CSS works internally?

When a browser displays a document, it combines style information and document content. The document is processed in two stages:

Conversion of HTML and CSS into Document Object Model

DOM displays contents of browser

17. Tell us about CSS float property.

The float property of CSS positions an image to the right or left as needed, including text wrapping around it. All properties of elements used before it remain unchanged.

file:///C:/Users/pione/OneDrive/Desktop/FED/CSS3/float_clear.html

18. How media types in CSS work? Or Can you name the four types of @media properties?

The four types of media properties are print, speech, and screen and all. Example of using print-media type:

Media query means writing css for different media like print, screen and speech and all, all means all the three medias.

Print means how the print out should be if print is taken out or writing css for print.

Screen means how should be the web when it is shown in mobile screen,tablet screen like ipad, monitor screen, laptop screen or app screen. So, writing css for different size of screens is called media screen.

Speech that reads the content for the blind.

viewport is the size of the screen and it can be different for different screen.

Commonly used view-ports are 320px, 480px, 760px, 960px, 1200px, 1600px etc.
900 -1680 for laptops
700 to 900 for tabs in px,
320 to 540 px for mobile
@media screen and (max-width:900px){ .container{ width:50%} }
The CSS that you write will be applicable till the screen viewport or width is between 0 to 900px. 900px is called break-point and it can be different for different screen. Here, .container{ width:50%} means if the screen width is between 0 to 900px then container class width will be 50%.

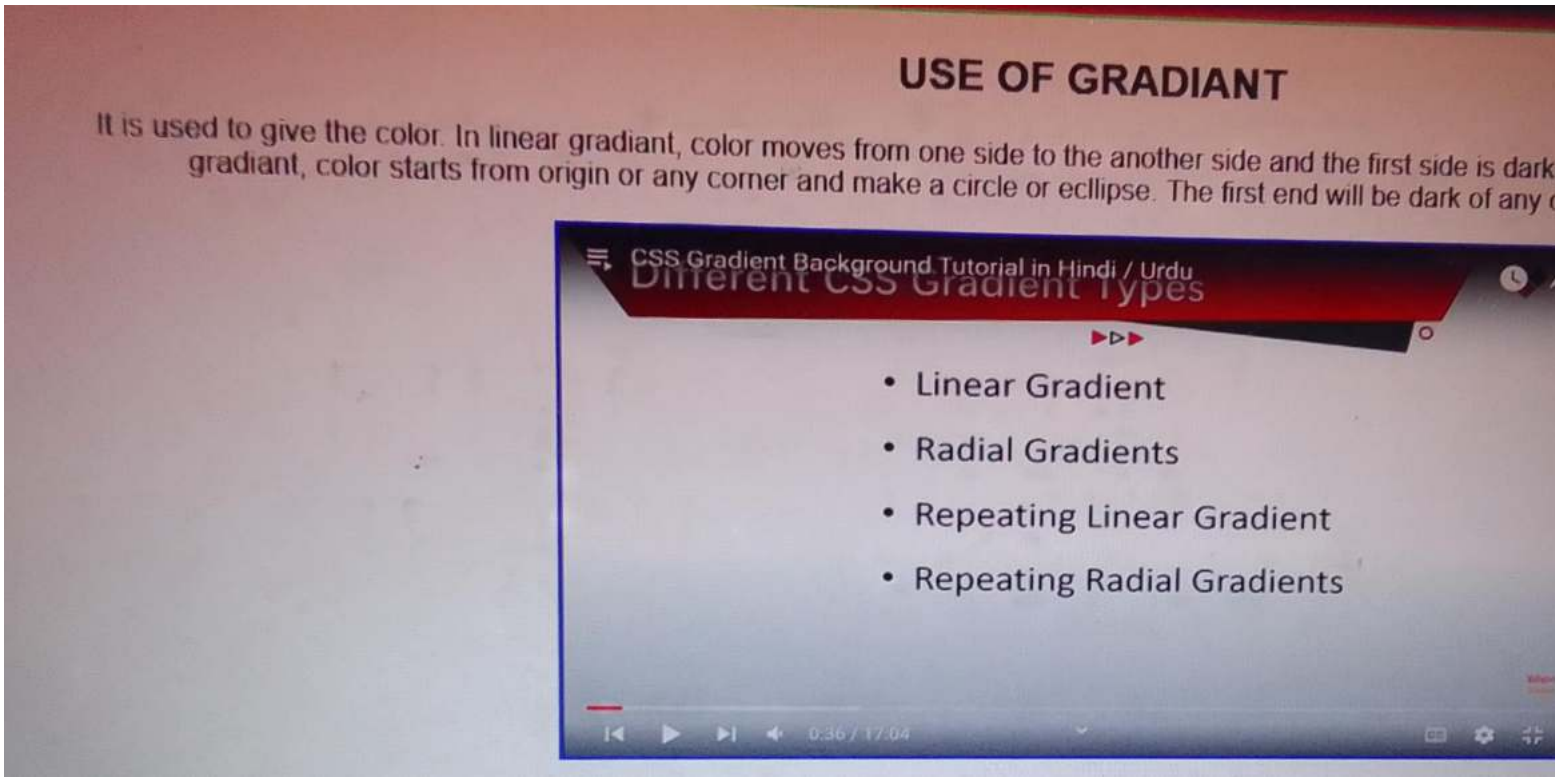
We can write different conditions as well

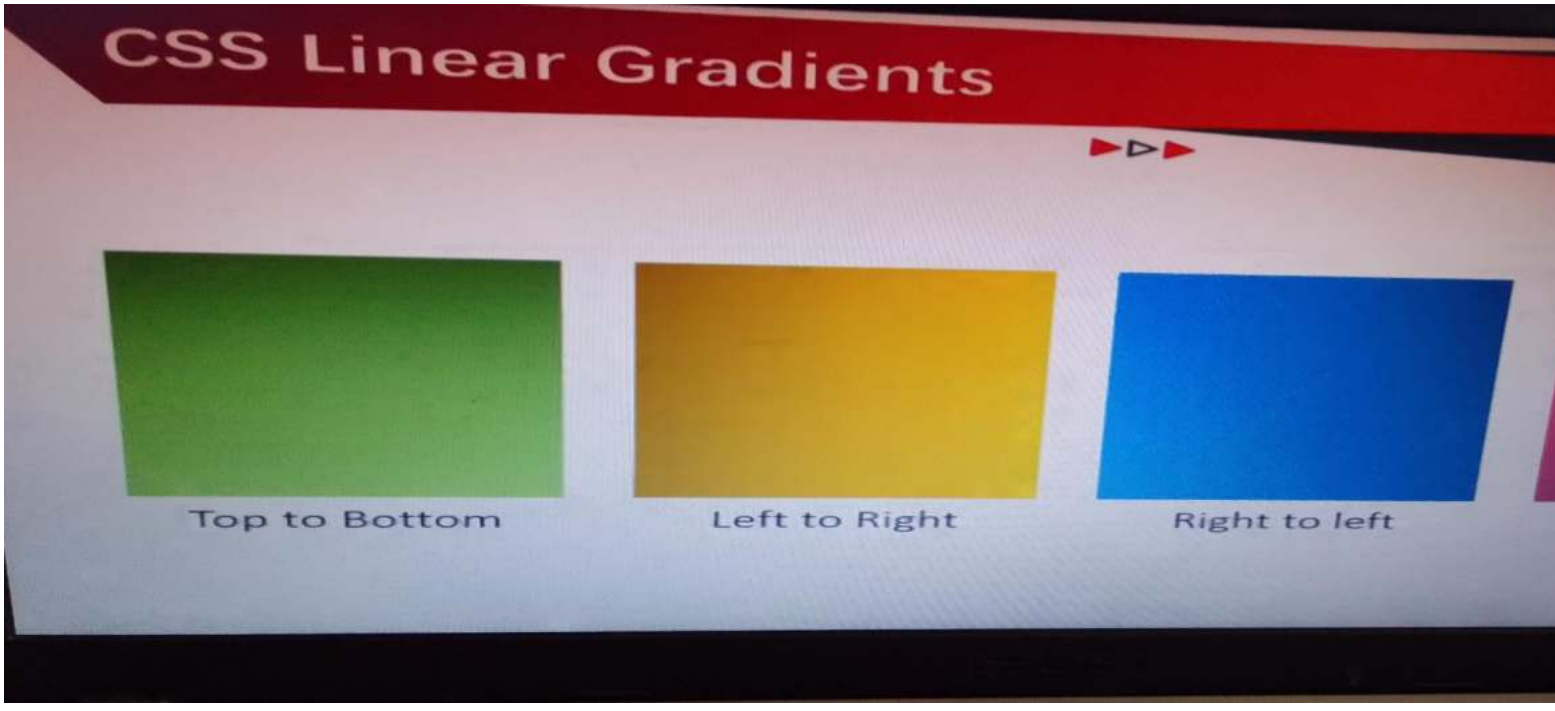
instead of max-with like min-width, min-height etc as shown below. If we want to give two conditions like max-width:200px and min-width:100px then we can use logical operators like “and, not, only” . Please note that we need to use meta viewport under head tag and set internal CSS (like under head not internal) and media query should be written after internal CSS or at the end of the CSS files under head tag then only media query will work properly. To use two conditions, we will write like @media screen and (max-width:800px) and (orientation:landscape) and (min-width:200px){ }

```
    @media   screen and (max-width:800px) {
       body {
          background: pink;
       }
       p img {
          width: 450px;
       }
    }
      OR //@ media used just before the closing head tag under style
    @media screen and (max-width:800px) and (orientation:landcape) and (min-width:200px){
       //code
    }

  </style>
</head>
```

19. Define gradients in CSS.

It is used to apply one or more than one colors in the background. It was introduced in CSS 3.0 . It moves from top to bottom by default in linear gradient.





A property of CSS that allows displaying smooth transformation between two or more specified colors. The types of gradients are linear and radial.

```
<p>Example of linear gradiant with direction</p>
```

```
<div style="background:linear-gradient( to left, red, orange);"></div>
```

to top left is the direction which means color will start from right and go towards left and So, right side will be darker and left side will be lighter. Direction can be given in degrees in plus and minus like 45 deg or -45 deg, In percentage like 45% , in top right etc.

```
<p>Example of repeating linear gradiant</p>

<div style="background:repeating-linear-gradient(red, orange 10%, pink 20%);">

It will repeat the colors with the percentage you give. If we don’t give any percentage, it will not repeat.
</div>
```

Redial gradient creates eclipse by default and it can be change to circle by adding circle in the first parameter.

```
<p>Example of radial gradient with percentage</p>
<div style="background: radial-gradient(circle,red 15% , orange 10%, pink 20%);">
        It will make circle shape with different color value. You can give eclipse
    </div>
```

We have different properties closest side and corner and farthest side and corner with direction like at 50%

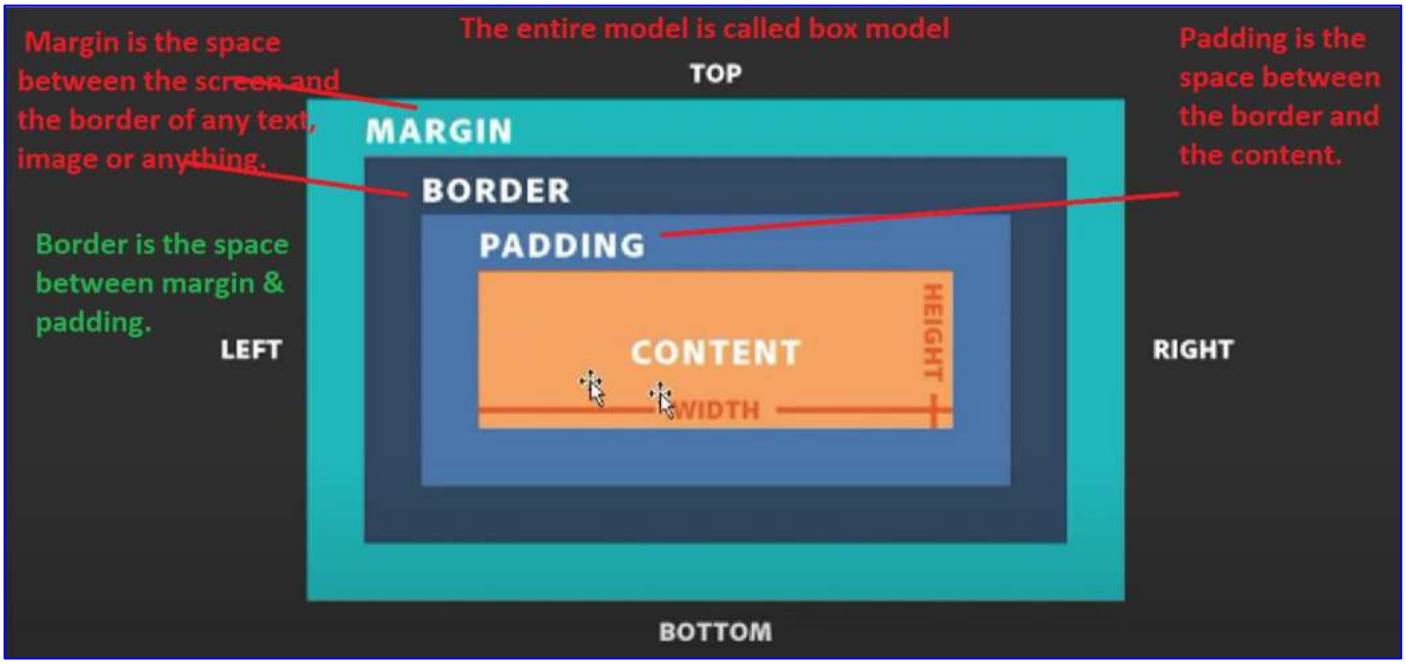
50% which means start from center. X to 50% horizontally and y to 50% vertically. Closest side will create small or shrink eclipse by default and placed at side and farthest side will create big eclipse or spread. Closest corner will make the eclipse small or shrink from the corner if we we set 10% 90% but in farthest corner it will spread from the corner

https://www.youtube.com/watch?v=ib2rSar5rZE

20. What is the Box model in CSS? Which CSS properties are a part of it?

As per WWW(w3c) every HTML element is a box and each element or box has its own properties. Box model is used to understand these properties. In another word, a box model is a combination of content, padding, margin and border. It is used to determine the height and width of the rectangular box. When we give height and width, it is applied to the content.

The CSS Box consists of Width and height of the content, padding, borders & margin.



When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.

- Total element width = width + left padding + right padding + left border + right border + left margin + right margin
- Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

21. How is border-box different from content-box? Or Define border-box/box-sizing

The border-box and content-box are two values of box-sizing.

The box-sizing tells how the width and height of an element are calculated.

In another word, we can say that it will tell whether padding and border should include or not.

We have two properties of border-box.

box-sizing:border-box;

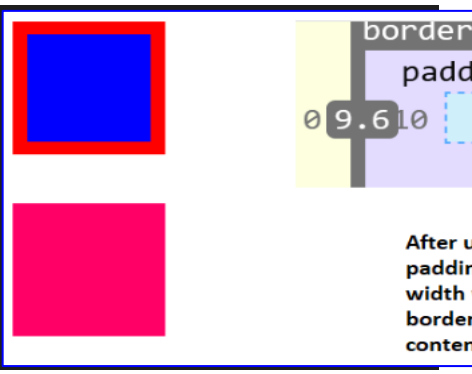
width will not increase. On the contrary, the actual width will be 200(width/content-width) - padding(from right and left) - border(from right and left) If width is 200px and padding is 10px from right and 10px from left and border is 5px from left and 5 px from right.

So, actual width will be 200 - 20 - 10 = 170px

box-sizing:content-box; It is a default property and width will increase.

The size of the width will increase if we take 10px border from left & right and 10px padding from left & right and the total width will be 240px . The margin is not calculated in box-sizing:border-box;

```
<style>
   .one{
      width: 100px;
      height: 100px;
      background-color: blue;
      padding:10px;
      border:10px solid red;
      box-sizing: border-box;
      /* It is including the padding and border size by removing from the content size,
      earlier content was 100px but not it will be 60px */

   }
```



```
   </style>
</head>
<body>
      <div class="one"></div> <br><br>
</body>
```

Padding-box: Width and height values apply to the element's content and its padding. The border is added to the outside of the box. Currently, only Firefox supports the padding-box value. **Tested but not working so ignore it**

23. What are the advantages of using CSS?

The main advantages of CSS are given below:

- Styling the HTML tags
- Responsive website

  We can make our website Responsive which means its width & height will adjust automatically according to the screen size with the help of media query launched in css3. Responsive website helps to change the design according to the screen.

- Animation on web page

  You can make Animation with the help of css3.

- 2D and 3D transformation of HTML elements

  Like you can rotate the heading in 90 degrees launched in CSS3.

- website development process fast

  You can target multiple HTML pages with one CSS file.

24. What are the limitations of CSS?

Disadvantages of CSS are given below:

Browser Compatibility: Some style selectors are supported and some are not. We have to determine which style is supported or not using the @support selector).

Cross Browser issue: Some selectors behave differently in a different browser).

There is no parent selector: Currently, Using CSS, you can't select a parent tag.

25. How to include CSS in the web page? Or Types of CSS.

Ways to implements CSS

- Inline Style

  Inside the HTML tage with style property.

  ```
  <h6 style="border: 2px solid black; background: green; width: 60px;"> I am sarfraz.</h6>
  ```

```
I am sarfraz.
```

Limitation of Inline style page is that pseudo class and elements can not be used in inline style.

- Inpage Stype Page

  It is applied within head

```
<head>

<style>    h6{border: 2px solid black; background: green; width: 60px;"} </style>

</head>
```

- External Style Sheet

  style.css is file name css extension.

```
<link rel="stylesheet" type="text/css" href="style.css">
```

4 - Import a style sheet file (An external file imported into another CSS file): Another way to add CSS is by using the @import rule. This is to add a new CSS file within CSS itself.

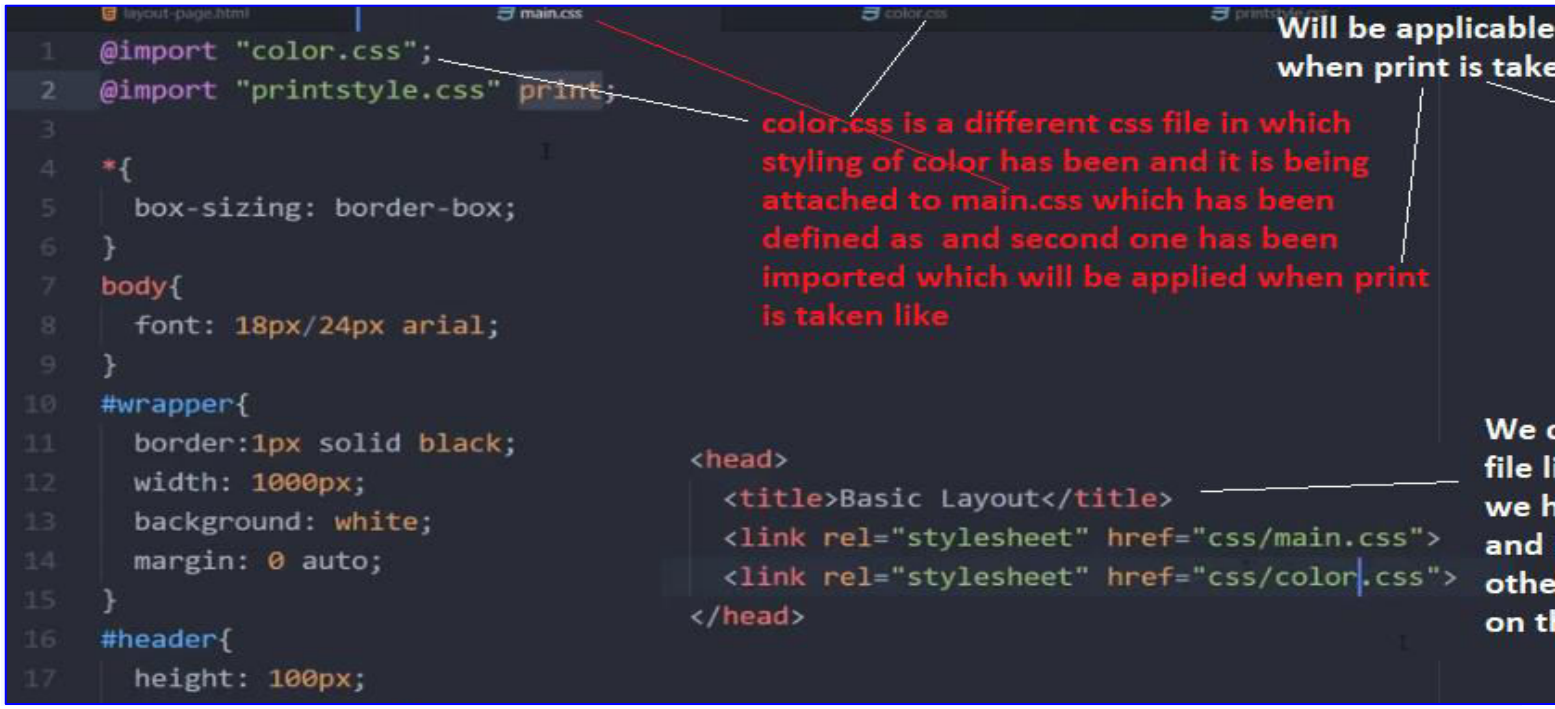
Import rule is used to use one css file to another css file if you have two or more than two css files.

Here, it is not required to make different links for different css file. We will define one link and import another css file into the defined css file or linked css file.

To use the another css file into a defined or linked css file we use **@import** "path/to/style.css"; // This is for relative path

We have second method as well **@import** url("https://www.sarfraz.com/style.css"); // This is for absolute path

Third way @import "printstyle.css" print; //Will apply css of printstyle.css when print is taken out.



26. What is a CSS Preprocessor? What are Sass, Less, and Stylus? Why do people use them?

A CSS Preprocessor is a tool that allows to do almost everything that we can do in JavaScript and other programming language. Like – variables, functions, mixins , for loop, nesting, and some extend of inheritance.

Some of the preprocessors are SASS, LESS AND STYLUS.

SASS: Sass is the acronym (संक्षिप्त रूप) A • kruh • nuhm for "Syntactically /sɪnˈtæk.tɪ kəl.i/ Awesome Style Sheets".

There are two ways to write SASS syntax.

SASS or SCSS

SASS vs SCSS

SASS+
63 is based on indentation and SCSS(Sassy CSS) is not.
SASS uses .sass extension while SCSS uses .scss extension.
SASS doesn't use curly brackets or semicolons. SCSS uses it, just like the CSS.

SASS Syntax

```
$font-color: #fff
$bg-color: #00f

#box
        color: $font-color
        background: $bg-color
```

SCSS Syntax

```
$font-color: #fff;
```

```
$bg-color: #00f;
#box{
        color: $font-color;
        background: $bg-color;
}
```

LESS: LESS stands for “Leaner Style-sheets”. LESS is easy to add to any JavaScript projects by using NPM or less.js file. It uses the extension .less.

LESS syntax is the same as the SCSS with some exceptions. LESS uses @ to define the variables.

```
@font-color: #fff;@bg-color: #00f

#box{
        color: @font-color;
        background: @bg-color;
}
```

Stylus: Stylus offers a great deal of flexibility in writing syntax.

The best part of Stylus is that it is not required to use brackets, colons, and semicolons. It doesn’t use @ or $ for defining variables.

```
/* STYLUS SYNTAX WRITTEN LIKE NATIVE CSS */
font-color= #fff;
bg-color = #00f;
#box {
        color: font-color;
        background: bg-color;
}
/* OR */
/* STYLUS SYNTAX WITHOUT CURLY BRACES */
font-color= #fff;
bg-color = #00f;
#box
        color: font-color;
        background: bg-color;
```

26. What is the difference between CSS variables and preprocessor(SASS, LESS, Stylus) variables?

CSS variables can be used without the need for a preprocessor. Currently, all the major browsers support the CSS variables.

CSS variable cascade but the preprocessor variables don’t cascade.

CSS variable can be accessed and manipulated in JavaScript.

27. What is VH/VW (viewport height/ viewport width) in CSS? OR How do you specify units in the CSS?. What are the different ways to do it?

It tells us about the ways to give height and width .
There are two types of units.
Absolute Unit which is fixed like pixel, length, inch etc.
Relative Unit which changes according to the main/parent div like % and the width will be responsive depending on screen size or parent div/container size. percentage is relative to the parent div but rem and em are relative to the browsers/viewport. em and rem(relative em) are used to give font size. 1em=parent font size. 2em=2 x parent font size 0.5=half of parent font size. 1em is usually equal to 16px and it will convert the font size into 16px if parent font size is not defined. rem is also like em but it is not dependant on the parent div on the contrary it depends on root tag size like HTML because it is a root tag so need to write html{font-size:10px;} under style then 1rem=10px or browsers font-size that can be checked from settings.
We use rem and em to make the font size responsive. vw(viewport width & vh(viewport height) work according to the viewport width and viewport height. if width:100vw and height:100vh; then the width and height will be according the screen viewport width and height. if width:50vw and height:50vh;then the width and height will be half or 50% of the screen viewport width and height if width:50vmax; then viewport mein jiska width jyada hoga(maximum) (screen ko chhota bada karne par) hoga usi ke 50% div ho jayega. if width:50vmax; then viewport mein jiska width jyada hoga(manimum) hoga usi ke 50% div ho jayega.

VW & VH

It's a CSS unit used to measure the height and width in percentage with respect to the viewport. It is used mainly in responsive design techniques. The measure VH is equal to 1/100 of the height of the viewport. If the height of the browser is 1000px, 1vh is equal to 10px. Similarly, if the width is 1000px, then 1 vw is equal to 10px.

There are different ways to specify units in CSS like px, em, pt, percentage (%). px(Pixel) gives fine-grained control and maintains alignment because 1 px or multiple of 1 px is guaranteed to look sharp. px is not cascade. em maintains relative size. you can have responsive fonts. Em, will cascade 1em is equal to the current font-size of the element or the browser default. If u sent font-size to 16px then 1em = 16px. The common practice is to set default body font-size to 62.5% (equal to 10px).

pt(point) are traditionally used in print. 1pt = 1/72 inch and it is a fixed-size unit.

%(percentage) sets font-size relative to the font size of the body. Hence, you have to set the font-size of the body to a reasonable size.

28. Difference between reset vs normalize CSS?. How do they differ?

Whenever we make any website and use HTML elements like head, paragraph and others and when we see the result without applying the css3/css. It may look different in different browsers as it may take default padding, margins or font-sizes of the browser and it may be different in different browsers. The output of the HTML element in which we don't apply any style may look different as if takes 5px padding in browser, it is also possible that it will take 7px padding in any other browser. So, output may be different in different browser as browser takes its default padding, margin or font-size is called **User Agent Style-sheet.**

In another word, The default css that we get from different browsers called **User Agent Style-sheet.**

But, when we reset the padding, font-size and margin to zero using *{padding:0, margin:0} then padding and margin will be set to 0 for the entire web-page and it is called **resetting the CSS.**

Reset is the process to reset the HTML elements to ensure it looks alike in different browsers. Here, in all the browsers, padding and margin will be zero.

The latest way of reset is normalize CSS. It is an alternate of CSS reset of HTML5 called normalize CSS. It only targets the style that needs to be normalized.

We can visit normalize.css to download the link and link in HTML file.

Normalize CSS helps to display the content similar in all the browsers. It also corrects bugs for common browser dependencies.

29. What is the difference between inline, inline-block, and block?

Block Element: The block elements always start on a new line. They will also take space for an entire row or width. List of block elements are , <p>, <ul>,

Inline Elements: Inline elements don't start on a new line, they appear on the same line as the content and tags beside them. Inline elements are text tags or text properties like, <bold>, <heading>, <i>, <a>, , , and <img> tags. These are inline elements and align horizontally and width, margin, padding will not work properly so to use margin, padding, width and height we need to make them block by using display: inline-block; it will become block in one line.

To show these in different lines we need to use span{display:block}. Now, span will work like block element.

We use display: inline; to align all the block element in one line p{display:block;} **Inline Block Elements:** Inline-block elements are similar to inline elements. Inline elements will become inline block elements when we add padding and margins, height and width values to them.

https://www.youtube.com/watch?v=kIASoZjmCG0&t=441s

30. Is it important to test the web page in different browsers?

It's most important to test a website in different browsers when you're first designing it, or when making major changes. However, it's also important to repeat these tests periodically, since browsers go through a lot of updates and changes.

We may test it using inspect

30. What are Pseudo elements and Pseudo classes?

Pseudo-elements allows us to create items that do not normally exist in the document tree, for example ::after.

::before
::after
::placeholder

::first-line
::selection
:: marker

In the below example, the color will appear only on the first line of the paragraph.

```
p: :first-line {
        color: #ff0000;
        font-variant: small-caps;
}
```

Pseudo-classes select regular elements but under certain conditions like when the user is hovering over the link.

:link
:visited
:hover
:active
:focus

Example of the pseudo-class, In the below example, the color applies to the anchor tag when it's hovered.

```
/* mouse over link */a:hover {color: #FF00FF;}
```

31. Does margin-top or margin-bottom have an effect on inline elements?

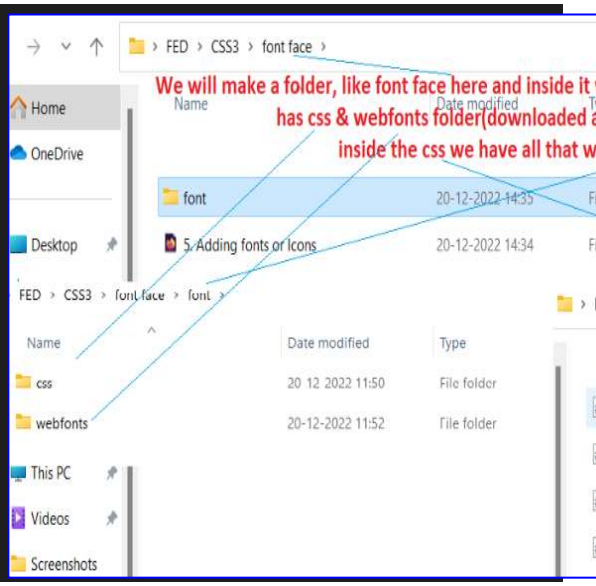
No, it doesn't affect the inline elements. Inline elements flow with the contents of the page.

32. What property is used for changing the font face? Or Define Font Or how to apply different icon on the website.

First, we need to visit the websit Click to visit the site
then click on Free For Web, it will download the zip file. Now, copy(Ctrl+c) it and paste inside your folder
then right click on it to extract here and then click on css and then copy all.css written as all and then
paste it to your own folder under css and also paste the webfonts under css.
We need to link the all.css as <link rel="stylesheet" type="text/css" href="awesome-font/all.css">
To set the icon, visit the link https://fontawesome.com/download and then click on icons and then seach
the icon you want and click on it and then copy the link and paste where you want to add.
The highlited icon is the free one that you can use.

```
<head>
    <link rel="stylesheet" type="text/css" href="font/css/all.css">
</head>
<body>
    <nav>
        <ul>
            <li><i class="fa-solid fa-magnifying-glass"></i>Search Me</li>
            <li>
                <a href="#" id="active"style="background-color: pink;">
                    <i class="fa-solid fa-house"></i> Home
                </a>
            </li>
        </ul>
    </nav>
```

- Search Me
- Home
- Contact
- About Us



We use font-family for changing the font face.
Font-family means which font you want to apply or want to see an an output of text on your web-browser. like arial,verdana etc.
font-family:arial; means it will set arial family on the text but it we write it as font-family:arial,Helvetica,verdana;
When you open the website, it will check your system if your system has Arial font if yes then Arial family will apply or else Helvetica will apply. If arial is not available but Helvetica is available if both are not available then verdana will apply if available or if all the three fonts are not available browser's default font will apply.

We can set font-family for body, header, footer, side-bar differently if we want.

```
<style>
    body{ font-family:arial,Helvetica,verdana; }
    footer{ sens-sarif}
    p{font-family: "Times New Roman", Times, serif;}
</style>
```

FONT

font shorthand

font:font-style font-variant font-weight font-size line-height font-family;

Font-style means make the text italic.
Font-variant means coloring the font.
Font-weight means how bold you do to the font or text.
p{font-weight:bold/bolder}
Bold: To bold or 700
Normal without bold or 400
Bolder will bolder the parent div font-weight if set as bold. Parent div se thoda bada. Or 900
Lighter also works in respect to the parent, it will little lighter from parent if set bold. Or less than or equal to 300
We have values in number as well from 100 to 900

Font-size is used to show how big you want to show the font.font-size has some predefined values like Xx-small, X-small, Small, Smaller, Medium, Large, X-large, Xx-large
small/smaller as shown in the picture which can be used to small or big the font-size.
font size value can be given in px, percentage or em.
P{font-size:15px}
P{font-size:100%} 100% is the parent div px Like if parent div has 20px then p will have 20px=100%

P{font-size:1em} 1em=16px if not assigned in the parent div.

Font-family means which font you want to apple like arial,verdana etc.
line-height means height/space between lines/paragraph or first and second line height.
P{line-height:10px/1em}

Use Of @font-face Rule

https://www.youtube.com/watch?v=5IniEmNymLk

It is used to download and set any font size so that you can see the same font size as others can see. When you use this rule, it makes the web slow as request to server to display the same font. We can use to two website to download the font. download the font in which you can see the web-font kit as it supports all the format.
https://www.fontsquirrel.com/tools/webfont-generator

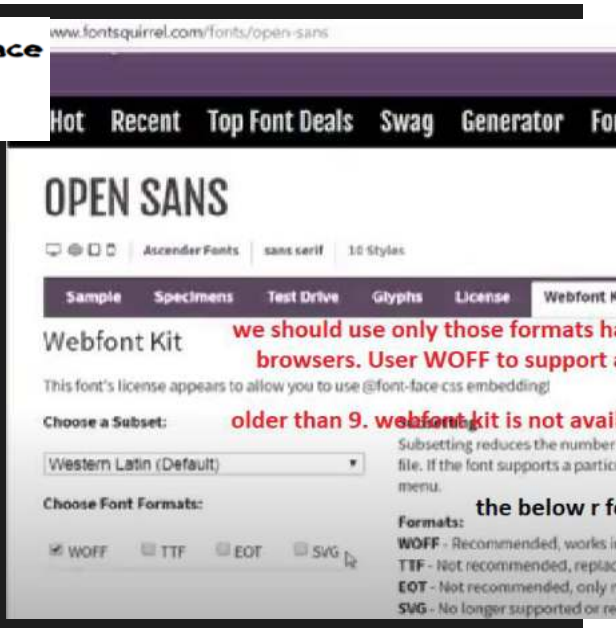
```
 <style>
        @font-face {
            font-family:sarfraz;
            src:url(RyujinAttack.otf) ;
        }
  /* It is better to use the same font-family name so that you can know
which font family is used.
 here we can write "RyujinAttack" the actual font family instead of sarfraz */
        h1,p{
            font-family:sarfraz ;
        }
        .not{
            font-family:serif;
        }
    </style>
</head>
<body>
    <h1>This is the Practical of @font-face</h1>
    <p>My favourite font family applied</p>
    <p class="not">My favourite font family applied</p>
</body>
```

OUTPUT

This is the Practical of @font-face

My favourite font family applied

My favourite font family applied



https://fontawesome.com/

.Use Of GOOGLE FONT

We can download the font family from google font as well and use with@font-face{} as show below:-
https://fonts.google.com/

```
@font-face {
            font-family:sarfraz;
            /* src:url(RyujinAttack.otf) ; */
            src:url(RubikGemstones-Regular.ttf) ;
        }
```

It is used to reduce the loading time. If we use google font, it will not take much time. To use this, we need to go to

https://fonts.google.com/

33. What are the differences between adaptive design and responsive design?

Adaptive Design	Responsive Design
The Adaptive Design loads the content of the web-page according to device screen which was already designed.	The Responsive Design adjust its content and width according to the device.
In Adaptive Design, a developer creates different layouts for different size of screen.	In Responsive Design, a developer creates one layout for different size of screen.
There are six layouts which are created for different size of devices like mobile screen, desptop, laptop, and many more in Adaptive Design such as 320px, 480px, 760px, 960px, 1200px and 1600px	We use @media queries to make the web-page responsive.
A developer will have to work more to create six different layout/pages.	A developer will have to work less because he needs to create a single layout.
If any new layout of the screen comes into the market, the designer will have create a new page/layout for that.	If any new layout of the screen comes into the market, the content is adjusted according to the screen in the responsive design.
Re-sizing of browser window has no affect on the design.	Re-sizing of browser window has affect on the design.
Gives a lot of control over the design to develop sites for specific screens.	No much control over the design is offered here.

34.How are the CSS selectors matched against the elements by the browser?

The CSS selectors matched against the elements by the browser in the order of right to left. The browser finds the element available in the DOM on the basis of key selectors(id, class, tag name) and then reaches to the parent for getting the matches.

Here is an example:

```
p span{
color: black;
}
```

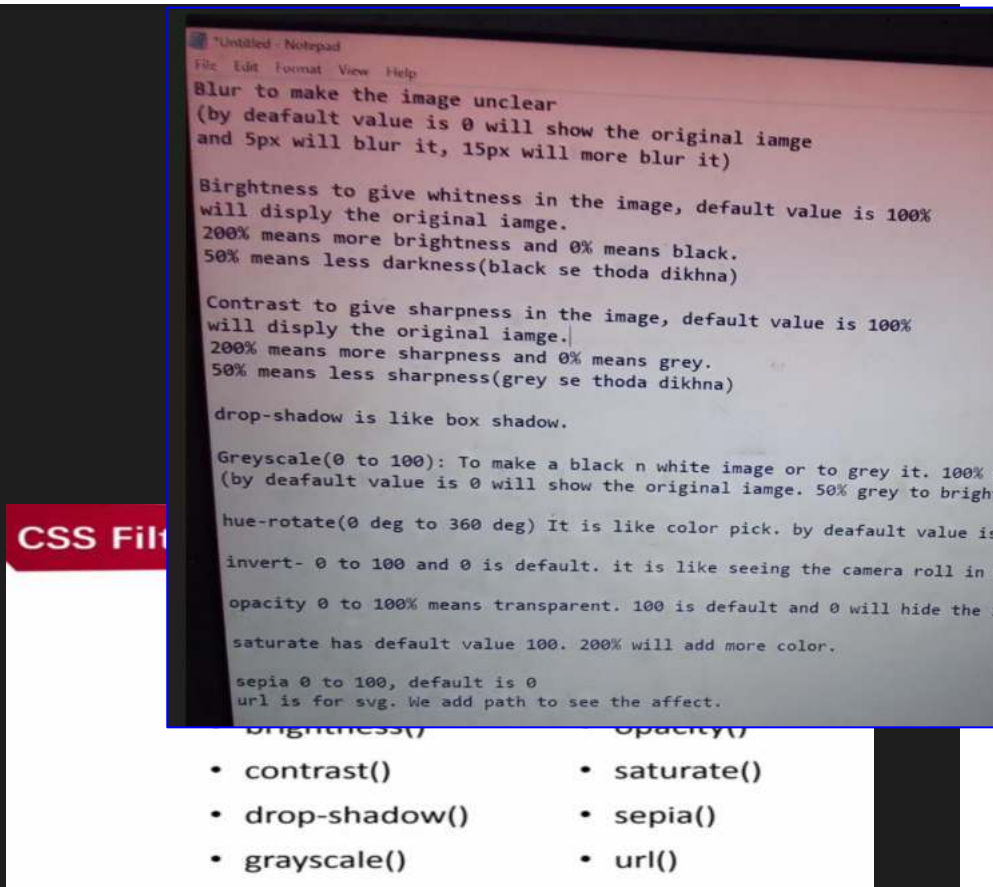
Here, the browser first finds all *span* elements in the DOM.

It will then check all the parents of span and when it finds the parent as an paragraph, it will apply the black color and then matching process will stop.

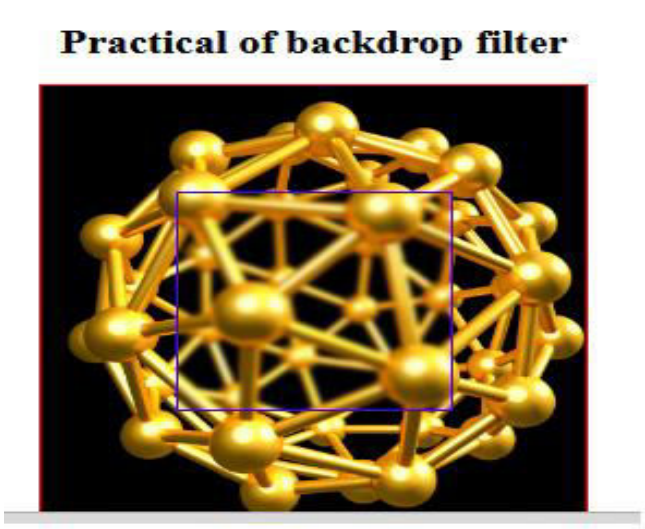
35. How is opacity specified in CSS3? OR FILTER VALUE

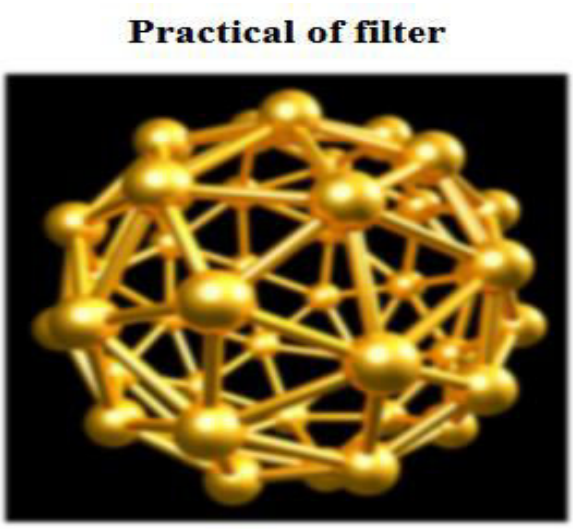
Filter is used to change the color of any image or gives affect on the image but backdrop-filter affects the area of the image only, with the following values as given below. It was introduced in CSS3.

```
    <title>filter & backdrop filter </title>
    <style>
        img[name='sar'] {
            filter: blur(2px);
        }
        .parent {
            background-image: url(image3.jpg);
            background-repeat: no-repeat;
            width: 400px;
            height: 400px;
            border: 1px solid red;
        }
        .child {
            width: 200px;
            height: 200px;
            border: 3px solid rgb(72, 0, 255);
            backdrop-filter: blur(2px);
        }
        main {
            display: flex;
            align-items: center;
            justify-content: center;
        }
        h1 {
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Practical of filter</h1>
    <main>
        <img src="image3.jpg" alt="image" name="sar">
    </main>
    <h1>Without blur</h1>
    <main>
```



```
        <img src="image3.jpg" alt="image">
    </main>
    <h1>Practical of backdrop filter </h1>
    <div style="display: flex; justify-content: center; align-items: center;">
        <main class="parent">
            <div class="child"></div>
        </main>
    </div>
</body>
```





none.It is default value.

blur(0) It is used to blur the picture.Its default value is 0 if blur(0) it will appear as the image is if (2px) it will blur written as blur(4px)

brightness & contrast: Its default value is 100 if brightness(100) it will appear as the image is if (102) it will be brighter and less than 100 darker, 0 means black. Its value is given in % as well.

drop-shadow()
It is just like shadow. 8px for left with plus value, 1px for bottom with plus vale, 0 is blue, color.

grayscale : It is used to make the image black and white. Its default value is 0 if grayscale(0) it will appear as the image and if (0 to 100) it will be getting balck and white.100% will be totally black and white.

hue-rotate : It is used to change the color between 0 to 360 degree. 360 is default

invert : It is used to change the color between 0 to 100%. 0 is default

opacity : It is used to make the image transparent.between 0 to 100% or 0 to 1. 100% or 1 is default which means opaque(no transparent at all).

0.5 means half transparent.

We can use the opacity property as follows:

```
div { opacity: 0.6;}
```

In the above example, an opacity of 60% is applied to the div section. The opacity property is not supported by the internet explorer browser. To make it work there, we need to use filter property as polyfill as shown in the example below.

```
div { opacity: 0.6;
    filter: alpha(opacity=60);}
```

36. Why should we use float property in CSS?

The float property is used for positioning the HTML elements horizontally either towards the left or right of the container

In another word, It is used to make move the div or any HTML tag of the container on the right and left hand side.
If we set float:left on both the divs , they will be on the left hand side attached and similar for the float right.

It has three values:-

```
h2{float: right; }
h2{float: left; }
h2{float: none; }
```

CLEAR

It is used when we use float property to remove the div or any elements which go behind the div on which float property was used.

Clear:left; ka matlab hai mujhe screen ke left mein kuchh bhi nahi chahiye. Example agar left mein pehle se ko cheez hai A div hai to agar B div mein clear:left lagayege to div A nahi hatega par Div B uske niche left mein aa jayega. Agar A div nahi hota to Div B, Div A ke jagah par hota..

37. What is a z-index, how does it function?

It is used to bring the div onto the top if z-index:1; and onto the bottom if if z-index:-1;

The default value of this property is 0 and can be either positive or negative. Apart from 0, the values of the z-index can be:

- Auto: The stack order will be set equal to the parent.
- Number: The number can be positive or negative. It defines the stack order.
- Inherit: It will take the property of parent which means if parent has z-index:1 , the same will be applied in it.

The elements having a lesser value of z-index is stacked lower than the ones with a higher z-index.

```
<style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;}
        #sarf {
            position: relative;
            border: 1px solid black;
            width: 500px;
            height: 500px;}
        div{
            width:200px;
            height:200px;
            position: absolute;
```

```
            border: 1px solid #000;
            font-size:25px;
            text-align: left }
    </style>
</head>
<body>
    <div id="sarf">
        <div style=" background:orange; top:100px; left:50px;">A</div>
        <div style="background:red; top:150px; left:100px;z-index:1;">B</div>
        <div style="background: yellow;top:200px; left:150px;z-index:2;">C</div>
        <div style="background: green;top:250px; left:200px;">D</div>
    </div>
</body>
```

38. What is cascading in CSS or CSS is called cascading style sheet?

Component of CSS Style. Selector , Property & Values

CSS is a cascading tool.

“Cascading” is set of rules that tells which style will be applied first and then displayed on the browser. The rule has been defined/made that which one will be applied first and last.

We know that there are three ways to apply CSS such as inline, in-page or external file.

The cascading tells which one style will be displayed on the browser first if we use these three ways.

So, we have certain priority of the rules of the styling called cascading. Here are the some rules of the style sheets.

1. Browser has zero priority which means browser’s default style will apply if no style is applied and the output will be decided by the browser you are using.
2. Tag/Element has one priority which means when you apply some styling to the tags, elements, that style will be applied.
3. If we apply any style in parent element then the same style will be applied in the child/children as well. This is rule of cascading style sheet.
4. If we apply same style multiple times to one element/tag then the last one will be applied.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>cascading</title>
    <style>
        p{  font-size:30px;
            font-size:40px;
        }
        p{  color:red;
            color:blue;

        }
    </style>
</head>
```

```
<body>
      <p> I am Sarfarz</p>
</body>
</html>
```

Here, the font-size:40px; and color:blue; will be applied.

Here is the priority level:-

Tag>Class>Id>Inline

Inline has the highest priority, Tag has the lowest priority.

We have also freedom to change this priority level using important hack written as !important as shown below to give the highest priority.

```
p{color:white !important;}
```

39. What is specificity? How to calculate specificity? Or If one element is styled by more than one rule, which style will be applicable?

Specificity is a rule that tell which style will be applicable if one element is styled by more than one rules.

More specific rule(more detailed information) will get more value but if inline style is applied then inline style will be applied and if !important is applied then it will be applied.

First Priority is of !important

2nd Priority is of inline

3rd Priority is of External will work according to specificity. (Jiska jyada details ho like if I tell him that give this pen to Sarfraz or give this pen to Sarfraz whose father is Reaz or give this pen to Sarfraz whose father is Reaz and lives at Katras then the last one will be more detailed or specific.

Specificity Rule

10,000 point to !important

1000 point to inline

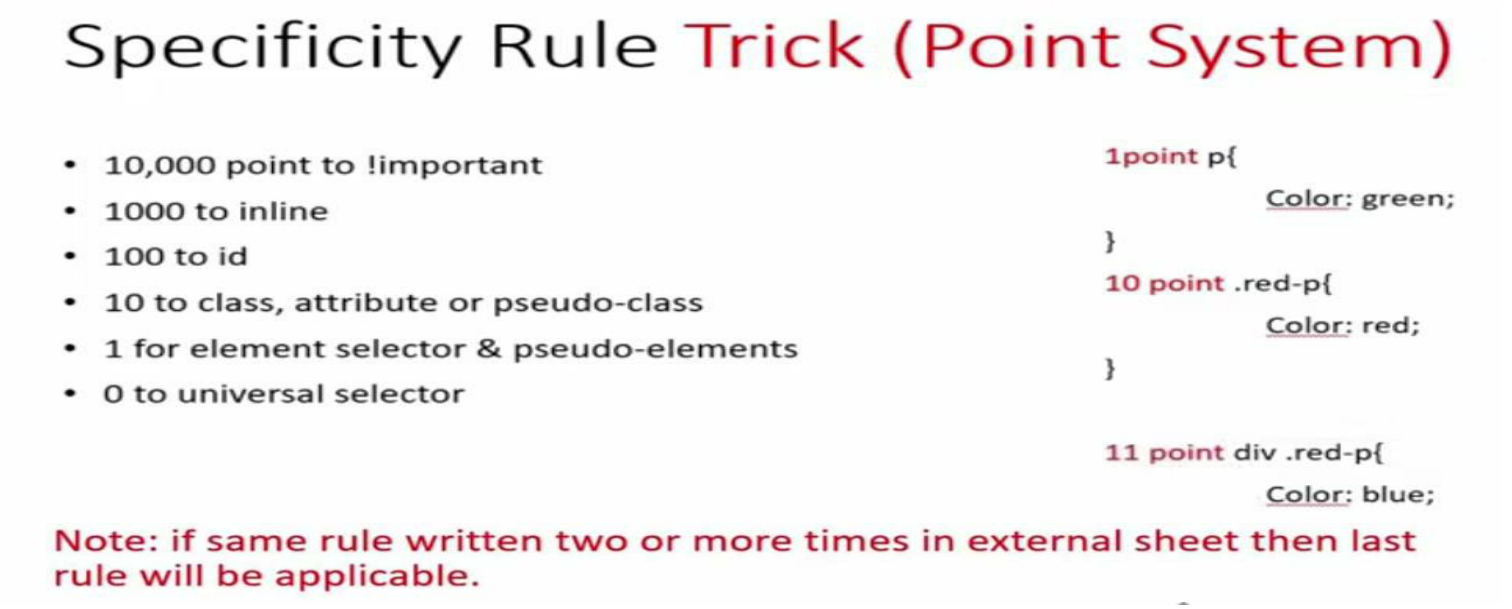
100 point id

10 point to class, attribute or pseudo-class

1 point for element selector & pseudo-elements

0 point to universal selector

A process of determining which CSS rule will be applied to an element first if more than one style is applied in the same element.



40. How do I restore the default value of a property? Or Explain:- initial, inherit, unset and revert.

https://www.youtube.com/watch?v=N8tFrMZp_wA

The keyword initial can be used to reset or restore its default value.

We know that every property has its default value like paragraph has default color "black" which will appear on the browser when we don't reset/give any color to paragraph tag.

```
<body>
    <div style="color: red;">
        <p style="color: initial;">I am sarfraz</p>
        <p style="color: inherit;">I am sarfraz</p>
    </div>
</body>
```

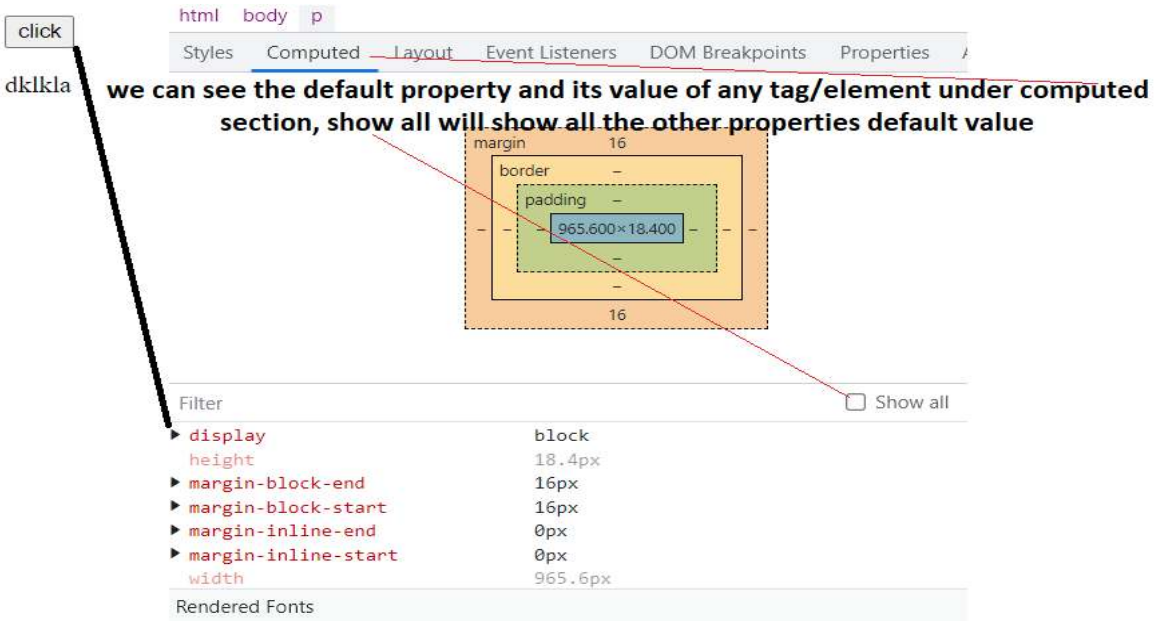
I am sarfraz

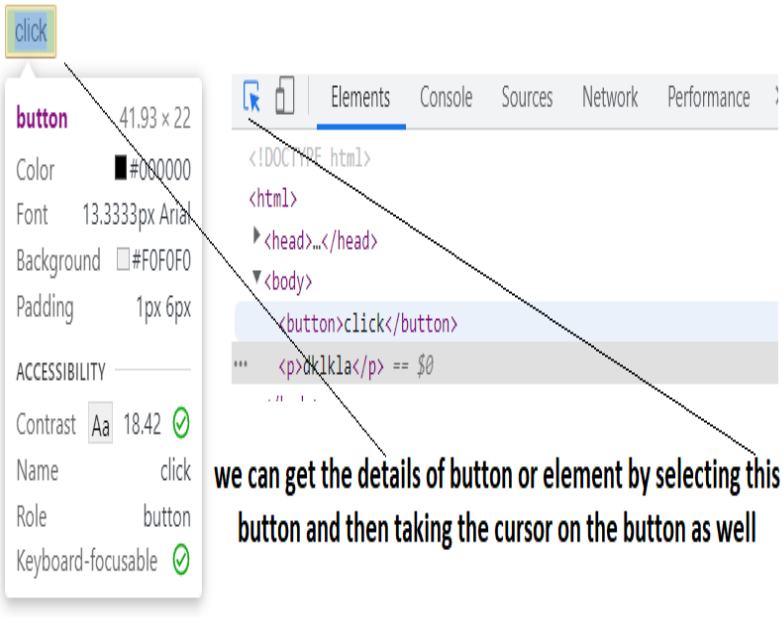
I am sarfraz

```
j // It is black because the default color of paragraph is black, initial returns the default color
/ It is red because the parent color of paragraph is red, inherit returns the parent property value
```

Similarly the default font-size of any paragraph is 16px which will apply if we don't give any font-size. So, if we want to know the default value of any property then we can use initial. For example p{color:initial},

it will set black color in the paragraph because initial return the default value. Similarly, p{color:inherit} will give the color given in parent of paragraph tag if any else return the default one.

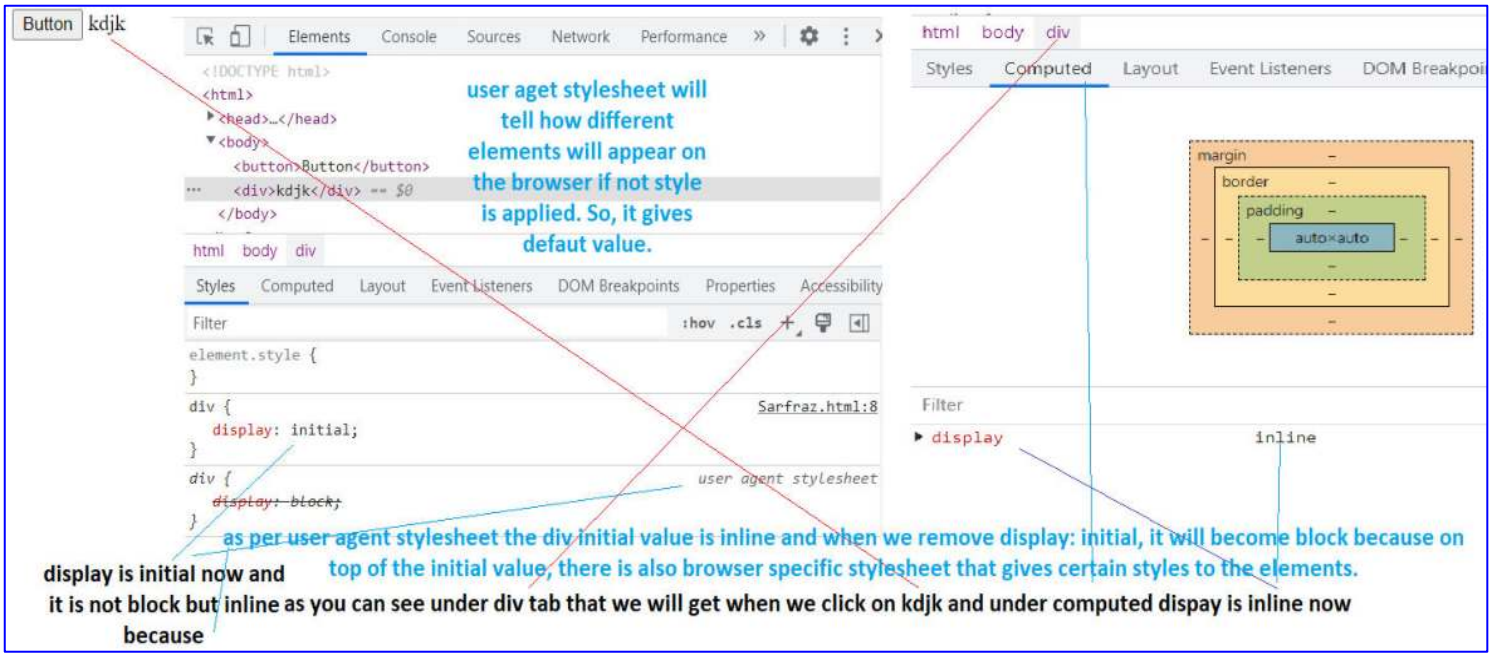




We know that the default value of div is block as you can see below:-


```
<body>
    <div style="display: block;">hello</div>       hello
    <div style="display: block;">hello</div>       hello  // both have the same output as it is default and nothing
changed
</body>
```

Since the default value of div is block by default so if we set display:initial it should be block but it will display inline as shown below:-



Every property has some default value like background has white, color has black etc.

Unset:- Unset will return initial value if nothing has been set in the parent div else inherit.

Agar parent mei kuch set hai to parent ka value yani inherit lega (display mein nahi lega kyonki display inheritance property nahi hai) aur agan parent mein kuchh set nahi hai to initial yani CSS ka default value lega.

Revert- will take the browser's default value; div{display: revert;} will return block(in 2 lines for 2 divs but div{display:initial} will return inline(in one line for 2 divs) because initial takes the CSS default value.

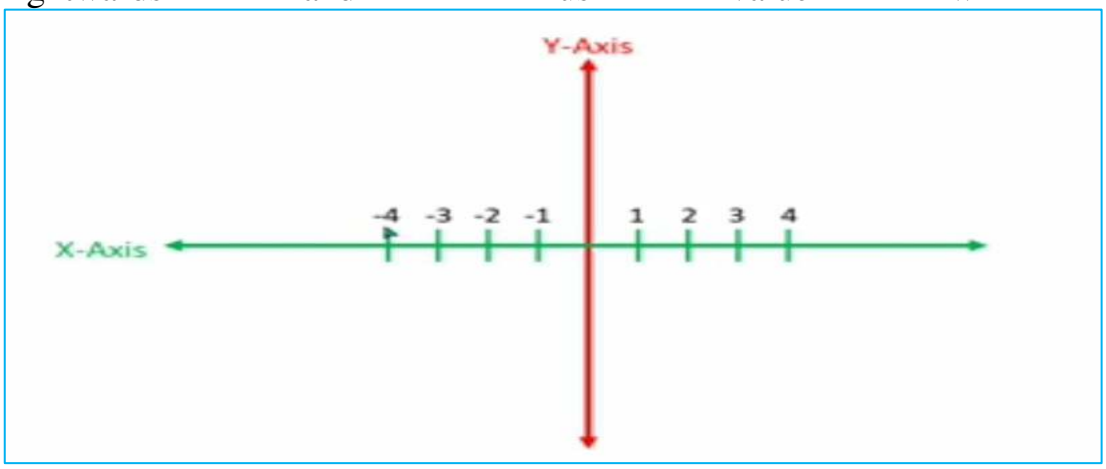
Difference between Unset and Revert:-

Unset will return initial value whereas Revert will return the user aget stylesheet default value(also called browsers style sheet value or value set by different browsers for different properties).

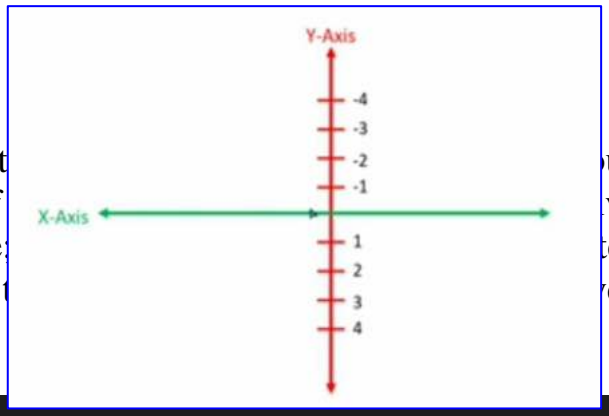
40. Explain CSS position property? Or What are the position states in CSS? Or Differentiate between absolute and relative in CSS.

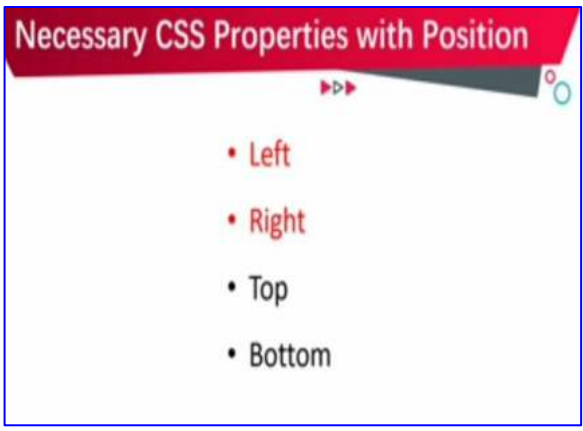
Position is used to position the HTML element. It is a property and it has five values (**Absolute, Relative, Fixed, Sticky, and Static**). The position property needs helper properties like left, right, top and bottom and they are necessary to use with position to make changes.

When we have to move any HTML element horizontally, we us left and right . Plus value will move rightwards and Minus value will move leftwards.



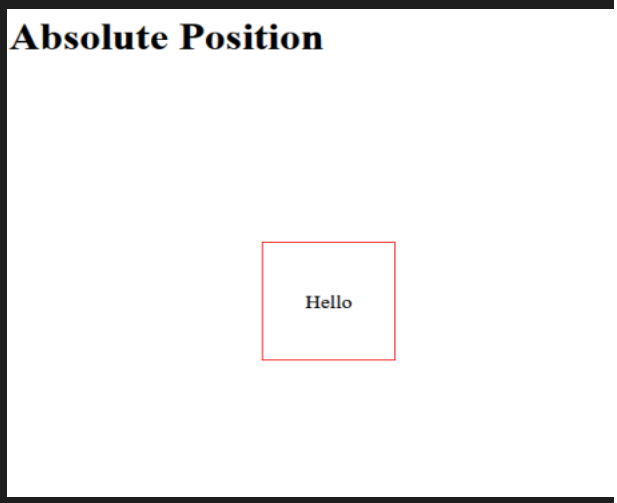
When we have to move any HTML element vertically, we us top and bottom . Plus value will move downwards and Minus value will move upwards as you can see in the image given below.





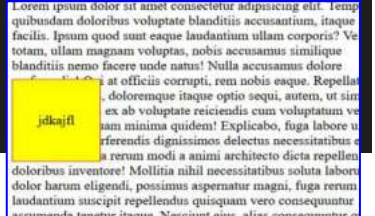
Absolute will move t...u are working on but if ...v and position:absolute; ...te will work according t... ...ve applied.

```
    <style>
        p {
            width: 100px;
            height: 100px;
            border: 1px solid red;
            text-align: center;
            line-height: 100px;
        }
        div{
            top:200px;  /*Will move 200px down against the screen*/
            left:200px;  /*Will move 200px righwards against the screen*/
        }
    </style>
</head>
<body>
    <h1>Absolute Position</h1>
    <div style="position: absolute;">
        <p>Hello</p>
    </div>
</body>
```



Relative will move from its own position.
Fixed will fix the image or anything and if you scroll the content, image will not move but in position: relative/absolute; image will move.

```
<style>
        #sar {
            width: 100px;
            height: 100px;
            border: 1px solid red;
            text-align: center;
```



```
            line-height: 100px;
            background-color:yellow;
            position:fixed;
           /* position:sticky;
            top: 50px; */  It will make the yellow image slide down 500px and then it will
           stop

        }
    </style>
</head>
<body>
    <h1>Absolute fixed</h1>
    <p id="sar">jdkajfl</p>
    <p>Lorem100   </p>

</body>
```

Sticky will also fix the image but after moving the till the value you set. like top:50px then it will move to 50px top and then it will fix.
Static is the default value which does not make any change.The only reason we would ever set an element to position: static is to forcefully remove some positioning that got applied to an element outside of your control.

Note:- If there are two divs and parent div has value position:relative then child div will move according to parent div if position:absolute; applied in child div because absolute of child div will be parent div.

41. How to center align a div inside another div?

Centering with Table:

HTML:

```
<div class=''cn''><div class=''inner''>your content</div></div>
```

CSS:

```
.cn {display: table-cell; width: 500px; height: 500px; vertical-align: middle; text-align: center; }
.inner { display: inline-block; width: 200px; height: 200px; }
```

Centering with Transform

HTML:

```
<div class="cn"><div class="inner">your content</div></div>
```

CSS:

```
.cn {position: relative; width: 500px;  height: 500px;
}
.inner {position: absolute; top: 50%; left: 50%; transform: translate(-50%,-50%);
width: 200px; height: 200px;}
```

Centering with Flexbox

HTML:

```
<div class="cn">
      <div class="inner">your content</div>
</div>
```

CSS:

```
.cn {display: flex; justify-content: center;  align-items: center;}
```

Centering with Grid

HTML:

```
<div class=''wrap_grid''>
       <div id=''container''>vertical aligned text<br />some more text here</div>
</div>
```

CSS:

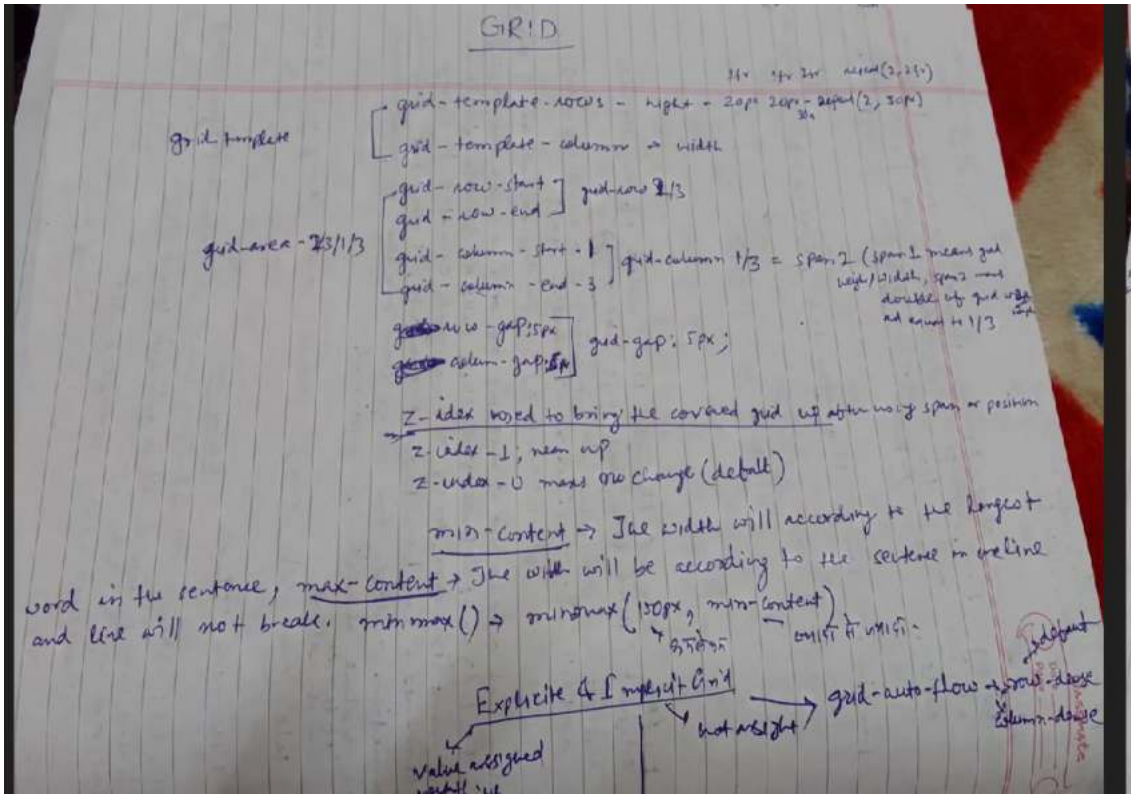
```
.wrap-grid {display: grid;  place-content: center; }
```

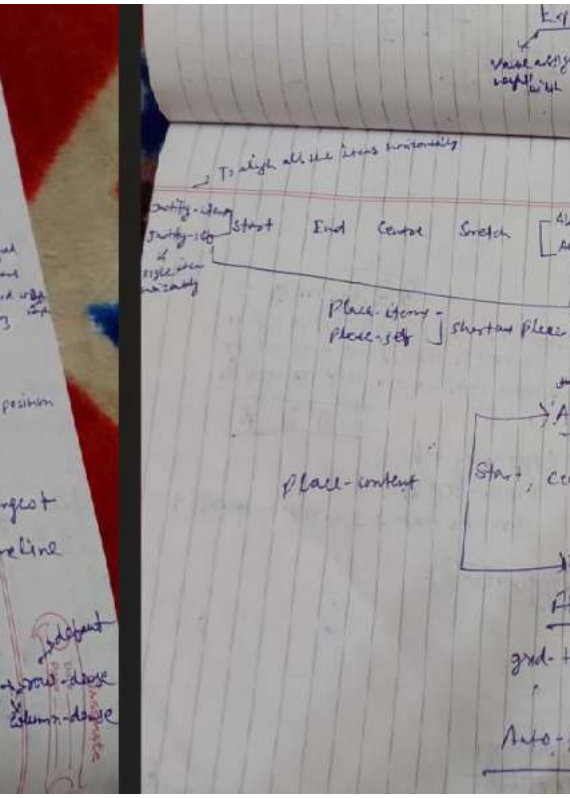
42. What is the grid system?

CSS Grid Layout is the most powerful layout system available in CSS. It is said to be a 2-dimensional system, meaning it can handle both columns and rows, unlike flex-box which is largely a 1-dimensional system.

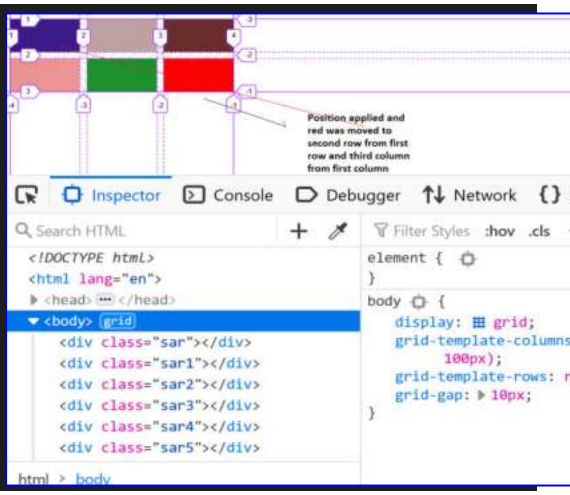
```
    grid-gap: 10px 20px;
    grid-gap: grid-row-gap grid-column-gap; //Shortcut
```

GRID IN SHORT



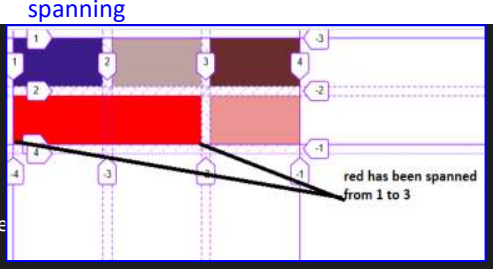



```css
body {
     display: grid;
     grid-template-columns: repeat(3, 100px); 100px  100px  100px
     grid-template-rows: repeat(2, 50px);  // 50px 50px
     grid-gap: 10px; // Shortcut of grid-row-gap grid-column-gap
   }

   .sar {
     background-color: red;
     /* grid-row-start:2;
     grid-row-end:3;
     grid-column-start:3;
     grid-column-end:4; */
     /* grid-row:2/3;
     grid-column:3/4; */
     grid-area: 2/3/3/4;
```



spanning

```css
   }
   .sar {
     background-color: red;
     grid-row-start:2;
     grid-row-end:3;
     grid-column-start:1; //starting from 1 and ending at 3
     grid-column-end:3; or grid-column-end:span2; both r same

}
```



FLEX IN SHORT

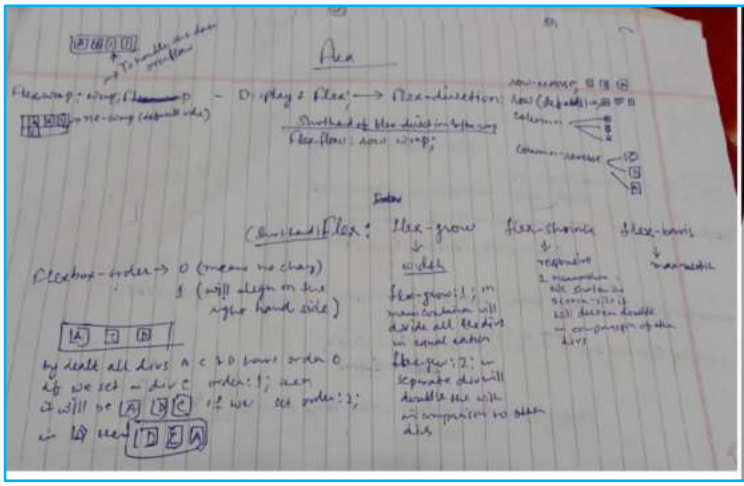
Difference between CSS grid vs flex box?

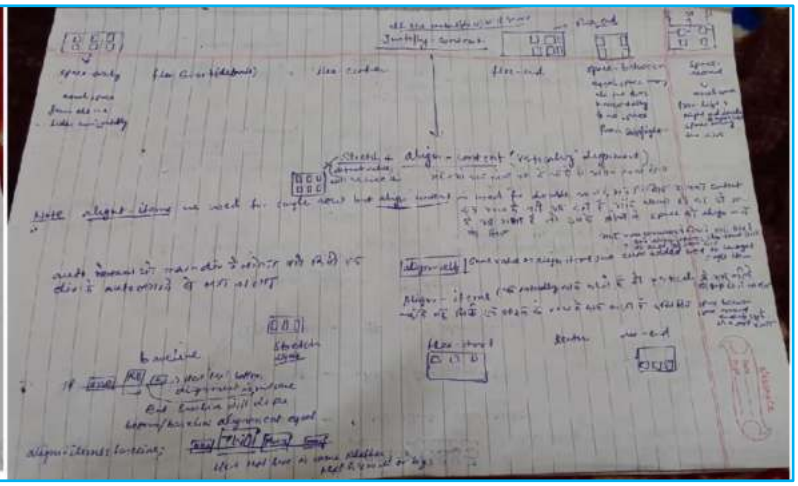
CSS Grid Layout is a two-dimensional system, meaning it can handle both columns and rows. Grid layout is intended for larger-scale layouts which aren't linear in design.

Flex box is largely a one-dimensional system (either in a column or a row). Flex box layout is most appropriate to the components of an application.

What are the properties of flex-box?

The properties of flex box are fl ex-direction, wrap, flow, content, and align-items, and content.

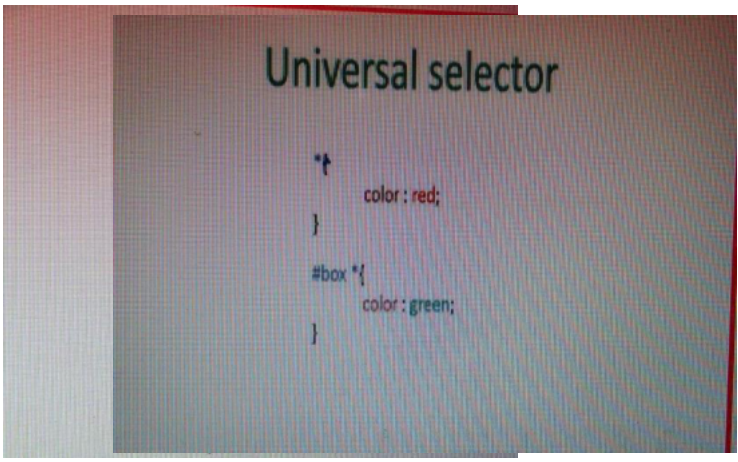


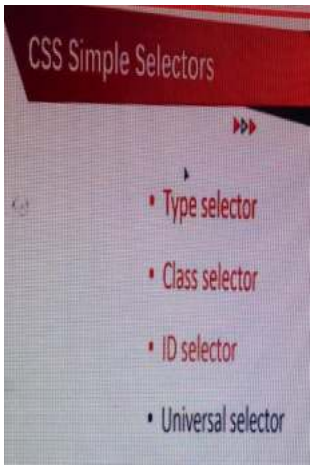


SELECTOR IN SHORT

How is a CSS selector used? What are the different types of Selectors in CSS?

A CSS selector is used to selects the content you want to style. Different types of selectors are listed below.





Universal Selector: The universal selector is used to select all elements on a page.

```
* { color: "green";}
```

Element Type or Tag Selector: This selector matches one or more HTML elements of the same name. In the given example, the provided styles will get applied to all the ul elements on the page.

```
ul { border: solid 1px #ccc;}
```

ID Selector: This selector matches any HTML element that has an ID attribute with the same value as that of the selector.

```
#container { width: 960px;}     <div id="container"></div>
```

Class Selector: The class selector also matches all elements on the page that have their class attribute set to the same value as the class.

```
.box { padding: 10px;  margin: 10px; width: 240px;}          <div class="box"></div>
```

We don't use style pseudo classes, element, id, class in inline style

Selector CSS-3

(Any HTML elements with property and value under curly braces after any tag is called selector)

1. Simple Selectors
   - Type Selector (tag name) – p, div, ul, b { color: red; }
   - Class Selector – .head { color: green; } → class is any name taken in any tag to target, denoted by .(dot); multiple classes can be made in one web pg
   - ID Selector – # box { color: blue; } →
   - Universal Selector – * { padding: 0 }   ul * { color: blue; }   # box * { color: green; }

2. Combinator Selector
   - Descendant Selector: It is used to target the child element

     <div>
      b
     <p> c </p>
     </div>

     To target we will write div span { }
     To target <p> div span p { }

   - Direct Child (some tags are being repeated so the first one is the direct child, denoted by (>))

     <div>   (direct child)
     <div> 6
       (grand child)
     </div>
     </div>

     div > span { It will target just direct child }
     div span { It will target both direct & grand child }

   - Adjacent-sibling, denoted by + sign
   - general-sibling, denoted by tilde (~) sign

      1 → Adjacent-sibling → ul + li { }
      2
      3 → General-sibling → ul ~ li { }

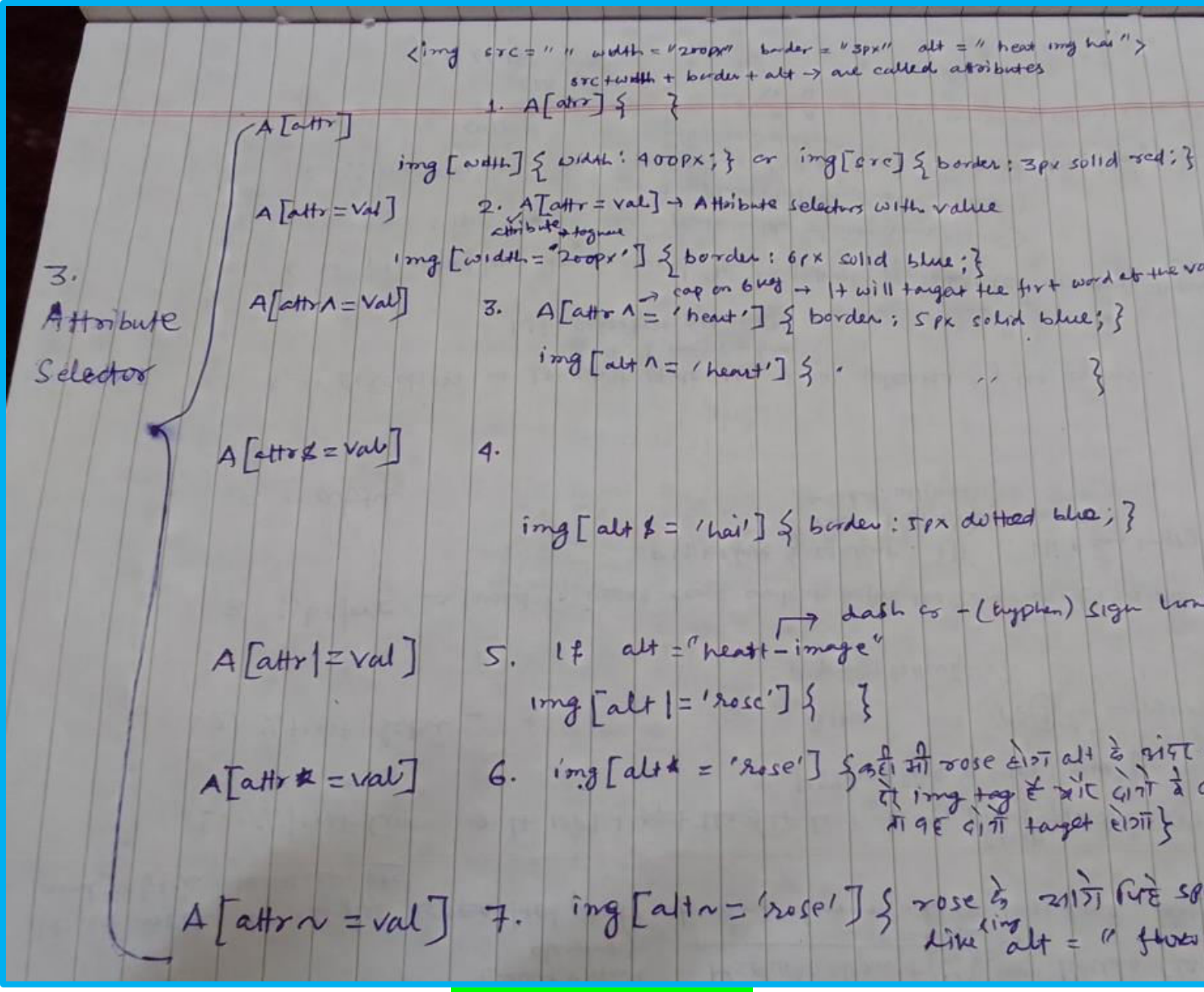
     (~ called tilde, just above the tab key)

<img src = " " width = "200px" border = "3px" alt = " heart img hai " >

src + width + border + alt → are called attributes

3. Attribute Selector
   - A[attr] — 1. A[attr] { }
     img [width] { width: 400px; } or img[src] { border: 3px solid red; }
   - A[attr=val] — 2. A[attr = val] → Attribute selectors with value (attribute + tag name)
     img [width="200px"] { border: 6px solid blue; }
   - A[attr^=val] — 3. A[attr ^= 'heart'] { border: 5px solid blue; } (^ → cap on 6 key → It will target the first word of the value like (heart) img (hai) ✓ $ will target)
     img [alt ^= 'heart'] { }
   - A[attr$=val] — 4.
     img [alt $= 'hai'] { border: 5px dotted blue; }
   - A[attr|=val] — 5. If alt = "heart-image" (dash is – (hyphen) sign, near pas (available before Enter key))
     img [alt |= 'rose'] { }
   - A[attr*=val] — 6. img [alt* = 'rose'] { जहाँ भी rose होगा alt के अंदर तो वह target होगा अगर दो img tag हैं और दोनों के alt = 'rose' rose होगा तो वह दोनों target होगा }
   - A[attr~=val] — 7. img [alt~='rose'] { rose के आगे पीछे space होना चाहिए like <img alt = " flower rose " > }

-



PSEUDO ELEMENT

(w)

Pseudo element (::) → double colon (Select Virtual element) → Faked → Pseudo class (:) → single colon (R)

It is used to style specified parts of an element. like first line, first letter, insert content before and after, ~~to make~~ etc

1. ::first line → It will target the first line of any paragraph (text) like <P>

P::first line { color: 'blue' }

2. ::first letter → " " " " letter

<P> I → will be targeted I am happy </P>

P::first letter { - - }

3. ::before → used to insert some content before the content of an element. <P> I

P::before { content: 'Hi' color: 'blue' } or content: url(s.gif); } Hi I am happy

4 ::after

5. ::selection → It will style the area selected by the user.

h1 { I am Sambav }

h1::selection { color: blue; }

6. ::placeholder - It is used to target the placeholder of any input t

form <input type="text" placeholder="Enter Your Name">

input::placeholder { color: blue; }

7 ::marker → used to target/select the markers (dot, country) of

will be colored blue

 A
 B

- A
- B

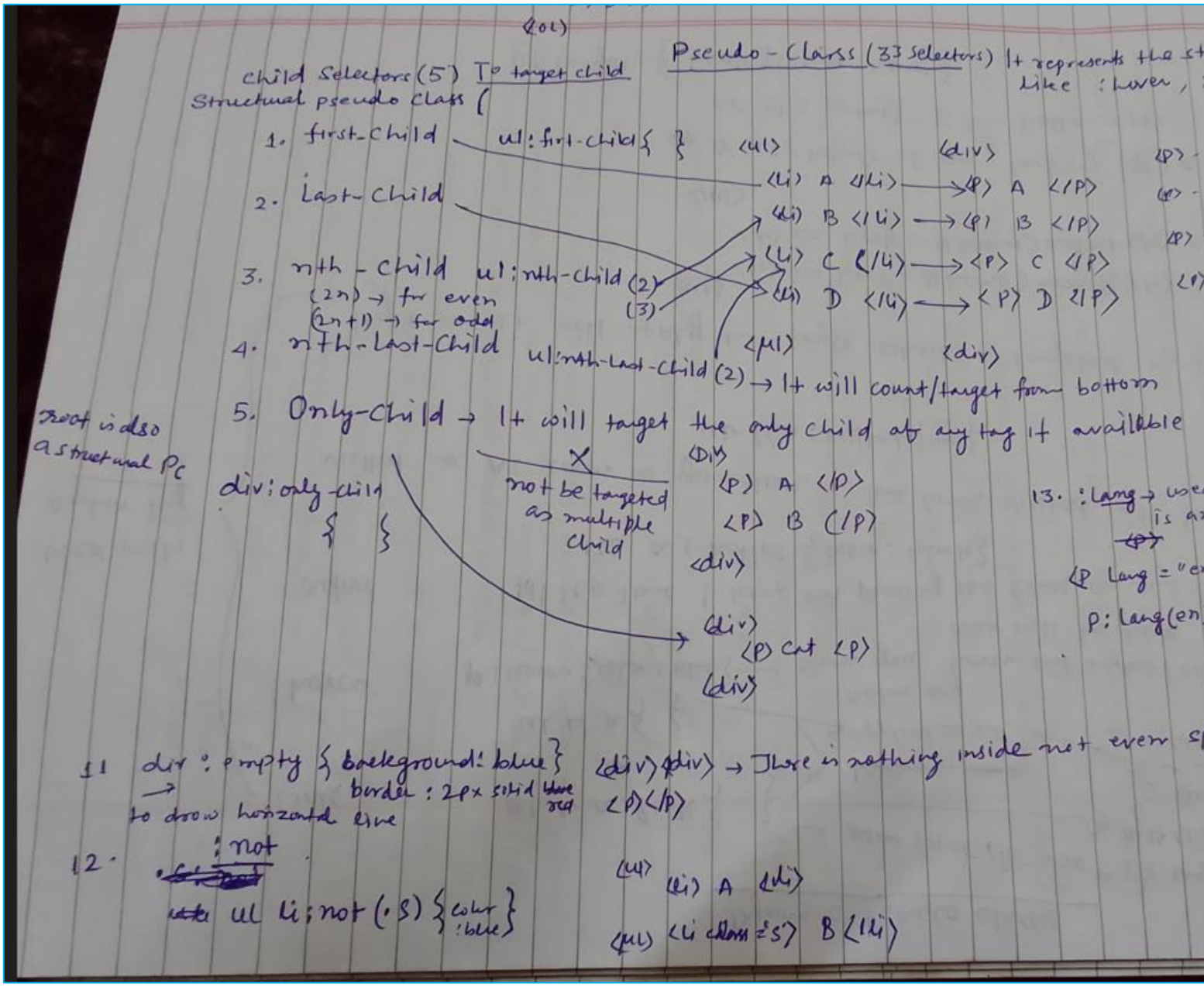
ul::marker { color: blue; }

 A
 B

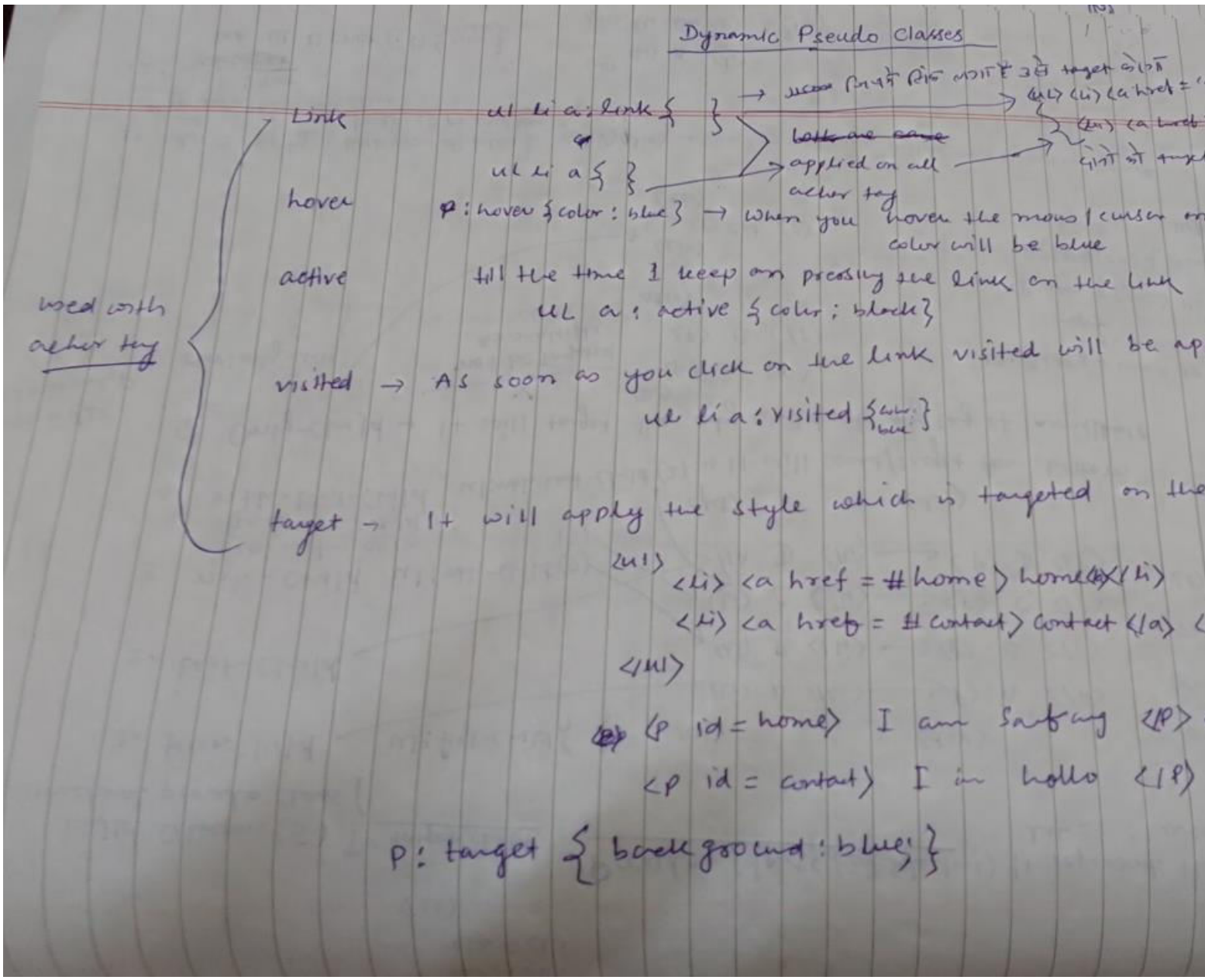
1. A
2. B

ol::marker { - - }

PSEUDO CLASS



DYNAMIC PSEUDO CLASS FOR ANCHOR TAG(LINK, HOVER, ACTIVE, VISITED, TARGET)



FORM SELECTOR

Form

1. focus : to focus the input area <input type = "text" >
input : focus { color : blue } as soon as we will
background : red keep the cursor
color will display

2. accent-color → used to change the default "blue" color
radio button and checkbox.

<label> Are you happy? </label>
<input type = "radio" name = "sanfamej"> <label> Yes </label>
Value = "Yes"

<input type = "checkbox" Value = "Yes" name = "pi"> <label>

Output

Are you happy? ○ Yes □ No

input [name = 'sanfamej'] { accent-color : red
width : 20px ; height : 20px ; }

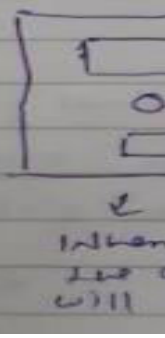
○ → size will be big and
color will be red

input [name = 'pi'] { accent-color : blue }
width : 20px height : 20px

□ will be blue

3. focus-within - used to target the parent element/to
~~focus/checked/radio~~ we focus in input area / textarea
the checkbox or check / select the radio button.

<input type = "text" >
<input type = "radio" >
<input type = "checkbox" >



div : focus-within { background : blue }

2. accent-color → used to change the default blue color of radio button and checkbox.

<label> Are you happy? </label>
<input type = "radio" name = "sanfoundry"> <label> Yes
Value = "Yes"

<input type = "checkbox" Value = "Yes" name = "pi"> <lab

Output

Are you happy? ○ Yes □ No

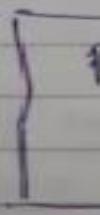
input [name = 'sanfoundry'] { accent-color : red
width : 20px; height : 20px }

○ → size will be big and color will be red

input [name = 'pi'] { accent-color : blue }
width : 20px height : 20px

□ will be blue

focus-within - used to target the parent element ~~focuse/checked/radio~~ we focus in input area / textar the checkbox or check / select the radio button.



```
<div>
    <input type = "text">
    <input type = "radio">
    <input type = "checkbox">
</div>
```

```
div : focus-within { background : blue }
```

4- Valid & Invalid - will target if we ente
like in email section or input area if we
formate of email ID then invalid will
valid will be triggered.

<input type = "email" placeholder = " En
<input type = " submit " >

[Enter Email ID] [submit]

input : invalid { color : red ; }

As soon as we enter wrong em
like surb or surb@ it will be

input : valid { color ; green ; }

surb@gmail.com → will be green
it will be green because red will highlig
you start writing and that is wrong.

5. required & optional - used when you
must enter the particular thing in the inpu
If you don't want it necessary, you can
If user dosent enter, it will pop up to ente
submit.

<input type= "text" required > <inp

[ ] [Submit]

input : required { color : blue; } ↓ necessary to fill then up else ad

<input type = " text " optional > < input

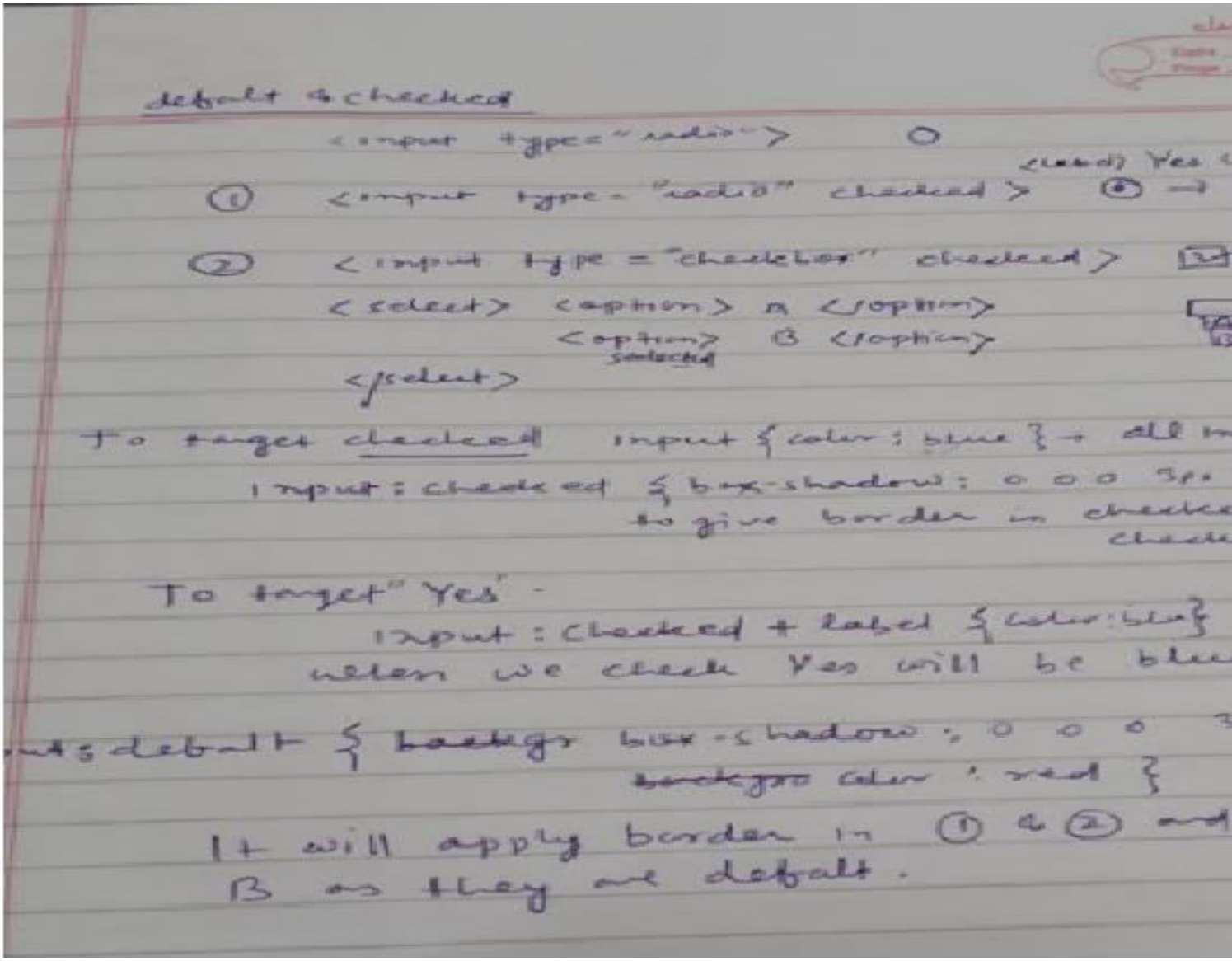
input : optional { color : gree; }

6. enabled and disabled : enabled will
disabled will disable the input area. So nothing

<input typ = "text" disable read-only>

↙ will disable ↓ can't write

input : disabled { }
enabled
read-only



42. What are the different ways to hide the element using CSS?

Using display property(display: none). It's not available for screen readers. The element will not exist in the DOM if display: none is used.

Using visibility property(visibility: hidden), will take up the space of the element. It will be available to screen reader users. The element will actually be present in the DOM, but not shown on the screen.

Using position property (position: absolute). Make it available outside the screen.

```
<style>
        p {display: none;}

        div {visibility: hidden;}
```

```
        main {position: absolute;  top: -100px;}
        /* Yes screen or viewport se upar chala gya isliye chhup gya */
</style>
</head>
<body>
```

```
    <p>I am display none</p>
    <div>I am visibility hidden</div>
    <main>I am position absolute </main>
</body>
</html>
```

43. What does the :root pseudo-class refer to? Or Define CSS Variable.

The :root selector allows you to target the highest-level “parent” element in the DOM, or document tree. It is defined in the CSS Selectors Level 3 specification.

It is used to stop repetition

For example, if a web-page has footer and header with the same color like red and want to change it to black then we can simply change the color one by one or We can use variable to change both the footer and header color with single command called CSS variable. It will save time. We need to use pseudo class root to make variables like :root{ variableName:red;} or :root{--main-color:red;} you can take any variableName instaed of --main-color. Here red means wherever red it is there in the web-page will change if we change it to :root{--main-color:blue;} Now, if we want to change the color of class A then we will write as .classA{ color:var(--main-color)};

We will make variable on the top under style.

```
 <style>
        /*
    :root{--main-color:yellow;  --head-font:verdana;} */
    /* We can write this as well and :root{} needs to take on the top under style */

        :root {
            --Sar: red;
            --Pam: 32px;
        }
        p,div,.a {
            color: var(--Sar);
            font-size: var(--Pam);
        }
 </style>
</head>
```

```
<body>
    <p>I am red</p>
    <div>I am div</div>
    <div class="a">I am "a"</p>



 </body>
```

```
</html>
```

44. What does Accessibility (a11y) mean? Or Semantic Element

Accessibility is used for the people have less vision or can't see or having hearing loss or some other disabilities.

We have modern browsers that provide screen reader for them so that when they move the cursor on the screen, the reader can read the content or element name for him. This is only possible when we use Semantic elements in our web-page or websites properly.

We also need to install screen reader extension in chrome browser.

Semantic Elements are those elements that clearly describe its meaning to both the browser and the developer.

Div and span are non-semantic elements because they are wrappers and don't tell whom wrappers they are but button, table, form are Semantic element as when you click on button, the screen reader will read or say that it is a button or in image alt="" , alt is semantic element as whatever you write under double quotation will be read by the screen reader and when we take the cursor on the form, article, aside, details, footer , header, sidebar or table etc. screen reader will read out that they are table and form but in the case of div and span it is will read the inside content like in the case of I am Sarfraz and Hello , the screen reader will say I am Sarfraz and Hello, so we will not understand that div and span are there. So, they are non-semantic elements.

45. How does Calc work?

The CSS3 calc() function allows us to perform mathematical operations on property values. Instead of declaring, for example, static pixel values for an element's width, we can use calc() to specify that the width is the result of the addition of two or more numeric values.

```
.foo {width: calc(100px + 50px)}
```

46. What do CSS Custom properties variables mean?

Custom properties (sometimes referred to as CSS variables or cascading variables) are defined by users that contain specific values to be reused throughout a document. The value is set using -- notion. And the values are accessed using the var() function.

```
 <style>
        :root{
            --sar:red;
        }
    </style>
</head>
<body>
    <p style="background-color:var(--sar);">jdkajfl</p>
    <h1 style="color:var(--sar);">Absolute fixed</h1>
</body>
```

47. What does * { box-sizing: border-box; } do? What are its advantages?

- ❖ It makes every element in the document include the padding and border in the element's inner dimension for the height and width computation.
- ❖ In box-sizing: border-box, The height of an element is now calculated by the content's height + vertical padding + vertical border width.

- ❖ The width of an element is now calculated by the content's width + horizontal padding + horizontal border width.

48. What does !important mean in CSS?

The style is having the important will have the highest precedence and it overrides the cascaded property.

```
p {
        color: red !important
t;
}
#thing {
        color: green;
}
<p id="thing">Will be RED.</p>
```

49. What is progressive rendering? How do you implement progressive rendering in the website?. What are the advantages of it?

Progressive Rendering Explanation
https://www.youtube.com/watch?v=AUDjRpk2OWY

Progressive Wep App(PWA)
https://www.youtube.com/watch?v=ZGvq3_e4awY

Progressive rendering is a techniques used to improve the performance of a web-page so that web page can display as quickly as possible.

Progressive rendering is a concept that every web developer should know because it helps a lot in the fast loading of a website. Most of the developers are using progressive rendering but they are unaware that the thing that they are doing to load the website quickly is actually progressive rendering.

What is Progressive Rendering?

Before Understanding Progressive rendering, We should understand what is rendering of the web page and what progressive is.

Rendering means converting code into an interactive web page that our users can see and utilize its functionality. Now, being Progressive means doing things such that the highest

priority thing will be done first, after that less priority thing will be done and in this manner, work will be completed.

So, Progressive Rendering means rendering the web page in such a manner that high priority component will be rendered first and then low priority component will be rendered. Before going deep into it, we should understand how we can give priority to different components of our web page.

So the priory of our component should depend upon the viewport. This means those components that come in the viewport after loading the website for the first time should get high priority and those components that are below that viewport should get low priority.

How Progressive Rendering is different from Server Side Rendering(SSR) and Client Side Rendering(CSR)?

So before knowing that difference, one should first know what client-side rendering and server-side rendering are:

Client-Side Rendering: It means rendering pages directly in the browser using JavaScript.

All logic, data fetching, templating and routing are handled on the client or browser on which you are working on rather than the server.

How it works behind the scenes.

Suppose you want to visit to facebook.com, when you visit facebook.com, you write facebook.com on the address bar and then browser like chorme, opera on which you are trying to open the facebook

1. Client/ browser will send a request to the facebook server to open the facebook.com then facebook server gives response to the client means browser(opera) in html and css, these html and css will be almost empty or there will be no content because all content is rendered by javascript.

3. Then clients will start parsing html and css but there will be no content because html has no content so we will see a blank page.

4. After parsing the parser will come across the JavaScript tag at the bottom of the html before closing tag of body as we put the script tag before closing tag of body. Now, parser will start downloading the JavaScript where our content available. Note parsing html and css is fast but not JavaScript and this is why a blank page will be rendered to the user until the JavaScript is executed because the HTML has no content so the user will have to wait JavaScript to be loaded and executed.

5. Once the JavaScript is executed, all necessary content is rendered and now application become viewable and interactive.

In Client-Side Rendering the content is rendered on the client side. Here when a user visits any web page then request to get an HTML and CSS file is sent to the server and then the server sends an HTML and CSS file to the client side. After processing that code, a request

to get the JS file is made and till the JS file is not loaded, HTML and CSS will not be shown on the browser. So the user will see a blank page until the JS file will not be loaded.

Server-Side Rendering: In Server-Side Rendering, the content is first rendered on the server side and then send to the client side. Here when a user visits and request to get a webpage is made to the server then the server first loads the HTML and CSS file on itself and then processes it and then sends the processed web page to the client side. After getting that webpage, the User can see the webpage and the browser will be waiting for the JS file to be loaded. Till the JS file is not loaded, the User can only see the webpage but can't interact with it. Interaction with the web page is only possible with the help of JS(Javascript).

Now here the user has to wait till the Webpage containing HTML and CSS is not shown on the screen. But, It is faster than Client-Side Rendering because in Client-Side Rendering, a web page is only shown when javascript will be loaded but this is not the case in Server-Side Rendering.

Now it's time to see how Progressive rendering is different from both of these.

Progressive Rendering: In progressive Rendering, the web page is divided into different parts on the basis of priority. So different parts will be rendered sequentially according to Server-Side Rendering and sent to the client side. This means that here first whole page will not be rendered on the server instead, high priority part of the page is rendered and sent to the client side, then low priority part of the page is rendered and sent to the client side.

Benefits of Progressive Rendering: These are some of the major benefits of Progressive Rendering-

It improves the load time of the website and that eventually leads to a better User Experience.

It optimizes the critical rendering path. Critical rendering path is referred to those steps that are done between receiving HTML, CSS, and JavaScript code and converting them into a visual web page.

It helps us to overcome the drawback of server-side rendering and client-side rendering.

Different ways to Achieve Progressive Rendering-

1. Lazy Loading of Image: Lazy loading of an image means loading an image when it is about to come in the viewport instead of loading all the images initially at once. This will lead to the progressive rendering of the web page because here the high-priority image is the image that is initially in the viewport and is loaded first and the low-priority image is that which is about to come in the viewport and will be loaded.

2. Not processing all the CSS initially: Most of the developers have a habit of adding all CSS in the head section and because of that whole CSS will be processed first before processing of HTML. But to achieve progressive rendering we should only add that much CSS in the head section that is necessary for those HTML that will be in the viewport at the

start. After that, Add the remaining CSS in the body before the HTML tags to which it will be applied.

50. Does style1.css have to be downloaded and parsed before style2.css can be fetched?

```
<head>
        <link h ref=" stylel. css" rel=" stylesheet">
        <link href="style2.CSS" rel="stylesheet">
</head>
```

No, the browsers will download the CSS in the order of its appearance on the HTML page.

51. How to determine if the browser supports a certain feature? Or @support

We can also visit www.caniuse.com to check if particular property is supported by this particular browser.

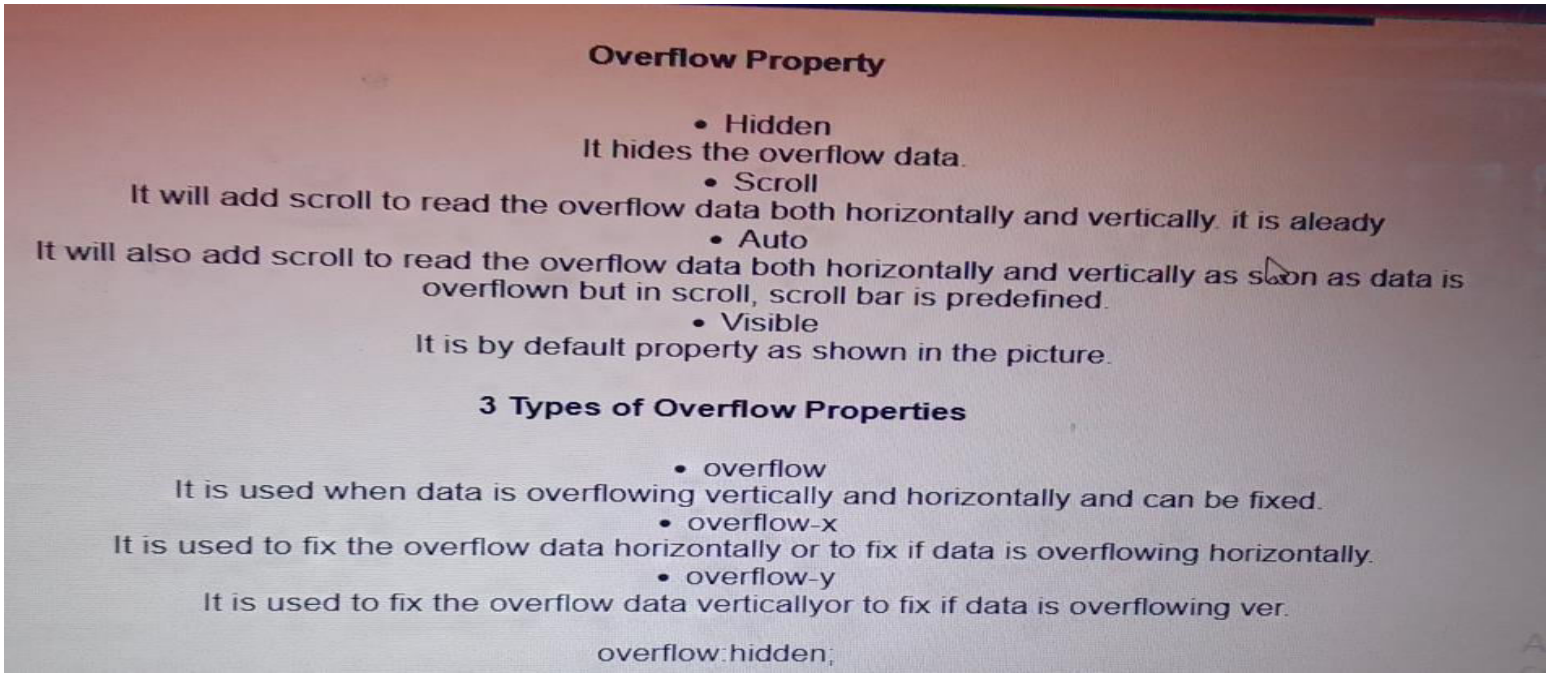
The @support in CSS can be very useful to scan if the current browser has support for a certain feature/property

```
        <style>

         background will be blue if display:flex is supported by the firefox. Since it is supported by
chrome
          and firefox, it wil work

         @supports(display:flex){
               body{background: blue;}
                              }

         background will not be blue if display:flex is supported by the firefox
         @supports not(display:flex){
               body{background: blue;}
                              }
          @supports (text-underline-offset:10px){
               body{background: #1022;}       }

        </style>
</head>
<body>
   <h2>@Support Selector</h2>
   It is new selector used to check CSS feature means CSS properties and its values
    supported by a particular browsers or not.

    <h3>@support can be used for mulitple conditions as well.</h3>

     @support(display:flex) or (display:-webkit-flex){   #main{display:flex;}

     /** Here 'or' operator means css for #main{display:flex;}  will work if anyone of these two
conditions are
     supported by the user's browser or both of them.  Here, @support(display:flex) means user's
browser is
     compatible for the display:flex and (display:-webkit-flex) means not compatible as using old
version of
```

```
     Chrome and firefox as used -webkit. Similarly, we can use 'and' operator. In 'and' operator both
the condition
     should run in the browser you are using then only the command will work.

     @support(display:grid) and (display:-webkit-grid){   #main{display:grid;}    }
     /**It will work if user's browser support both the conditions.
<
```

52. How does this property work overflow: hidden?

```
We use overflow property to fix the overflow issues when more content added to a fixed height element.
```



53. What do you have to do to automatically number the heading values of sections and categories?

We now that when we take any ordered list with 5 lists as children, all the children will be aligned in 5 different lines with 1 to 5 order list. Earlier, it was not possible to change the order or set something else in the place of order list 1 to 5 but now it is possible to set your own order list and string you want with the help of **counter()** function and its properties **counter reset and counter increment.**

There are three properties of counter and they are actually three phases to run the counter.

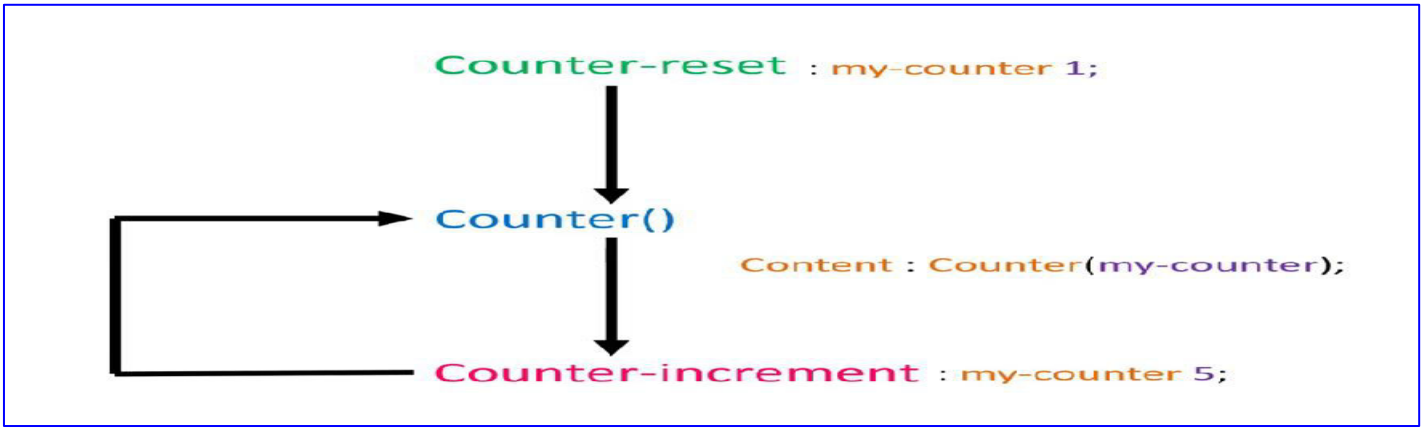
First one is, Counter reset: my-counter 1; here we give one name and also define the initial or starting point from where we have to start the counter.

Second is Counter() function that will run the counter (my-counter from 1) but for this we will use

Content: counter(my-counter); // It will print the counter and to increment from 1 to 2 we will use third property

Counter-increment: my-counter 5; // Here we define that my-counter needs to increment by 5, you can given 2,3 etc.

So, it will print 1, 6 , 11 etc.




```
<style>
        body{
            counter-reset:my-counter 0;
        }
        h1::before{
            /* content: counter(my-counter);         It will display 0 Heart */
            /* content: counter(my-counter) " : ";   It will display 0 : Heart */
            content:"Sarfraz " counter(my-counter) " : ";   /* It will display Sarfraz 1 : Heart */
            counter-increment:my-counter 1;          /* It will increase the counter 1, 2 etc
        }
    </style>
</head>q
<body>
        <h1>Heart</h1>
        <p>Heart is pure</p>
        <h1>Heart</h1>
        <p>Heart is pure</p>
</body>
```

Sarfraz 1 : Heart

Heart is pure

Sarfraz 2 : Heart

Heart is pure

55. How is the nth-child() different from nth of type selectors?

Both are pseudo-classes (Pseudo-classes are those keywords that specifies the special state of the selected element). The *nth-child()* pseudo-class is used for matching elements based on the number that represents the position of an element based on the siblings. The number is used to match an element on the basis of the element's position amongst its siblings.

For example, in the below piece of code, if we give nth-child(4) for the example class, then the 4th child of the example class is selected irrespective of the element type. Here, the fourth child of the example class is the div element. The element is selected and a background of black is added to it.

```
.example:nth-child(4) {
    background: black;
}
<div class="example">
        <p>This is a paragraph.</p>
        <p>This is a paragraph.</p>
```

```
        <p>This is a paragraph.</p>
        <div>This is a div.</div> <!-- 4th Element to select and apply style-->
        <div>This is a div.</div>
        <p>This is a paragraph.</p>
        <p>This is a paragraph.</p>
        <div>This is a div.</div>
</div>
```

The nth-of-type() pseudo-class is similar to the nth-child but it helps in matching the selector based on a number that represents the position of the element within the elements that are the siblings of its same type. The number can also be given as a function or give keywords like odd or even.

For example, in the below piece of code, if we give p:nth-of-type(even) for the example class, then all the even paragraph tags are selected within the example class and the style of background black is applied to them. The selected elements are marked in comments in the below code:

```
.example p:nth-of-type(even) {
    background: black;
}
<div class="example">
        <p>This is a paragraph.</p>
        <p>This is a paragraph.</p> <!-- Select this and apply style-->
        <p>This is a paragraph.</p>
        <div>This is a div.</div>
        <div>This is a div.</div>
        <p>This is a paragraph.</p> <!-- Select this and apply style-->
        <p>This is a paragraph.</p>
        <div>This is a div.</div>
        <p>This is a paragraph.</p> <!-- Select this and apply style-->
        <div>This is a div.</div>
</div>
```

56. What is the importance of CSS Sprites?

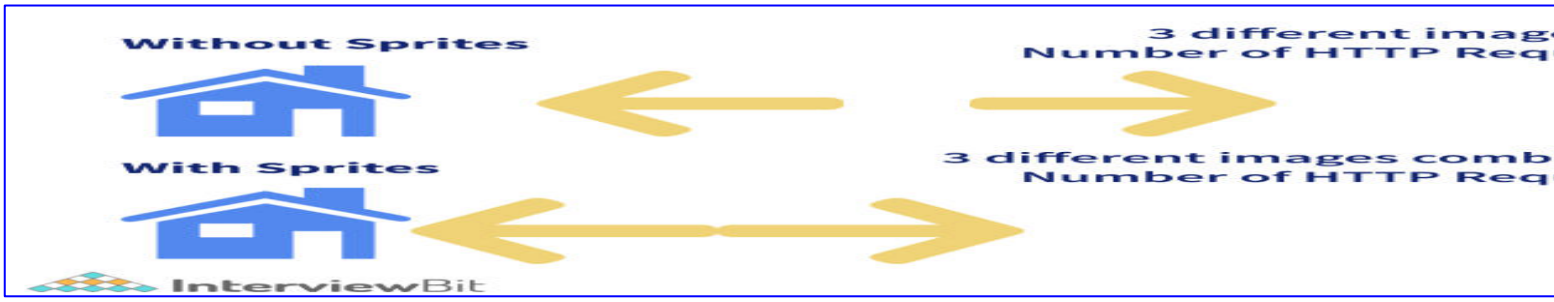
CSS sprites are used for combining multiple small images in a single larger image and then use that single larger image into multiple small images in CSS.

For example, if I have three different flags for different country. I will combine them and make a larger image and at the time of using that larger image in CSS we will use the three flags separately whenever I need whichever country flag.

In the development of the project, we may need several small icons, social media icons or small images and the more images we use will go to the server more times. So, we will combine the all small images in one so that it reduces the number of HTTP requests to get data of multiple images. Now, we will send only one single request.

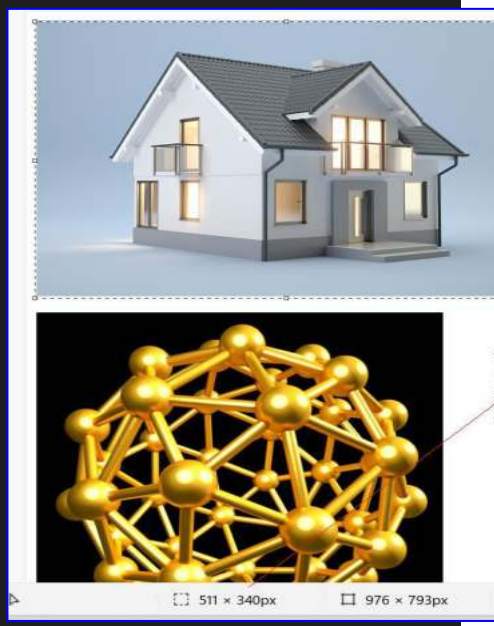
- The limitation of image sprites is that it can be used as a background image only not work in direct image.
- It reduces the number of HTTP requests to get data of multiple images as they are acquired only by sending a single request.
- It helps in downloading assets in advance that help display icons or images upon hover or other pseudo-states.
- When there are multiple images, the browser makes separate calls to get the image for each of them. Using sprites, the images are combined in one and we can just call for that image using one call.

Consider an example where our application requires 3 images as shown below (Without Sprites Section). If we are trying to load the images independently, we require 3 different HTTP Requests to get the data. But if we have CSS Sprites where all 3 images are combines into 1 separated by some spaces, then we require only 1 HTTP Request.

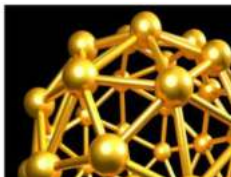


Limitation- We can use it with background-image only not direct image.

```
  <style>

 #sprite{
          background-image:url("css_sprites.png");
         background-repeat:no-repeat;
         height: 340px;
         width:511px;  // Only home will be displayed by using
                       // height:340p
x & width: 511px
     }
  </style>
</head>
<body><div id="sprite"></div></body>
```




```
                                                    <style>
   #sprite{
```



```
            background-image:url("css_sprites.png");
            background-repeat:no-repeat;
            height:400px;
            background-position:0px -375px;
    }
  </style>
</head>
<body><div id="sprite"></div></body>
```

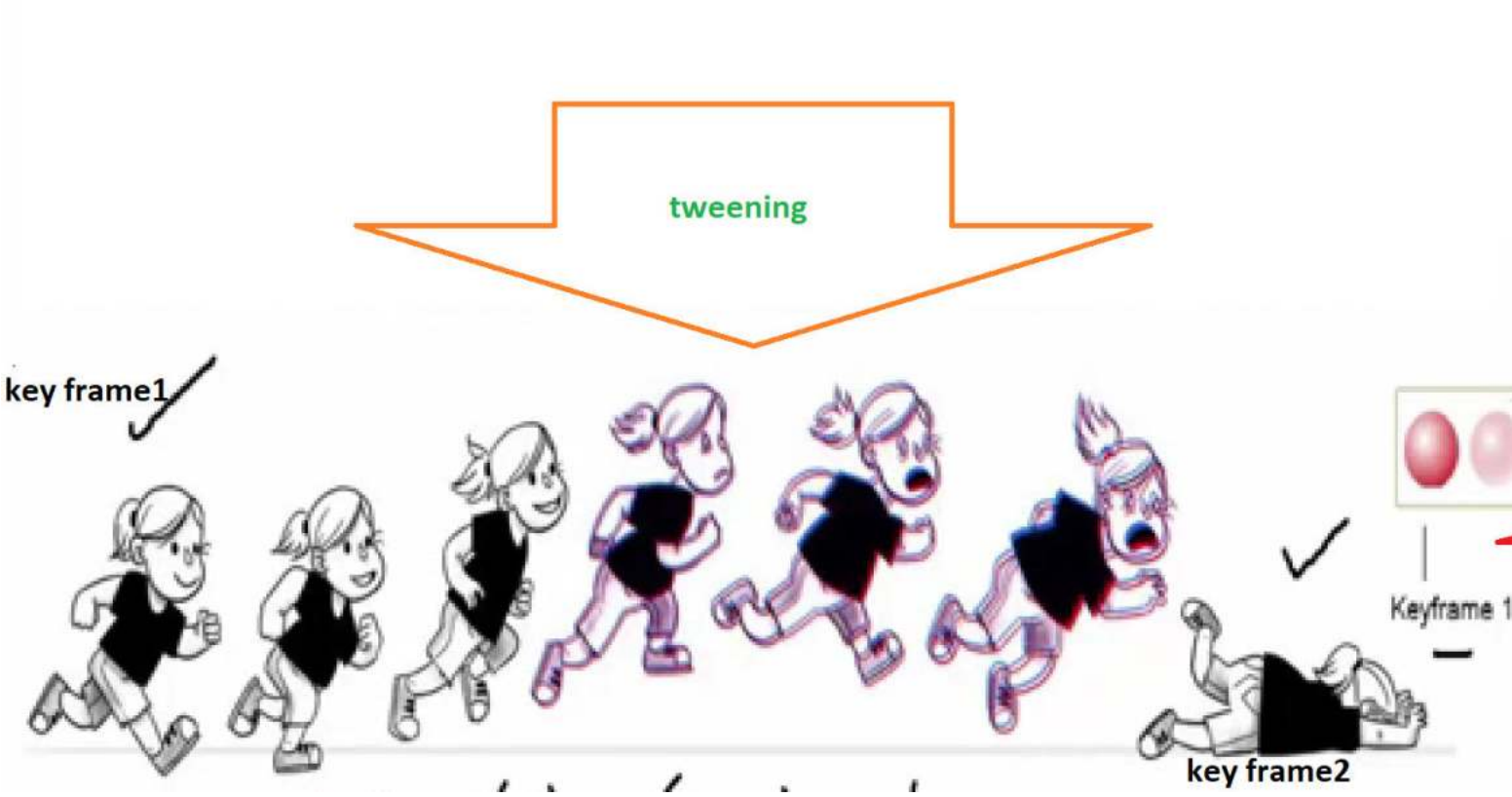
57. What do you understand by Key Frame, Tweening or In-Between in CSS? animation : name duration tyming-fuction delay count direction (ND/TD/IC,DI)

Whenever we use any animation we have a starting point from where we have to start the animation and ending point where we have to end the animation.

The starting and ending point of the animation is defined by key frame or called key frame. The starting point is called key frame1 and ending point is called key frame2.

The intermediate key frames between the key frame 1 and key frame 2 are called tweening or in-between in CSS.

In another word, Tweening is the process of filling the gaps between the key frames that are already created or the intermediate(in-between) keyframes are called tweening.



For this purpose, we use properties like transforms - matrix, translate, scale, rotate etc.

In the below example, we are generating intermediate frames of paragraph elements to slide through from the start to the right edge of the browser.

```
p {
  animation-duration: 2s;
  animation-name: sarfraz;
}

@keyframes sarfraz {
  from {
    margin-left: 100%;
    width: 300%;
  }

  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

Here, the paragraph element specifies that the animation process should take 2 seconds for execution from start to the finish. This is done by using the `animation-duration` property. The animation-name of the `@keyframes` is defined by using the property `animation-name`. The intermediate keyframes are defined by using `@keyframes` rule. In the example, we have just 2 keyframes(frame 1 and frame2 used). We can define tweening or intermediate keyframes as well to do different actions at different points. The first keyframe starts at 0% and runs till the left margin of 100% which is the rightmost edge of the containing element. The second keyframe starts at 100% where the left margin is set as 0% and the width to be set as 100% which results in finishing the animation flush against the left edge of the container area.

58. How will you fix browser-specific styling issues?

Whenever we make any website, we need to check whether that website is working fine or looking fine in different screen sizes and different browsers or not and to check this, we need to do some test and that test is called cross browsers compatibility testing or browser-specific styling testing to fix the issues.

We also call it CBT(Cross Browser Testing)

Web Testing Coverage(The data of all the browsers, devices, Screen-Sizes and Operating Systems on which website will be tested)

We create three kinds of applications.

1. Web Application:- We will focus on Web Application and we will test it in different browsers like C hrome, Firefox, IE etc. and laptops of different companies of different screen-sizes. So, we need to check/test our web in these different platforms to ensure my web will work and look fine.
2. IOS Application
3. Android Application

We can **write browser-specific styles separately** in different sheets and load that only when the specific browser is used. This makes use of the server-side rendering technique. In another word, write CSS for different browsers like Chrome, IE and load IE style when user is viewing the website on IE or Chrome style when viewing in Chrome and so on.

We can use **auto-prefix** for automatically adding vendor prefixes in the code.

We can also use **normalize.css or reset CSS** techniques.

There are some ways for avoiding browser compatibility issues too. They are as follows:

1. **DOCTYPE ERROR**- It is better to use DOCTYPE because older browser versions check for DOCTYPE tag at the beginning and if not found, the application rendering won't be proper.We know that !doctype(document type) is used to tell the version of document of HTML to the browser. If we write <!DOCTYPE html> it means that we have used html5 and the other developer will understand that in this document html5 has been used and edit accordingly. It is not case sensitive. So, can be written in small.
2. **Browser Detection**:- Browser Detection is used to get user's data like which browser the user is using, what the version and agent of the browser are so that we do the coding accordingly.

In javaScript we use code to detect the AppName(Browser's name), AppVersion(Browser's version) and agent using

navigator.appName(to get app) navigator.aapVersion(for version) and navigator.userAgent.

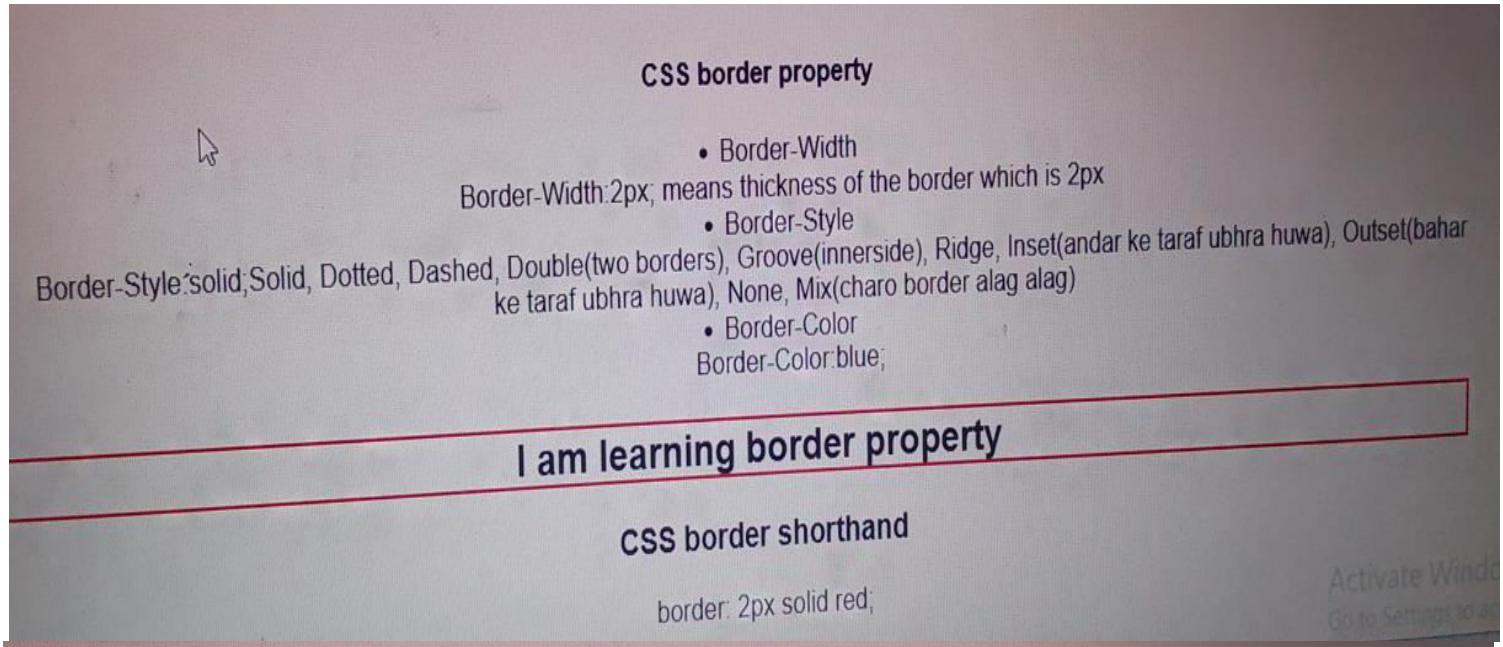
```
<h3>Browser Detection</h3>
    <script>
        "use strict"
        let a = navigator.appName;
        let b = navigator.userAgent;
        let c = navigator.appVersion;

   console.log(a);  // Netscape
   console.log(b);  // The default browser will show even if you open in chrome, //Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:107.0) Gecko/20100101 Firefox/107.0
   console.log(c);  // 5.0 (Windows)

    </script>
```

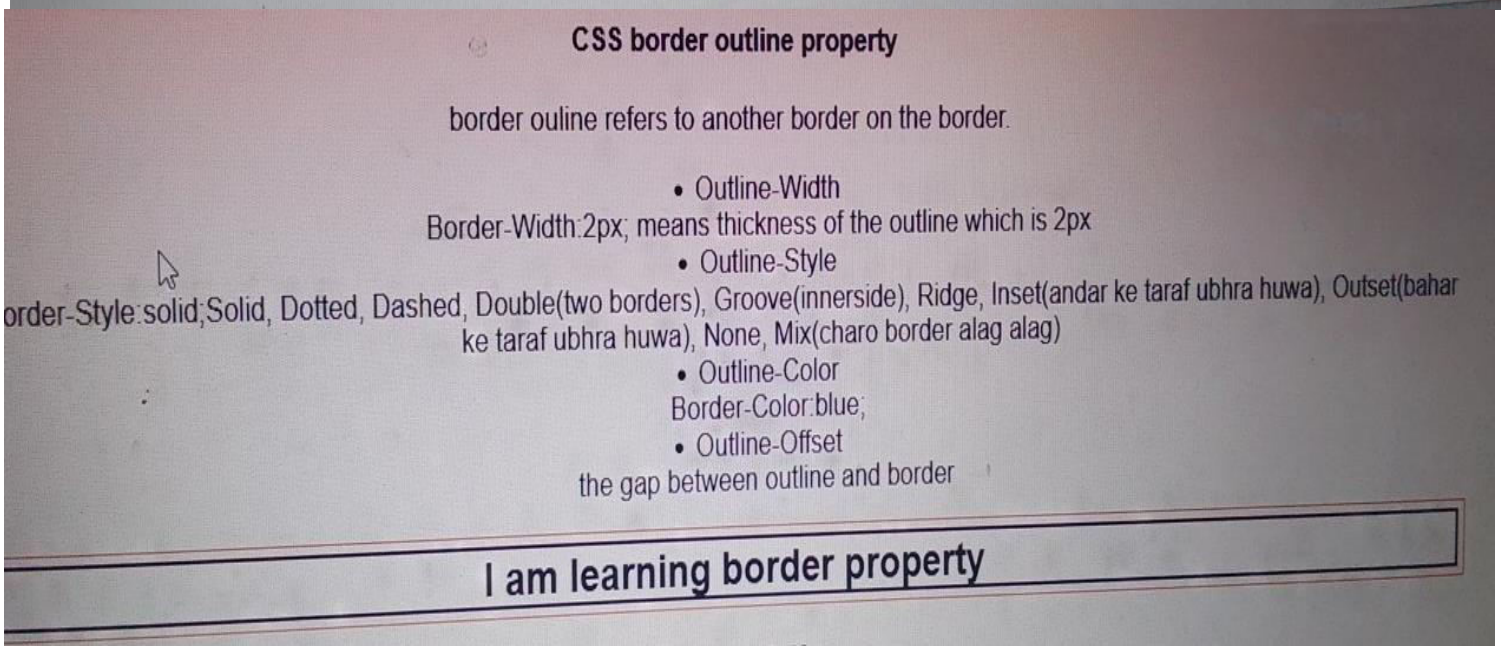
3. **Validation of HTML and CSS code**: We know the codes we have written in the HTML and CSS files will be used by different browsers. So, we need to validate or check these codes like if closing tags, semicolons are missing or syntax is wrong which can create problems while rendering the code. We can use tools for validate such as W3C, HTML and CSS validator or Jigsaw.
4. **CSS reset**:- To ensure our website look alike in all the browsers used by the users. If we don't do so, browser's default value such as line-height, color, font-size, and margins etc. will be applied and it can be different for different browsers.
5. **Layout compatibility**:- We need to check this as well and better to use flex and grid for making layout to avoid layout issues.
6. Use **friendly libraries and frameworks** like bootstraps
7. **Separate Style Sheet for different browsers**.
8. **Test on real devices:** It is good to use tools like Test sigma for this purpose that enables us to test in real devices.

In short, we can use BrowserStack for Cross Browser Testing using Selenium.

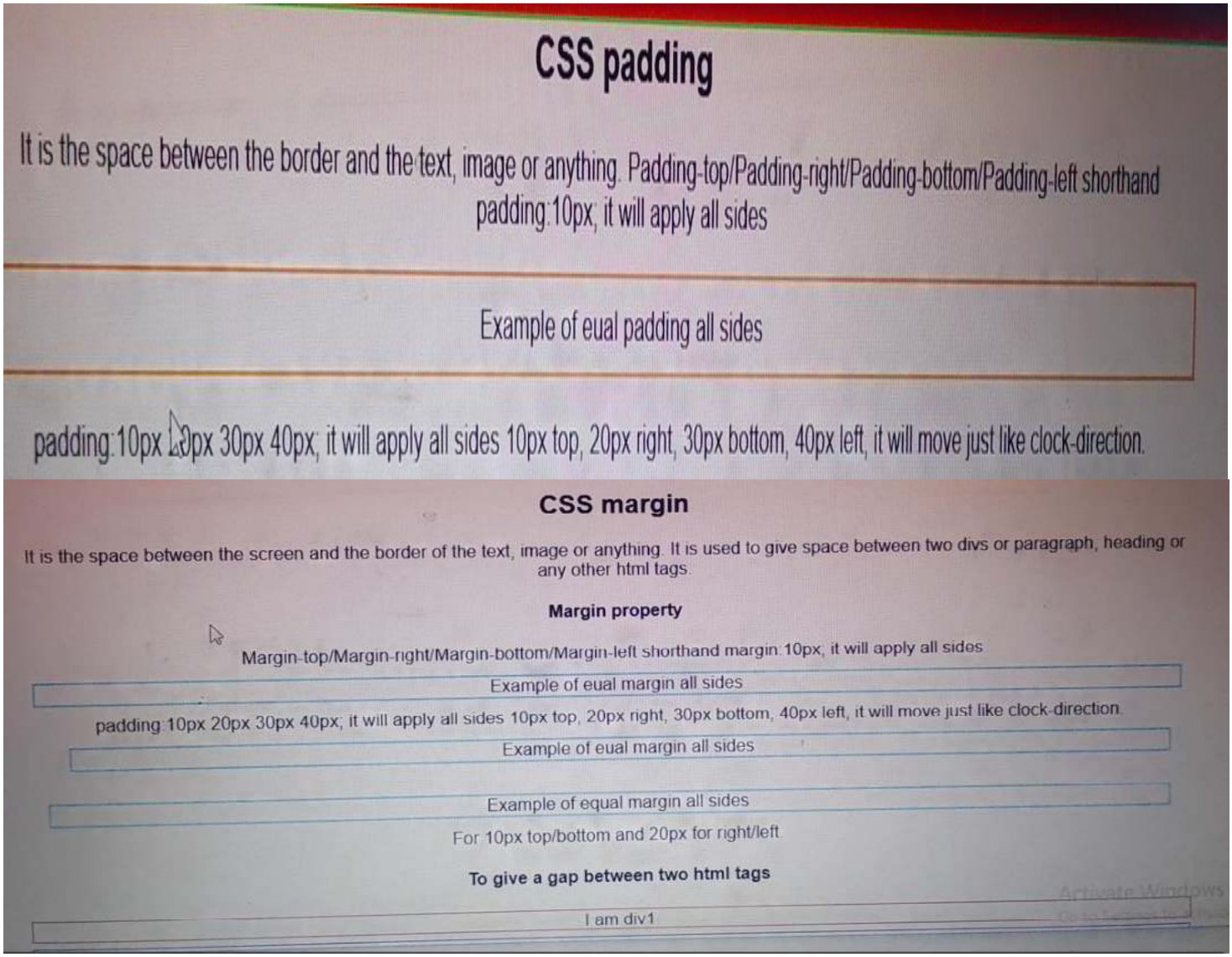
We have other tools as well. Like https://comparius.app/ https://browsershots.org/

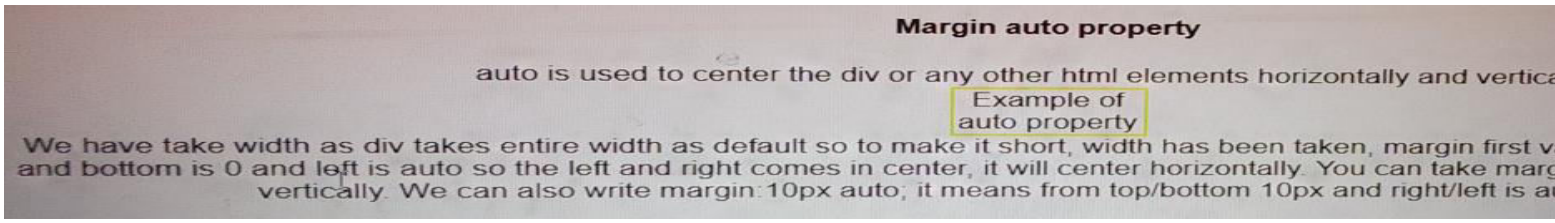
59. Define border, border-outline and its properties.

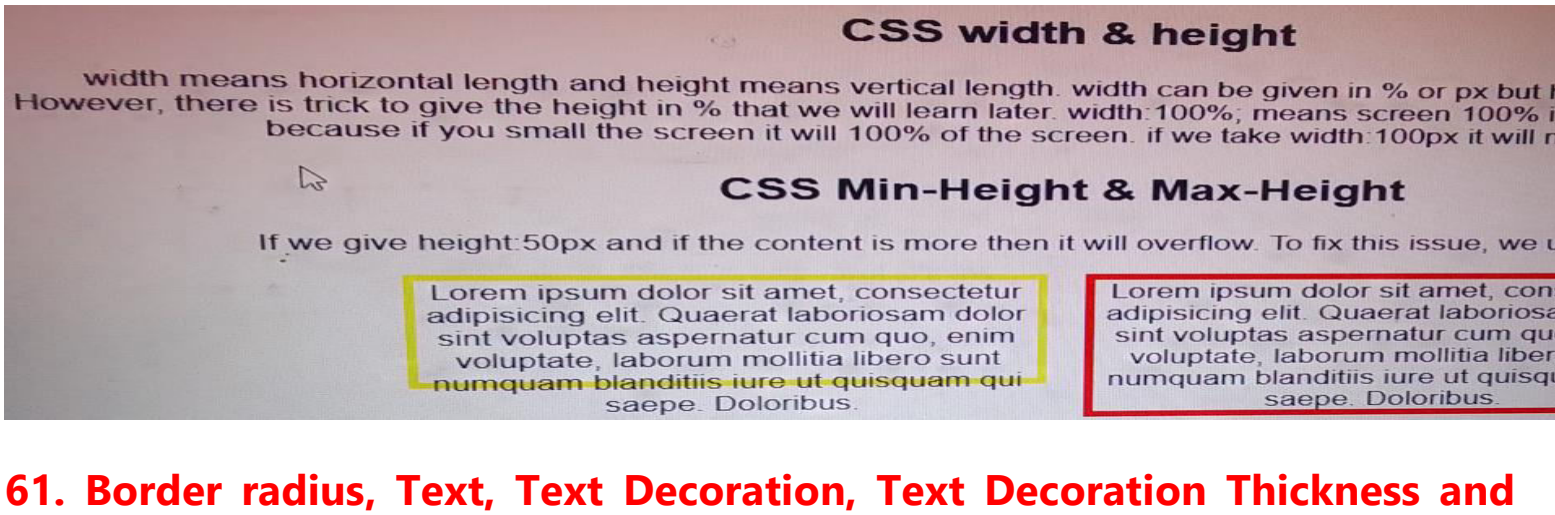




60. Define padding, margin, margin auto, min height and max-height.







61. Border radius, Text, Text Decoration, Text Decoration Thickness and Underline-offset.

CSS3 Border radius property

It is used to make curve in the corner. You can make a circle or ovel.

- border-top-left-radius
  width will not increase.
- border-top-left-radius
  It is a default property and width will increase.
- border-bottom-right-radius
  width will not increase.
- border-bottom-left-radius
  width will not increase.

shorthand:- border-radius:20px 30px 50px 10px; 20px for first(top-left), 30px for second(top-right) 50px for third(bottom-right)& 10px for the fourth one(bottom-left)

We use prefix to support the border-radius property

We use -webkit-border-radius:10px; It will support in old browsers like chrome and safari and -mozkit-border-radius:10px; It will support in old browsers Like mozillafirefox as border-radius is a new property.

We can give the value of border radus in percent as well.

It will become a circle when height&width are same and border-radius is 50%.

TEXT

Font-size is used to show how big you want to show the font. fontfamily meas which font you want to apple like arial,verdana etc. font-weight means

CSS Text Formatting Properties Tutorial in Hindi / Urdu

CSS Text Properties

- Text-align
- Text-align-last
- Text-transform
- Text-indent
- Word-spacing
- Letter-spacing

text-align-last:left, right, center, justify

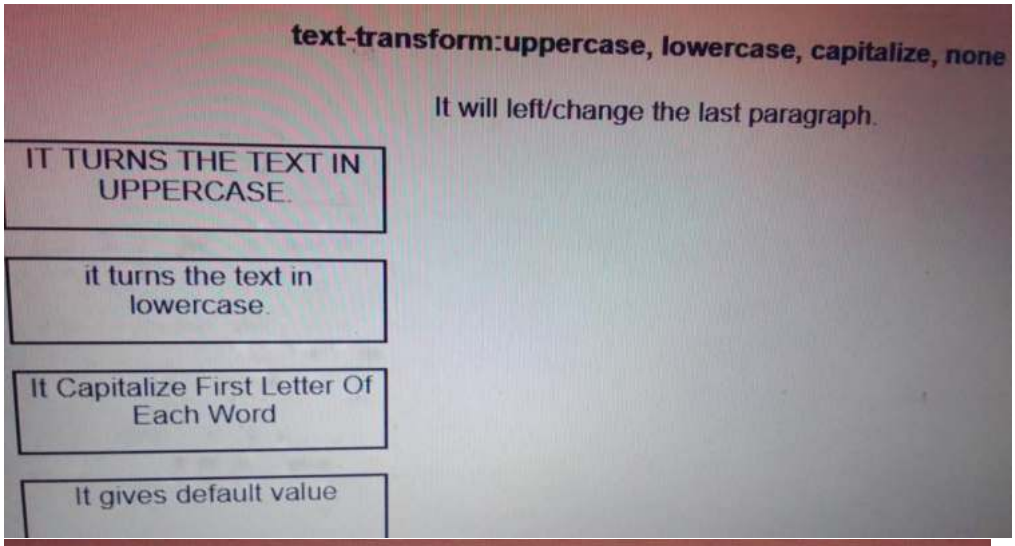
It will left/change the last paragraph

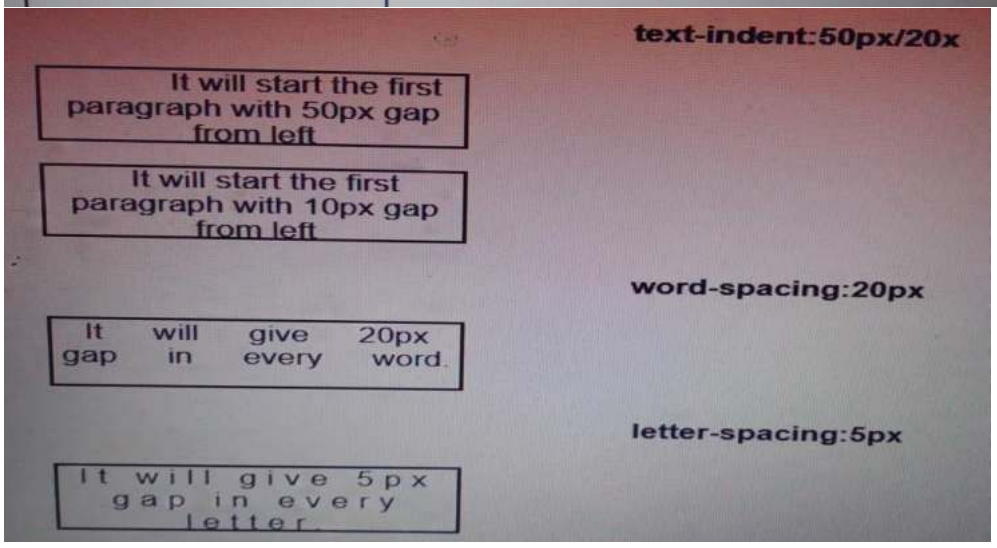
My last paragraph is on the left hand side

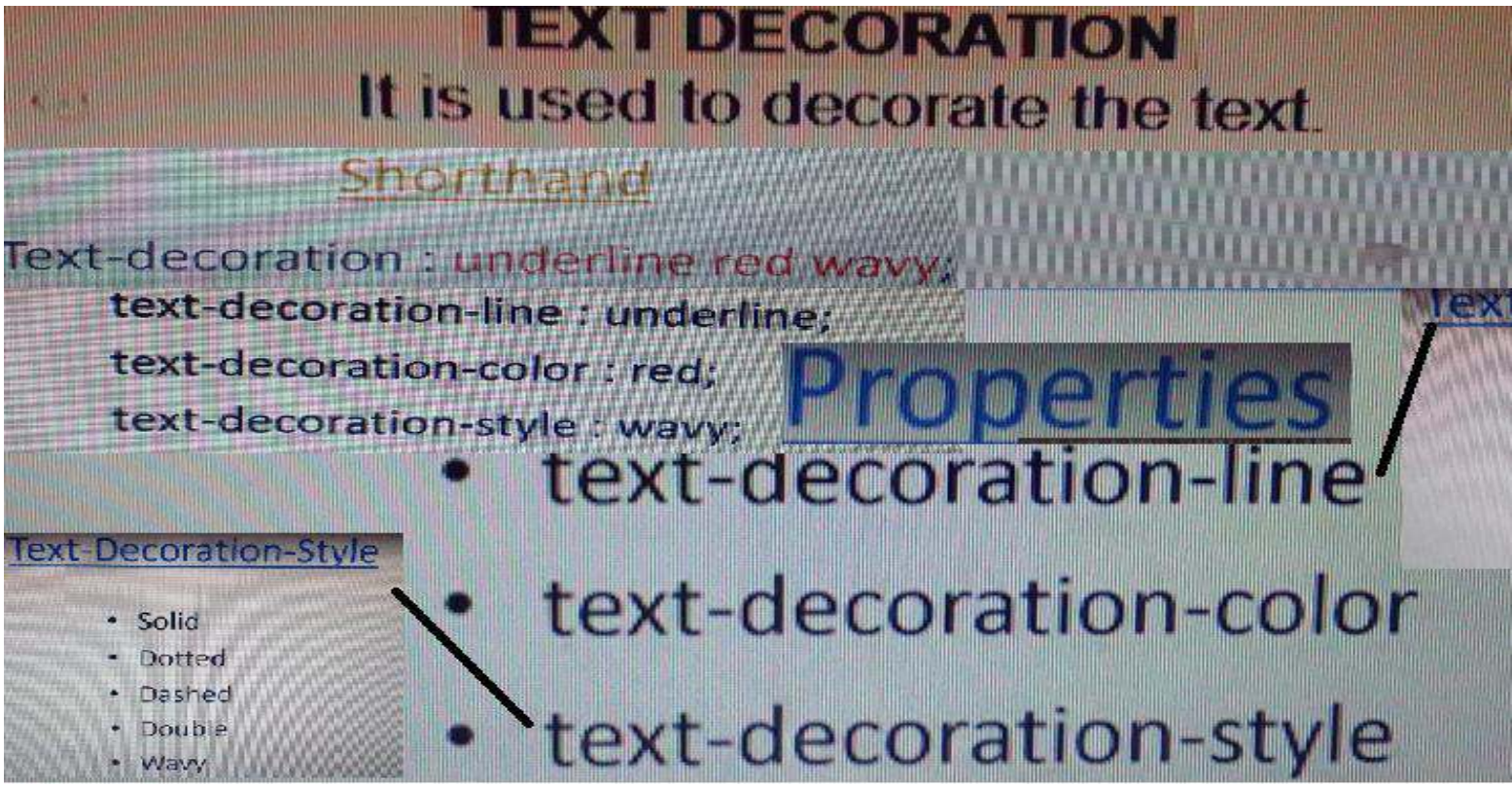
My last paragraph is on the right hand side

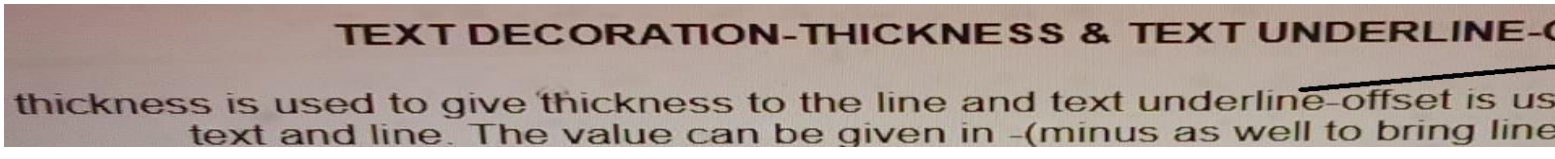
My last paragraph is on the coter

My last paragraph is on the center as it is auto

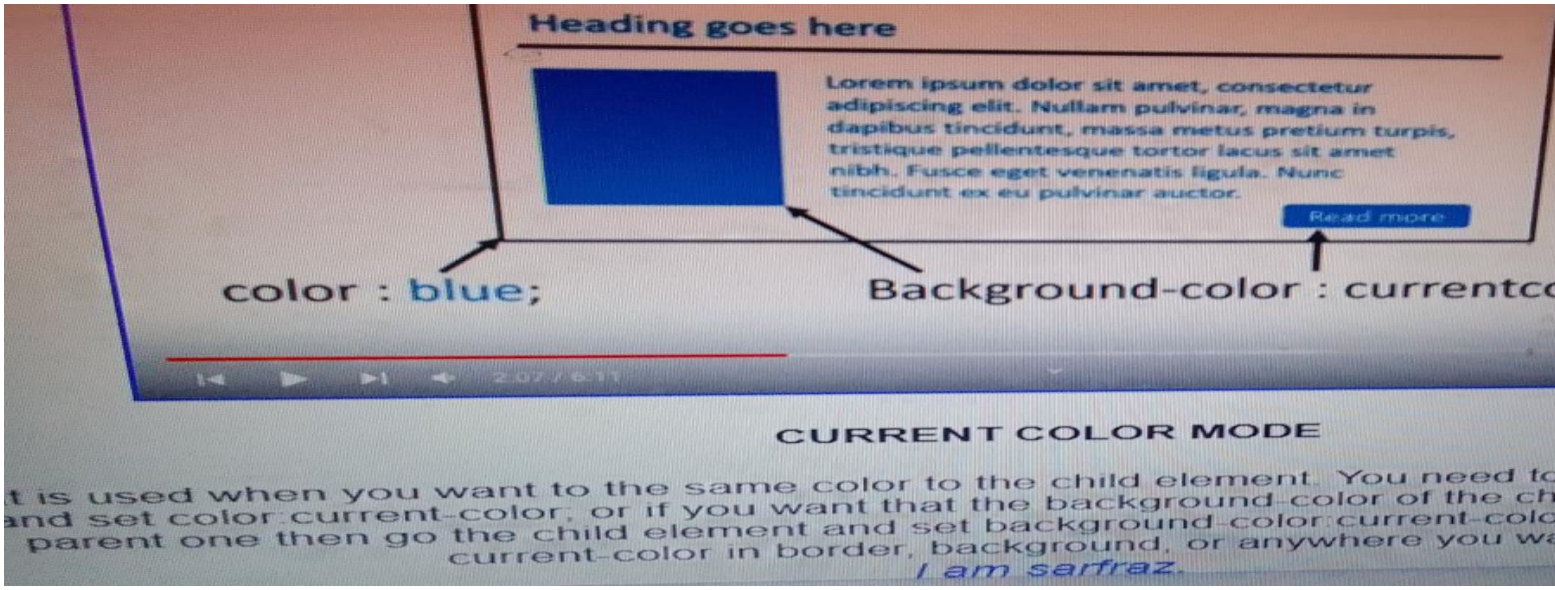

text-transform:uppercase, lowercase, capitalize, none
It will left/change the last paragraph.
IT TURNS THE TEXT IN UPPERCASE.
it turns the text in lowercase.
It Capitalize First Letter Of Each Word
It gives default value

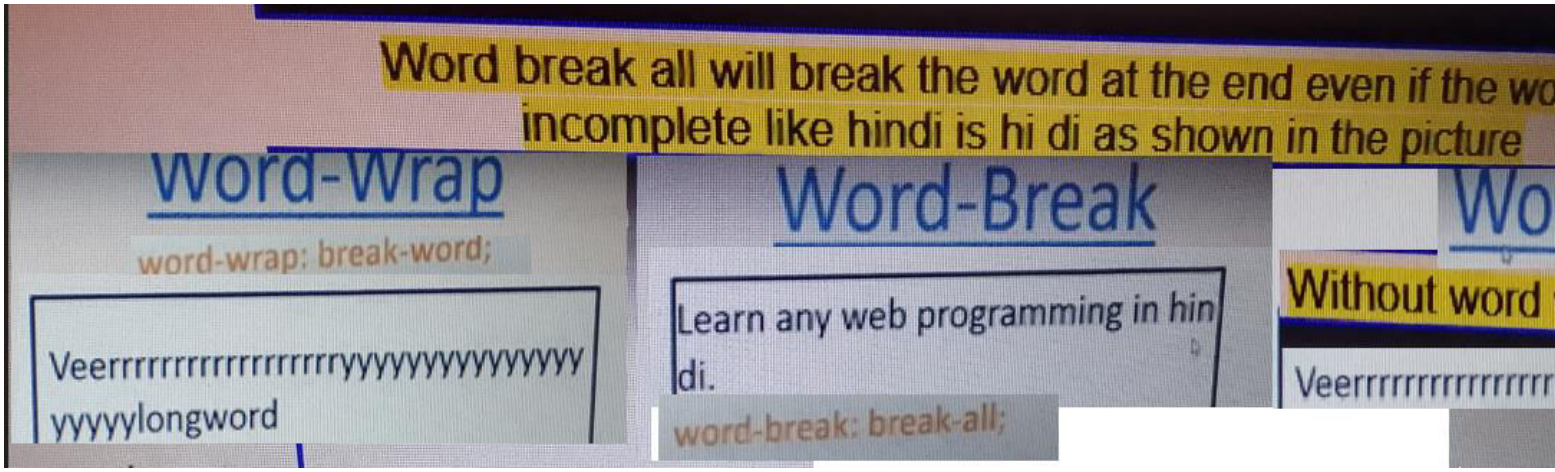

text-indent:50px/20x
It will start the first paragraph with 50px gap from left
It will start the first paragraph with 10px gap from left
word-spacing:20px
It will give 20px gap in every word.
letter-spacing:5px
It will give 5px gap in every letter.

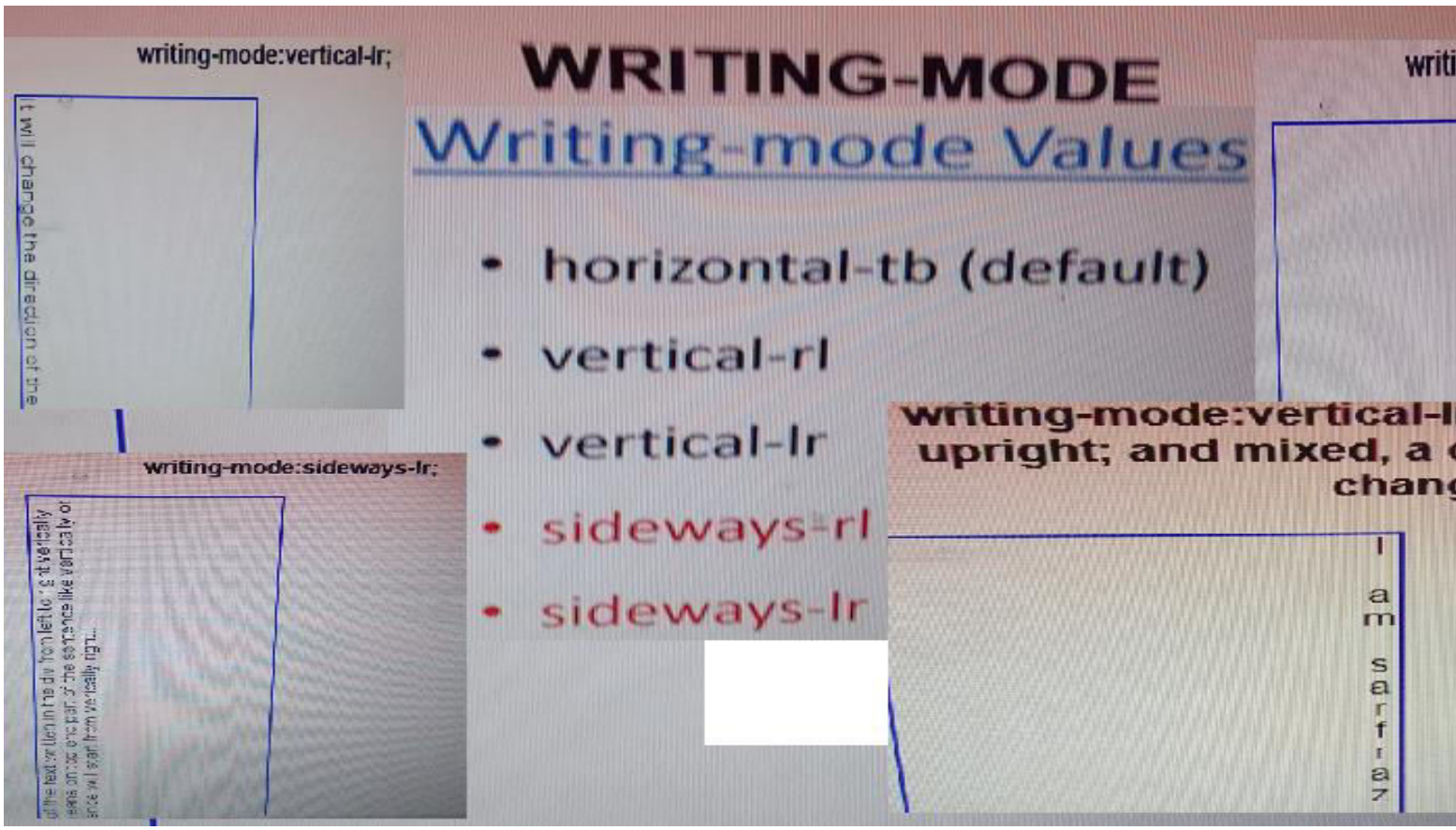

TEXT DECORATION
It is used to decorate the text.
Shorthand
Text-decoration : underline red wavy;
text-decoration-line : underline;
text-decoration-color : red;
text-decoration-style : wavy;
Properties
• text-decoration-line
• text-decoration-color
• text-decoration-style
Text-Decoration-Style
• Solid
• Dotted
• Dashed
• Double
• Wavy

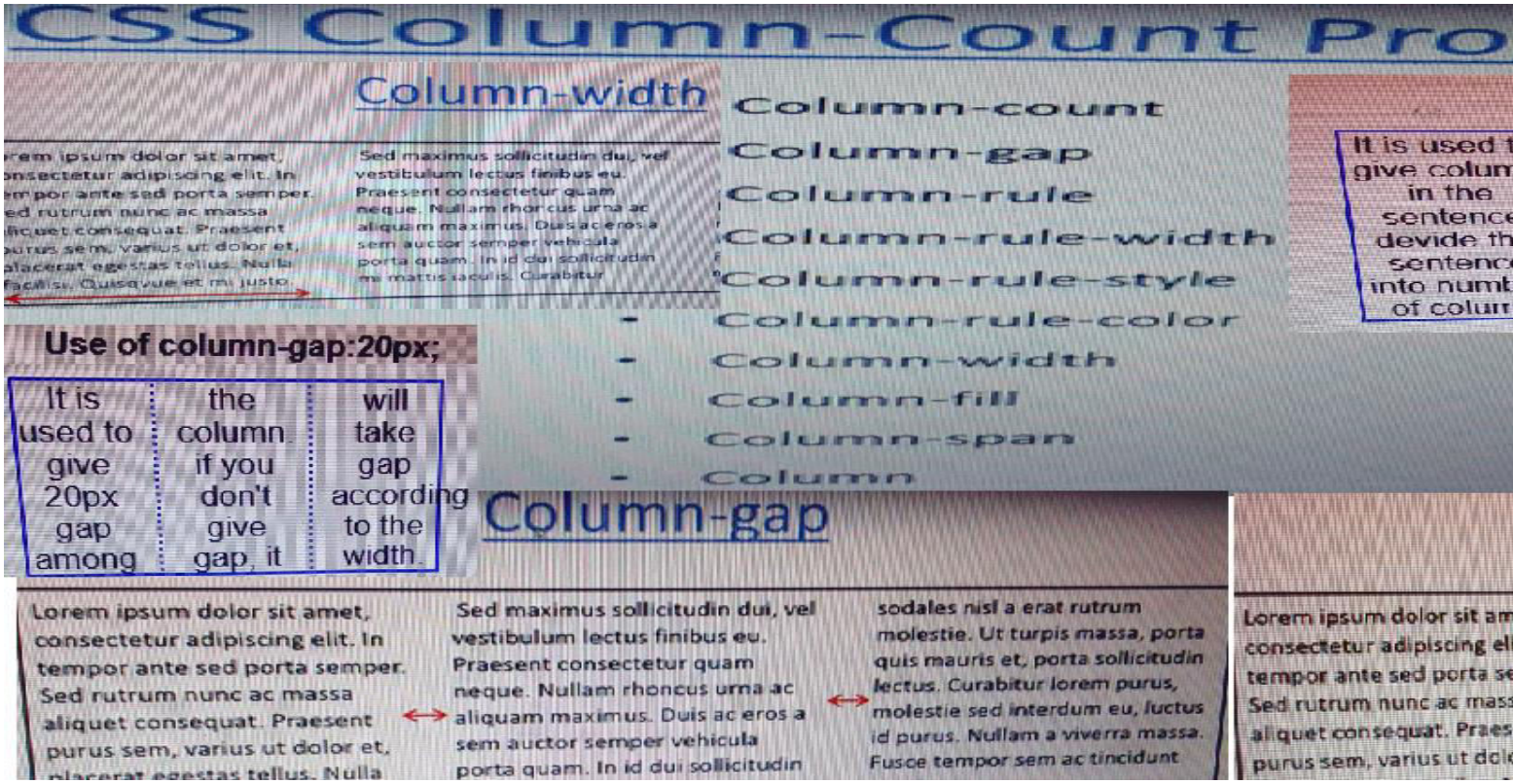

TEXT DECORATION-THICKNESS & TEXT UNDERLINE-
thickness is used to give thickness to the line and text underline-offset is us
text and line. The value can be given in -(minus as well to bring line

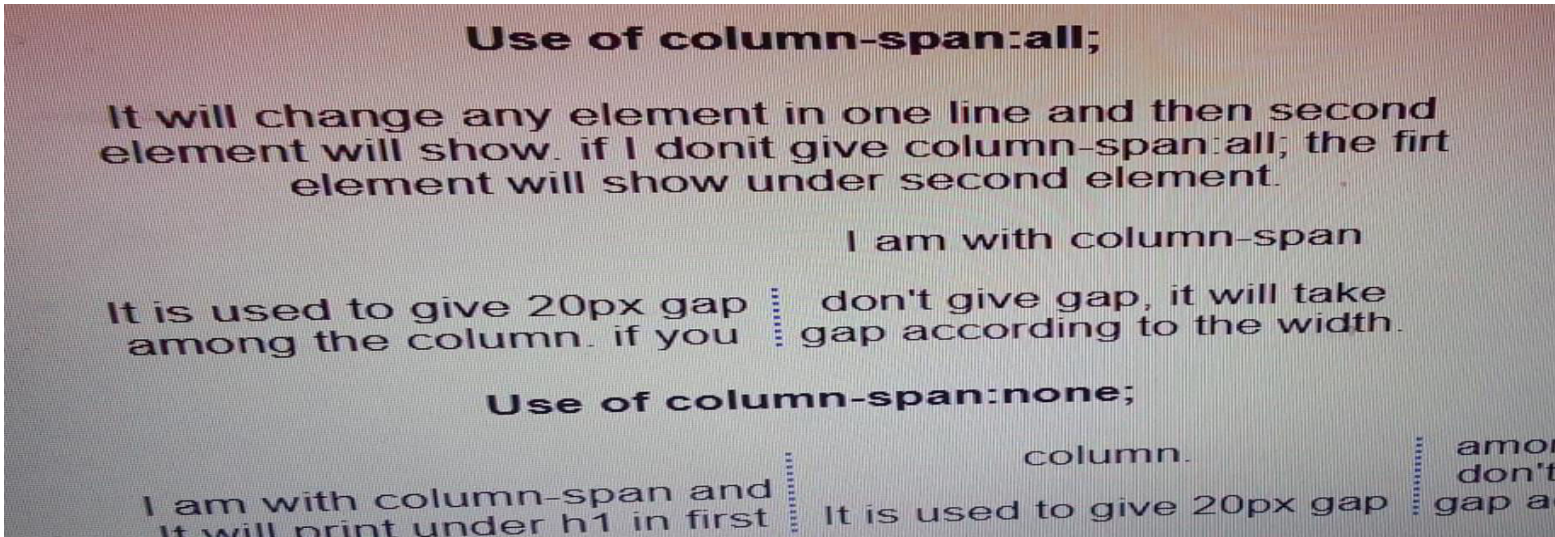
62. Define Current Color Mode, Word Wrap, Word Break, Writing Mode and Column Count.











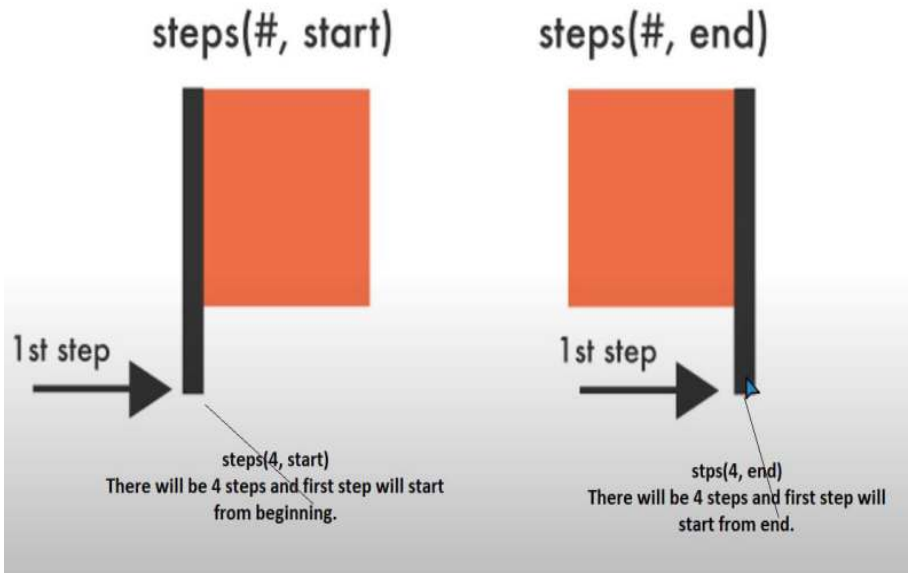
63. Define Transition.

It is used to give smoothness when you hover, without it if we apply, it will jerk.
transition:width, height, background 2s, 2s, 2s; is shorthand of transition-property.
transition:all 2s: all will apply for width, height etc. and 2s if all the duration is same.
transition-delay means after how long hover should start or end.

```
    <head>
        <style>
            div{
                background-color: orange;
                width: 100px;
                height:100px;
                /* transition-property:width, height;
                transition-duration: 2s;
                transition-timing-function:steps(4);
                transition-delay:1s; */
                transition:all 2s steps(4) 1s;  // Shortcut transition: all means all the property taken  transition-duration
transition-timing-function transition-delay
```

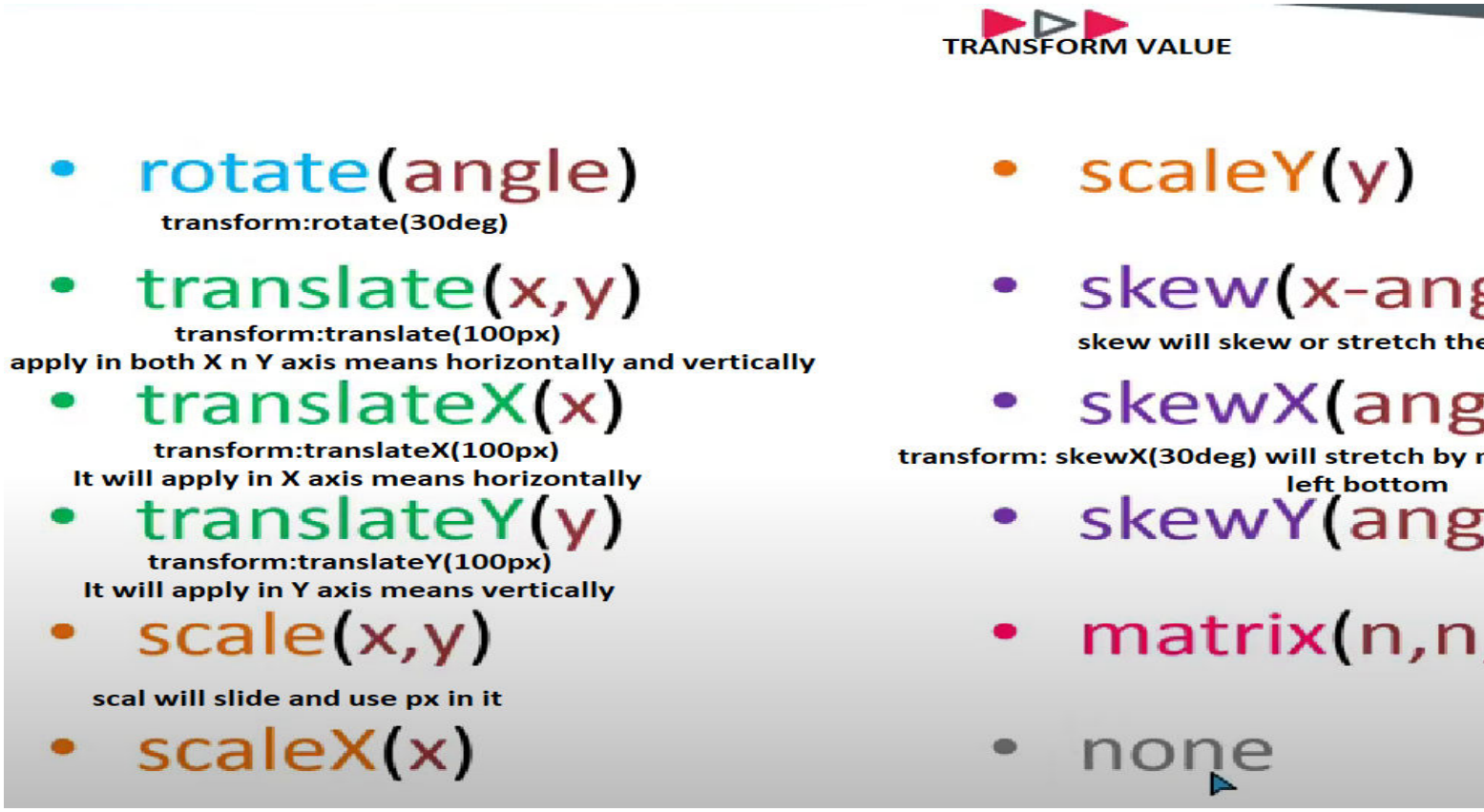
```
            }
            div:hover{
                background-color: rgb(0, 255, 166);
                width: 200px;
                height:200px;}
        </style>
    </head>
<body>
    <div>
    </div>
</body>
```

Unit	Description
ease	slow start, then fast, then end slowly
linear	same speed from start to end
ease-in	slow start
ease-out	slow end
ease-in-out	slow start and end — normal speed will be in the middle
step-start	animation will go where it has to start from at one step
step-end	animation will go where it has to end in single step
steps(4, end)	We can take more that 4 steps or any steps you want
cubic-bezier	lets you define your own values — Here only 4 steps allowed



64. Define Transform.

It is used to rotate the div or any image. transform: rotate(30deg);will rotate 30degree. degree can be written in minus as well. transform: (30deg);



The above are 2D values of transform, we have the same values for 3D but Z index will be added.

If the width and height of two divs are same and if we use transform: rotateX(60deg); in the child div then then child div will act like a curtain closing horizontally.

perspective gives a 3d look and also shows how it is moving. the more is the perspective(800px) the smaller area it will spread and the better 3d will look as when we sit in the cinema hall and watch from the balcony, it gives clearer than sitting at front. perspective is used in 3d only. perspective(100px)will have bigger spread and perspective(800px) will have smaller. perspective-origin has default value (center center) horizontally & vertically.

```
    <head>
        <style>
            div{
                background-color: orange;
                width: 100px;
                height:100px;
                transition:all 2s linear 0s;}
            div:hover{
                transform: rotate(360deg); /* It will rotate the div 360 degree */
                /* transform: rotate(-30deg);  to move anticlock wise */
            }
        </style>
    </head>
<body>
    <div>Hello</div>
</body>
```


```
        <style>
            div{
                background-color: orange;
                width: 100px;
                height:100px;
```

```
            transition:all 2s linear 0s;
            margin: 300px auto;  /* to fix in center */
        }
        div:hover{
            transform: rotate(360deg) scale(2);
             /* It will rotate the div 360 degree and also double the size*/
        }
    </style>
```

transform-origin

Earlier the point of the div was center from where the div was being moved. Now, we can change the origin like left-bottom(x asix(horizontal) se left and y axis(vertical) se bottom)and right top etc. It can be given in % as well like transform-origin:100% 20%;

```
<head>
    <style>
        div{
            background-color: orange;
            width: 100px;
            height:100px;
            transition:all 2s linear 0s;
            margin: 300px auto;  /* to fix in center */
        }
        div:hover{
            transform: rotate(360deg) scale(2);
            transform-origin:left center ;
             /* It will rotate the div 360 degree and also double the size and also chnage the direction*/
        }
    </style>
</head>
<body>
    <div>Hello</div>
</body>
```

It is used to write all the values in the sequence as given below. We can also call it shorthand.

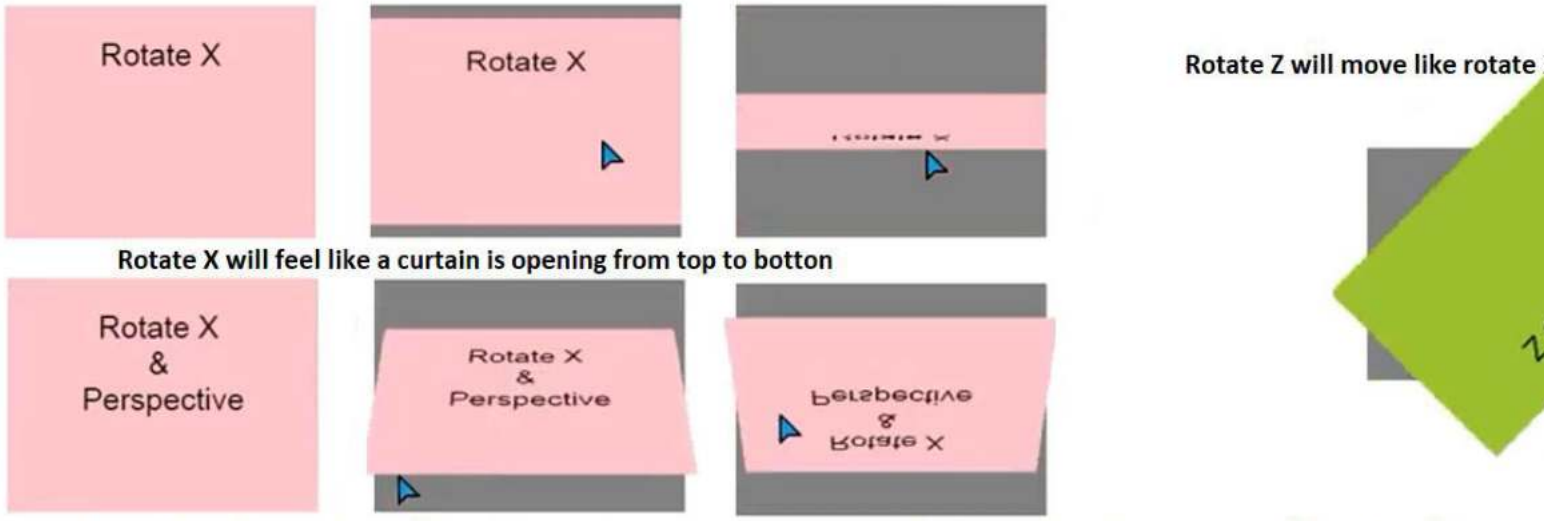
matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateX())

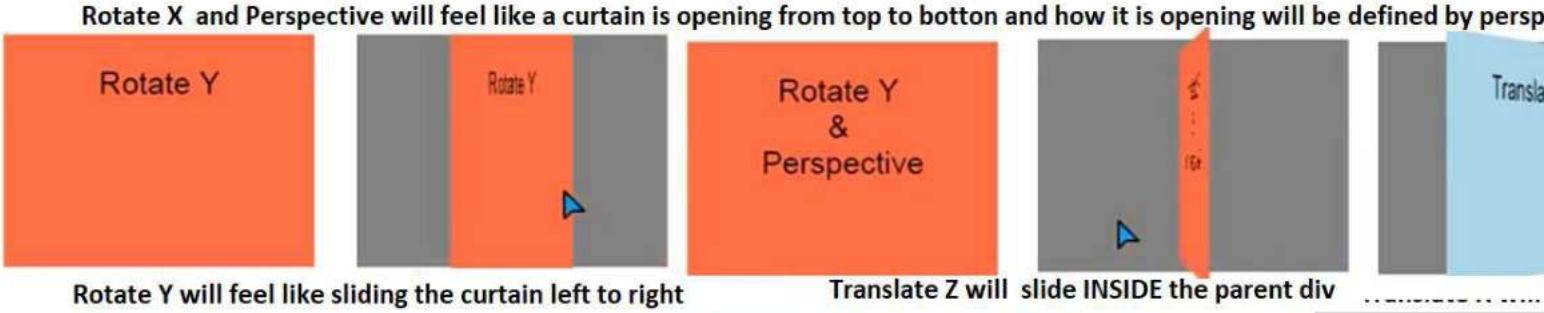
CSS Transform – 3D Values

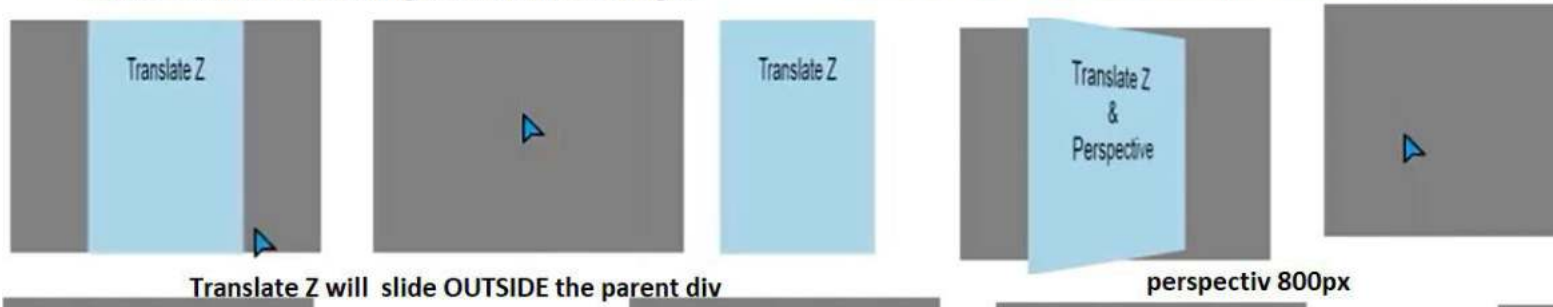
- rotateX(angle)

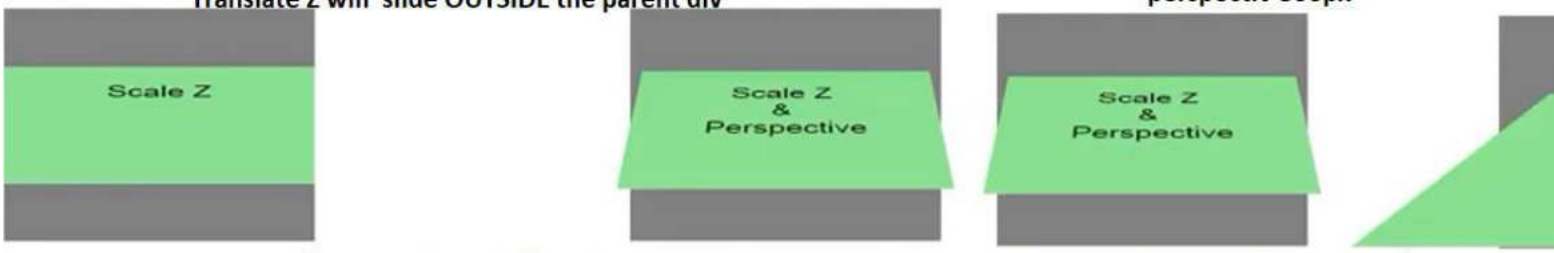
We use rotate X(30deg) in the child div to display a curtain closing top to bottom and this will be applicable when both parent andchild have the same width and height.

- rotateY(angle)
- rotateZ(angle)
- rotate3d(x,y,z,angle)
- translateZ(z)
- translate3d(x,y,z)
- scale3d(x,
- scaleZ(z)
- perspectiv
- matrix3d(
  n,n,n,n,n,
- none

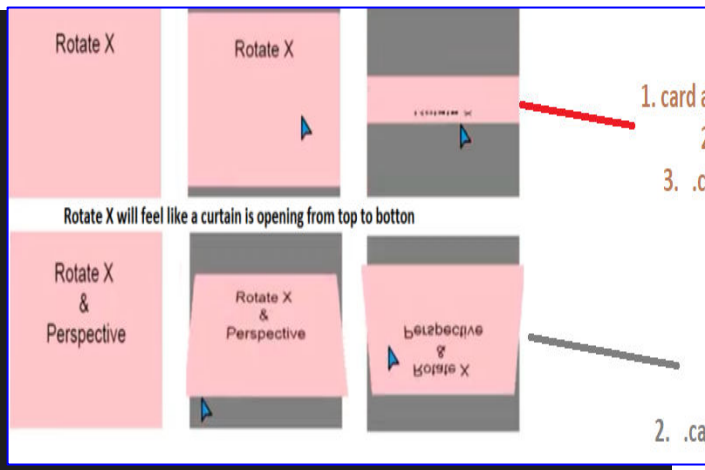






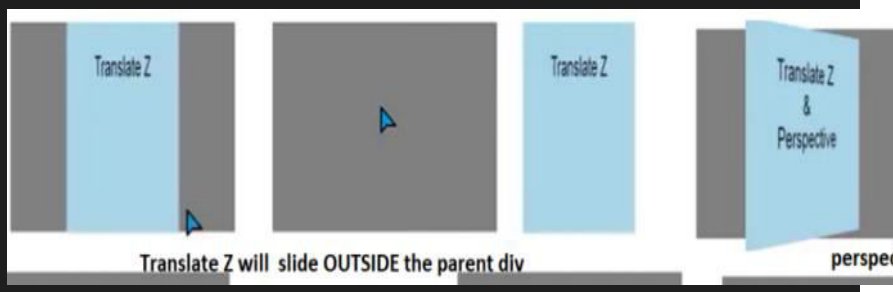



```
<head>
    <style>
        .wrapper {
            background-color: orange;
            width: 100px;
            height: 100px;
            margin: 300px auto;
            /* to fix in center */
        }
        .card{
            background-color: rgb(47, 0, 255);
            width: 100px;
            height: 100px;
            transition: all 2s linear 0s;
        }
        .card:hover {
            transform:perspective(800px) rotateX(360deg);
          // perspective used to show the 3d look, we can use rotateY or Z instead of X to get the output as mentioned above.

        }
    </style>
```

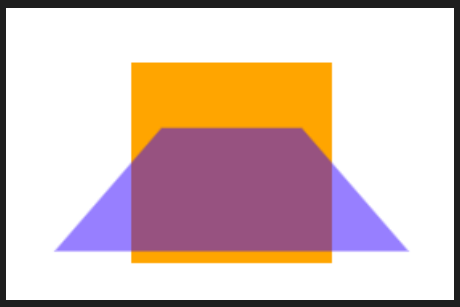


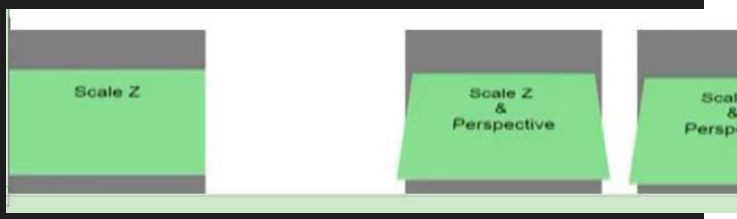
```
    </head>
<body>
    <div class="wrapper">
        <div class="card">
        </div>
    </div>
</body>
```

```
<head>
        <style>
            .wrapper {
                background-color: orange;
                width: 100px;
                height: 100px;
                margin: 300px auto;
                /* to fix in center */
            }
            .card{
                background-color: rgb(47, 0, 255);
                width: 100px;
                height: 100px;
                transition: all 2s linear 0s;
                opacity:0.5;
            }
            .card:hover {
                /* transform:translateX(100px);  It will slid 100px leftwards
                transform:translateY(100px);   It will slid 100px downwards*/
                transform:   perspective(800px) rotateY(60deg)  translateZ(100px);
                /* We need to use perspective and rotate with translate Z  */
            }
        </style>
```




```
 <head>
        <style>
            .wrapper {
                background-color: orange;
                width: 100px;
                height: 100px;
                margin: 300px auto;
                /* to fix in center */
            }
            .card {
                background-color: rgb(47, 0, 255);
                width: 100px;
                height: 100px;
                transition: all 2s linear 0s;
                opacity: 0.5;
            }
            .card:hover {
                /* transform:scaleX(3); */
                /* It will tripe the size horizontally */
                /* transform:scaleY(3);    */
                /* It will tripple the size vertically */
                transform: perspective(100px) rotateX(60deg) scaleZ(3);
                /* We need to use perspective and rotate with scale Z  */
            }
        </style>
    </head>
<body>
    <div class="wrapper">
        <div class="card">
        </div>
    </div>
```

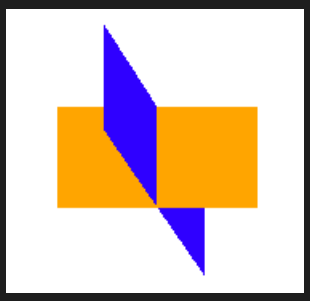




TRANSFORM-STYLE: PRESERVE 3D

It need to be applied in parent div and also it will work with transform: rotateY(30deg); or rotateX() for horizontally. It has two values, first is flat(default) and second is transform style:preserve-3d

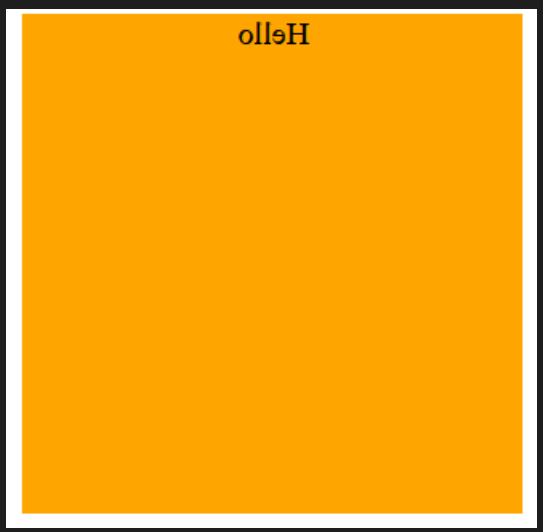
```
    <head>
        <style>
            .wrapper {
                background-color: orange;
                width: 100px;
                height: 100px;
                margin: 300px auto;
                transform: rotateX(60deg);
                transform-style: preserve-3d;
                /* to fix in center */
            }
            .card {
                background-color: rgb(47, 0, 255);
                width: 100px;
                height: 100px;
                transition: all 2s linear 0s;
                transform: perspective(800px) rotateY(60deg);
                /* transform-origin:left center; */
            }
        </style>
    </head>
```



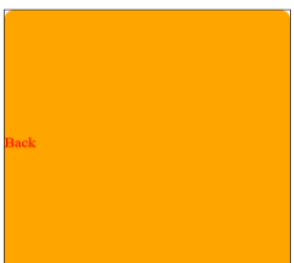
BACKFACE

It needs to be applied in parent div and also it will work with transform: rotateY(30deg); or rotateX() for horizontally. It helps to show the back of any image/div when we rotate any div by 180deg in 3D/2D. We will be able to the back when we rotate any div by 180deg. In the given example if we set backface-visibility:hidden; the backface does not appear when we hover but when backface-visibility:visible; the backface appears as you can see.
You can see, the back is visible when it is hovered and it is because backface-visibility:visible;

```
<head>
        <style>
            #backface{
                text-align: center;
                width: 250px;
                height: 250px;
                background: orange;
                backface-visibility: visible;
                transition: transform 2s;
            }
            #backface:hover {
                transform: rotateY(180deg);
            }
        </style>
    </head>
<body>
    <div id="backface">
        <p>Hello</p>
    </div>
</body>
```




```
    <head>
        <style>
            #backface:hover {
                transform: rotateY(180deg);
            }
          .card:hover .content {
```



```
            transform: rotateY(180deg);
        }
    </style>
</head>
<body>
    <div class="card" style="width: 300px; height: 300px; border: 1px solid black; perspective: 500px;">
        <div class="content" style="width: 100%; height: 100%;
                transform-style: preserve-3d; transition: transform 1s;">
            <div class="font" style="width: 100%; height: 100%; position: absolute; background: yellow;
                    color: blue;backface-visibility: hidden;line-height: 300px; border-radius: 10px;">Front
            </div>
            <div class="back" style="width: 100%; height: 100%; position:absolute;background:orange;
                    color:red; transform: rotateY(180deg); backface-visibility: hidden;
                    line-height: 300px;border-radius: 10px">Back
            </div>
        </div>
    </div>
</body>
```

Explanation of the picture

fornt- hidden karne se front ka back nahi dikhega becuase I need only front of the form and back to show and this is why both front & Back were hidden.
line-height: 300px to align vertically and 300px is parent dive height. Back-hidden will hide the back of the hidden and front will appear because back ke pickhe ke badd front hai. To see differnt movement.

```
<a href="https://css-tricks.com/almanac/properties/t/transform/">To see differnt movement. </a>
```

CSS MCQ

1. Which CSS property allows you to hide an element but still maintain the space it occupies on the web page? visibility:hidden

2. Which of the following is NOT a valid CSS length unit? cm em dm All the above

3. What is the CSS selector which allows you to target every element in a web page? Universal selector denoted by *

4. What would be the color of text “I am awesome” for the following rules?

```
    <style>
        #awesome {
            color: red;
        }
        .shopping-list li.favorite span {
            color: blue;
        }
    </style>
</head>
<body>
    <ul class="shopping-list" id="awesome">
        <li><span>Milk</span></li>
        <li class="favorite" id=”must-buy”>
            <span class=”highlight”>I am awesome </span>  // blue
        </li>
    </ul>
```

```
</body>
```

5. Which of the following CSS properties DOES NOT influence the box model? Outline

Outline

6. How will you select the anchor element whose href attribute starts with https?

a[href^="https"]

7. What CSS3 property is used for capitalizing the text or converting them to lowercase or uppercase?

text-transform

8. In the following piece of code, does the stylesheet2.css sheet has to be loaded and parsed before the first p tag is loaded?

```
<head>
        <link href="stylesheet1.css" rel="stylesheet">
        <!-- //This will load first then body and then sheet2 and the style of sheet2 will be applicable -->
    </head>
    <body>
        <p>Paragraph 1</p>
        <p>Paragraph 2</p>
        <link href="stylesheet2.css" rel="stylesheet">
    </body>
```

YesLoads after the paragraph are loaded

9. How would you select which anchor element will have href consisting of the substring "js"?

```
 <a href="js"></a>
```

a[href$="js"]

10. What property should be used in case we need to display a nice blue border that is dotted in nature around an image?

border-style

11. Which among the following options represent correct syntax for selecting all paragraph elements in a div element?

div p

12. What CSS3 property is used for the set distance between borders of adjacent tables cells?

border-spacing

13. Which of the below options are used for defining the difference between two lines of the content?

line-height

14. Which among the below property is used for setting the blend mode of background layers in an element?

background-blend-mode

15. Which among the below options are used for giving line over text?

text-decoration: overline

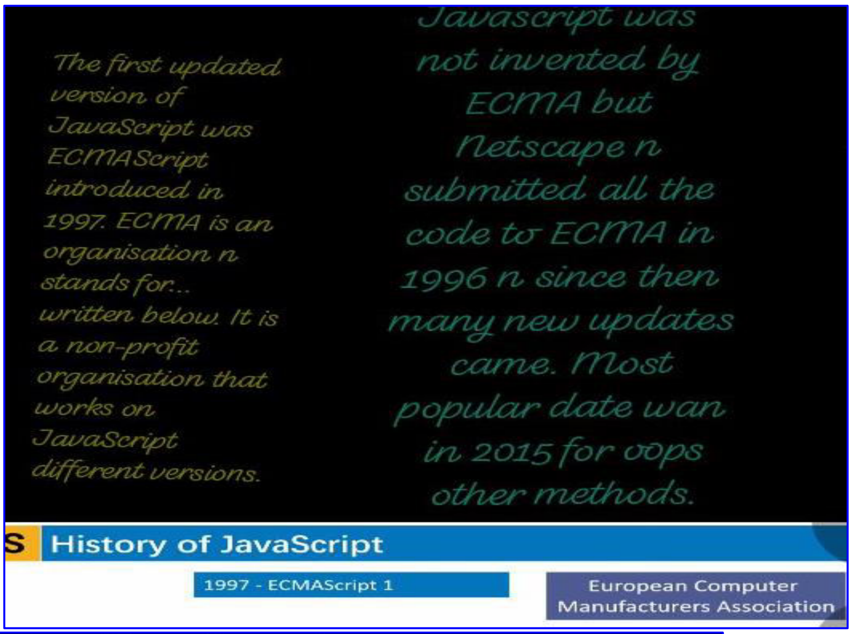
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**THE END**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

1. What do you understand about JavaScript and its history?

JavaScript is a scripting language. It is different from Java language. It is an object-based, lightweight, cross-platform translated language. It is used for client-side and server-side development. It is widely used for client-side validation. The JavaScript code can be inserted into HTML pages that can be understood and executed by web browsers because the JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

The browser does not interpret server side scripting language because it can interpret or execute HTML, CSS, DXTML and Client Side Script. We need to add an interpreter to interpret or execute server side script installed in the server called “Zen-engine”. PHP and .NET are Server Side Script.

Note- Server Side Script is not available at the server because it is sent from the browser.



Define ES6(2015) , ES7(2016) AND ES8(2017)

MAP FUNCTION

```
<script>
     let arr = [10, 20, 30, 40];
     let a = arr.map((num) => {
        return document.write(num)    // 10 20 30 40
     })

     document.write(a);
  </script>
  <script>
     let arr1 = [10, 20, 30, 40];
     arr.map((num, i) => {
        document.write(`${i} : ${num} ` + "</br>");  // 0: 10 1:20 so on
     })
  </script>
```

ES5 in 2009 [PART OF JAVASCRIPT]

('USE STRICT, SOME NEW ARRAY METHODS SUCH AS isArray(), filter(), map(), reduce(), reduceRight(), forEach(), some(), every(), indexOf(), lastIndexOf(), JSON METHODS SUCH AS parse(), stringify(), DATE NEW METHODS SUCH AS now() and valueOf() , GETTERS AND SETTERS like get() & set() methods and PROPERTY METHOD SUCH AS Object.defineProperty();

```
 <script>
        let ary = [2 ,3 , 4, 5];
        let output = ary.reduce(function(total, currentValue){
          document.write(currentValue); //2 ,3 , 4, 5
          console.log(currentValue);
          return total*currentValue;

        });
        document.write(output); //120
 </script>
```

```
<script>
     let arr = [10, 20, 30, 40];
     let a = arr.map((num) => {
        return document.write(num)    // 10 20 30 40
     })

     document.write(a);
  </script>
    Reducer has a call back function that iterate the each element of an array and reduce them to a new element by adding,
    mutiplying or dividing or any other methematical operation, it has two parameters, total and currecntValue
  <script>
        let ary = [2 ,3, 4, 5];
        let output = ary.reduceRight(function(total, currentValue){
          document.write(currentValue); //5, 4, 3, 2
          console.log(currentValue);
          return total*currentValue;

        });
        document.write(output); //120
 </script>
```

JSON SUPPORT

parse(): It parses a JSON string that is like an object.

```
let jsonString = '{"a": 1, "b": 2}';
    let obj = JSON.parse(jsonString);
    console.log(obj.a); //1
```

stringify(): This method converts an object to a JSON string.

```
console.log(JSON.stringify({ x: 5, y: 6 }));    //"{"x":5,"y":6}"
```

Object.defineProperty(): This method lets the user define the p roperty of an object and/or change its value.

ES6 in 2015 (17) [PART OF ADVANCE JAVASCRIPT]

arrow functions, Block scope(let and const keywords), Built-Ins, classes, default parameters, destructuring method, fetch, generators, rest operator, spread operator, set, weakSet, map, weakMap , modules, symbols,promises, template string and Object literals, Unicodes, proxies, Binary and Octal, reflect.

DEFAULT PARAMETERS: Provides default values to function parameters if no value or undefined is passed.

```
function fun(a, b, c=0) {
    console.log('a: ');
    console.log('b: ');
    console.log('c: ');
}
fun(2,3);       //prints a: 2 b: 3 c: 0
fun(2,3,1);     //prints a: 2 b: 3 c: 1
```

ES7 in 2016

Exponetial operator(**) inclues()

```
<script>
   let x = 10;
   let y = 2;
   document.write(x**y); // 100
</script>
<script>
          let x = [10, 20, 30];
          let y = x.includes(10);
          document.write(y);  // true
</script>
```

ES8 in 2017

Padstart(), padend(), sync/await(), object.entries(), tailing commas, object.values(), OBJECT.GETOWNPROPERTYDESCRIPTORS():

ES9 in 2018

ASYNCHRONOUS ITERATION, REGULAR EXPRESSION IMPROVEMENTS, REST/SPREAD PROPERTIES, PROMISE.PROTOTYPE.FINALLY()

0. Difference between client-side and server-side

Client-side scripting	Server-side scripting
Source code is visible to the user. The processing takes place on the user's computer or It runs on the user's computer. It usually depends on the browser and its version. We need browser and its version to run the client side scripting language. There are many advantages linked with this like faster. response times, a more interactive application. It does not provide security for data. It is a technique used in web development in which scripts run on the client's browser. HTML, CSS, and JavaScript are used. No need of interaction with the server.	Source code is not visible to the user because the output of server-side is an HTML page. The processing takes place on the web server or It runs on the web-server. In this any server-side technology can be used and it does not depend on the client. The primary advantage is its ability to highly customize, response requirements, access rights based on user. It provides more security for data. It is a technique that uses scripts on the web-server to produce a response that is customized for each client's request. PHP, Python, Java, Ruby are used. It is all about interacting with the servers.

It reduces load on processing unit of the server.	It surge the processing load on the server.

1. Implicit Type Coercion in JavaScript (in detail with examples) (koh·uh·shn)

When the value of one data type is automatically converted into another data type, it is called Implicit type coercion in JavaScript.

The conversion of data is called coercion. There are two types to do it. The first is done by JavaScript interpreter as much as it knows as designed and 2nd is JavaScript method that we can use to convert like String() to convert into string to any boolean or number.

The automatic conversion of JavaScript done by the interpreter called Coercion.

Number coercion

Var a = 5 + null will return 5(Number) (It will convert null into number and since null means nothing or zero so 5 was returned)

String coercion

Var a = “5” + null will return 5null; (It will convert null into string)

Var a = “5” + 2 will return 52 (It will convert 2 into string)

10+20+"30" will return 3030 because 10+20 will be 30. If there is numeric value before and after +, it treats as binary + (arithmetic operator). "10"+20+30 will return 102030 because after a string all the + will be treated as string concatenation operator (not binary +).

Var a = “5” - 2 will return 3 (It will convert “5” into number because minus operator is used to subtract only but plus operator was supposed to do both adding the number and adding the string. So, it was confused and converting the 2nd operator as per first operator).

Var a = “5” * 2 = 10 (we can only multiply with multiply operator)

Var a = “5” * “2” = 10 (we can only multiply with multiply operator)

Var a = “5” * “yes” = NaN(When the compiler see the multiplication operator, it converts “5” into number and then try to convert Yes into number but unable to do it as finds that YES is not a number so return NAN)

Similarly,

```
<script>
var a = "5" * "yes";
console.log(a);
</script>
5yes
<script>
var b = "5" * "no";
console.log(b);
        </script>
        NaN
```

```
<script>
var a = "5" / "yes";
console.log(a);
        </script>
        NaN
```

We have three function methods to convert data type manually provided by JavaScript

1. String()[Number to String]

```
<script>
            var a = 5;
            var b = String(a);
            document.write(typeof(b));  // string
            document.write(String(a));  // 5
</script>
```

2. Number() [String to Number]

```
var a =”5”;
console.log(Number(a));
var b =”Hello”;
console.log(Number(b)); It will be number (NaN)
```

3. Boolean()

```
var a =1; // any other number except 0 will return Yes
console.log(Boolean(a));   It will be Yes
var b =0;
console.log(Boolean(b));   It will be False
```

```
<script>
            var a = 0;
            var b = Boolean(a);       // converted to boolean
            console.log(typeof(b));  // boolean
            document.write(b);  // false
</script>
```

```
var c =” ”  //empty string
console.log(Boolean(c));   It will return False
var d =”sarfraz”  //any statement or  string
console.log(Boolean(d));   It will return True
```

```
const z = "5";
document.write(Boolean(z));    // true
```

2. What’s the difference between JavaScript and Java?

JavaScript	Java
JavaScript is an object-oriented scripting language.	Java is an object-oriented programming language.
JavaScript applications are run inside a web browser.	Java applications are usually used in operating systems and virtual machines.
JavaScript needs compiler before running the application code.	Java does not need compiler before running the application code.

3. What is the difference between JavaScript and JScript?

Netscape provided the JavaScript language. Microsoft changed the name and called it JScript to avoid the trademark issue. In other words, you can say JavaScript is the same as JScript, but Microsoft provides it.

4. What are the various data types that exist in JavaScript?

To know the type of a JavaScript variable, we can use the typeof operator.

The data type of JavaScript can be divided into two types.

Primitive and Non-Primitive

1. Primitive types

Primitive data types can store only a single value.

String - It is used to write characters and alphanumeric values with quotes. A string can be represented using a single or a double quote.

Example :

```
var str = "Vivek Singh Bisht";          //using double quotes
var str2 = 'John Doe';                  //using single quotes
var str3 = 'John Doe99';
typeof "John Doe"                                   // Returns "string"
```

```
<script>

var a = "sarfraz";
document.write(typeof(a));
</script>
```

Number - It represents a number and can be written with or without decimals.

These are the different types of data that JavaScript supports:

Example :

```
var x = 3; //            without decimal
var y = 3.6;               //with decimal
typeof 3.14                                   // Returns "number"
```

BigInt - This data type is used to store large integers/numbers. It should end with n

Example :

```
var bigInteger =  234567890123456n;
```

```
<h4>bigin(biginteger)</h4>
<script>
document.write(typeof(2345678901234567n))
</script>
```

Boolean - It represents a logical entity and can have only two values : true or false. Boolean are generally used for conditional testing.

Example :

```
var a = 2;
var b =  3;
var c =  2;
(a == b)                // returns false
(a == c)                 //returns true
typeof true                                   // Returns "boolean"
```

```
<script>
var a = 10;
var b = 20;
var c = (a==b)
document.write(typeof(c));
</script>
```

Undefined - When a variable is declared but not assigned/defined/initialized.

The value of undefined is undefined and it's type is also undefined.

Example :

```
var x;                                  // value of x is undefined
var y = undefined;          // we can also set the value of a variable as undefined
```

```
<h4>Undefined Example</h4>
<script>
var cat;
var catt=undefined;
document.write(typeof(cat)+ "<br>");
document.write(typeof(catt));
</script>
```

Null - It represents a non-existent or an invalid value or used for empty or unknown values.

Example :

```
var z = null;
typeof(null);               // Returns "object" (kind of a bug in JavaScript)
```

```
<h4>null Example</h4>
<script>
var n=null;
document.write(typeof(n));
</script>
```

Symbol - It is a new data type introduced in the ES6 version of JavaScript. It is used to store an anonymous and unique value.

Example :

```
var a = Symbol;                                 // It will return function
var b = Symbol('sarfraz');              // both b and c  will return symbol
var c = Symbol();
```

```
<h4>symbol Example</h4>
<script>
var s=Symbol;
var s2=Symbol();
document.write(typeof(s) + "<br>");
document.write(typeof(s2));
</script>
```

Note- It is important to remember that any data type that is not a primitive data type, is of Object type in JavaScript.

2. Non-primitive types

Primitive data types can store only a single value. To store multiple and complex values, non-primitive data types are used.

Object and Array are example of Non-primitive types.

Object - Used to store collection of data in key-value pairs.

Example:

```
var obj1 = { x:  43,  y:  "Hello world!",  z: function(){return this.x;}  }
```

Array - Used to store collection of data in an ordered list

```
var array1 = [5, "Hello", true, 4.1];
```

5. What are the features of JavaScript?

These are the features of JavaScript:

- Open-source
- Object-oriented
- Cross-platform compatible
- Interpreted programming language
- Integration with other back-end and front-end technologies
- Lightweight
- Used especially for the development of network-based applications

6. What are the advantages of JavaScript over other web technologies?

These are the advantages of JavaScript:

Speed. Since JavaScript is an 'interpreted' language. it reduces the time required by other programming languages like Java for compilation. ...

Enhanced Interaction - JavaScript allows to contract with static web pages and makes them react to users' inputs.

Quick Feedback - There is no need for a web page to reload when running JavaScript. For example, form input validation.

Rich User Interface - JavaScript helps in making the UI of web applications look and feel much better.

Countless frameworks and libraries - JavaScript has countless frameworks and libraries that are extensively used for developing web applications and games of all kinds.

7. List some of the disadvantages of JavaScript.

Some of the disadvantages of JavaScript are:

- No support for multi-threading.
- It will stop the rendering if you make any mistake in the code and difficult to find.
- No support for multi-processing
- JavaScript codes are always visible to everyone.(by control+U, inspect, F12) so it is not secured.
- We know that the code of JavaScript is executed on the user's or client machine and it some case it is used for

  Malicious purposes and because of this some people disable the JavaScript.
- No support for networking applications

8. How do you create an object in JavaScript?

```
//Using Object literal
```

```
<script>
    var b = {
        firstName: "Sarfraz",
        lastName: "Alam",
        myBro: ["Kousar", "Quaisar"],
        salary: function () { return 8000; },
        fullName: function () { return this.firstName + " " + this.cityName },
        school: { schoolHall: 5, schoolTeacher: "Pammi" }
    }
    console.log(b);  // It will return entire object in console.
    document.write(b + "<br>");  // It will return [object object] So, it will not print
    document.write(b.firstName + "<br>");
    document.write(b['myBro'] + "<br>");// To print myBro value which is an array or To print array value in an object
    document.write(b.myBro[1] + "<br>");// To print myBro first value
    document.write(b.salary() + "<br><");// To print salary which is function. we need to take () sign as function used
    document.write(b.school.schoolHall + "<br>");//Use of another Object in an Object print or To print object value in an object
</script>
```

// Second Way To Create Object using create method

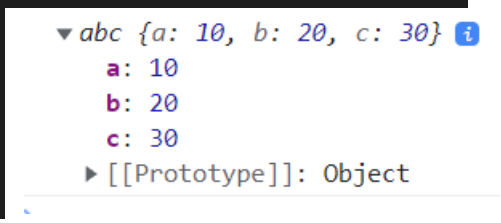
```
<script>
    var family = new Object();
    family.father = "Md Reazuddin"
    family.mother = "Manower"

    console.log(family);  // Object { father: "Md Reazuddin", mother: "Manower" }
    document.write(family.mother + "<br>");  //Output Manower
    document.write(family.father + "<br>"); //Output Md Reazuddin
    document.write(family['father'] + "<br>"); //Output Md Reazuddin
</script>
```

We Can Use For.. in Loop to iterate the value of object

```
<script>
    let obj = {
        firstName: "Sarfraz",
        lastName: "Alam",
        age: 27
    }
    for (let i in obj) {
        document.write(obj[i] + "<br>");     // output sarfraz alam 27 in three l
</script>
```

//Using constructor method:- Mowbikwick id - 2709115

```
<script>
    let abc = function (a, b, c) {
    this.a = a;
    this.b = b;
    this.c = c;
    }
    let d = new abc(10, 20, 30)
    console.log(d);
</script>
```



Q9. Name the types of functions.

The types of function are:

Named - These type of functions contains name at the time of definition.

```
<script>
    function display() {
        document.write("Named F`unction");
    }
    display();
</script>
```

Anonymous - These type of functions do not contain any name. They are declared dynamically at runtime. The given one is an anonymous function as there is no name but it will give error.

```
<script>
    function () {
        document.write("Anonymous Function");
    }
```

```
</script>
```

In order to remove the error we will store this function in any variable called function expression as shown below:-

```
<script>
    Var display = function () {
        document.write("Anonymous Function");
    }
    display();
</script>
```

We use anonymous function when we have to use that function once after a some interval or we have to pass a function as a parameter.

We use anonymous function in set timeout . Here the given function will be called after 3 seconds so the JavaScript saved the function name abc() in the ram(memory) and executed after 3 seconds.

```
<script>
    function abc(){
        document.write(" I am an anonymous function");
    }

    setTimeout(abc, 3000);
</script>
```

But we need to store something when we have to use that again and again but here we are using only once so just to avoid the saving or storing process, we directly pass the function without any name called anonymous function as shown below.

```
<script>
       setTimeout(function () {
           document.write(" I am an anonymous function");
       }
           , 3000);
   </script>
```

Anonymous function with arrow function, anonymous function used in setTimoutwhich has no name

```
<script>
    setTimeout(()=>{
        document.write(" I am an anonymous function");
    }
        , 3000);
</script>
```

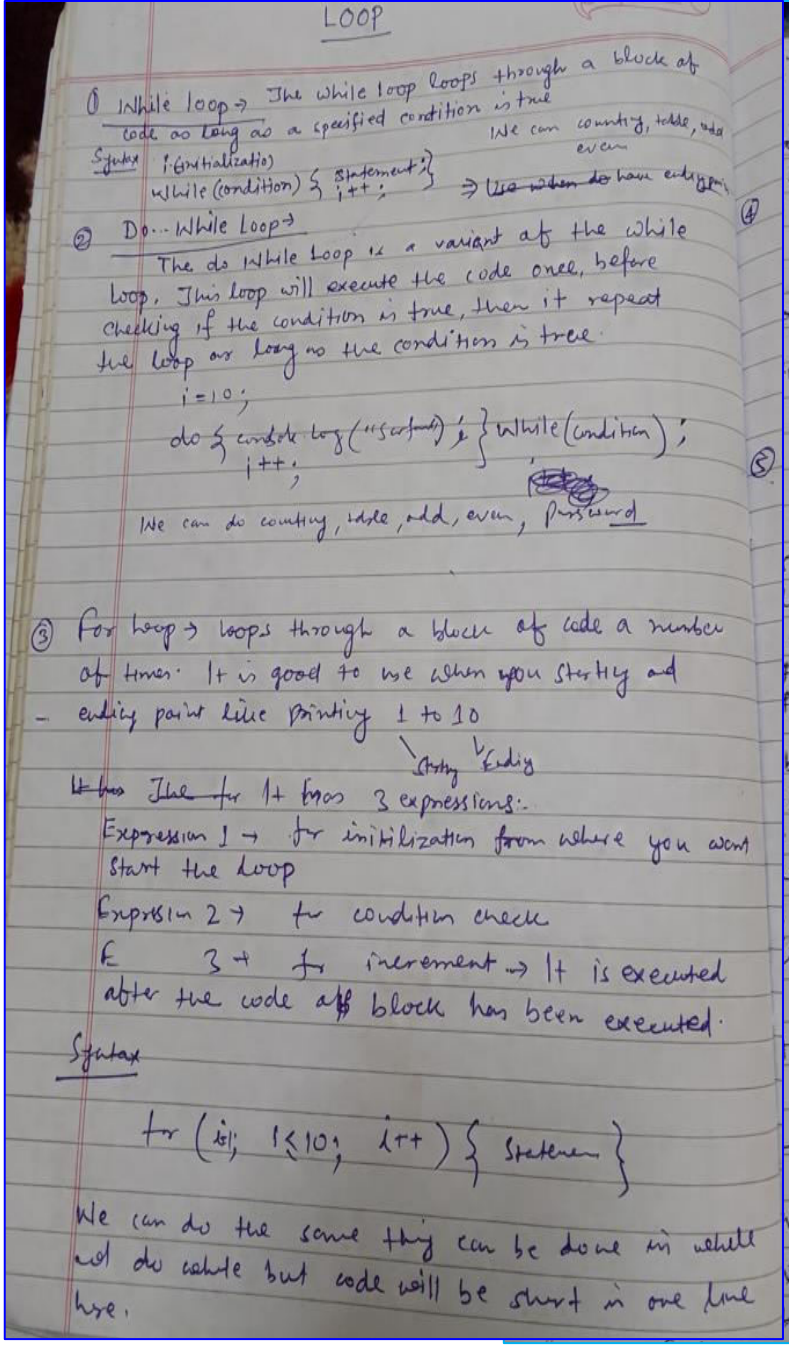
Can an anonymous function be assigned to a variable?

Yes, you can assign an anonymous function to a variables

10. LOOPS

While loop:- It loops through a block of code as long as a

specified condition is true.



forEach()

```
<script>
    let array = ["sarfraz", 52, "alam"]

    array.forEach(function(value, index){
        document.write(`${index}  :  ${value} <br>`);
    });
</script>
```

DO WHILE LOOP

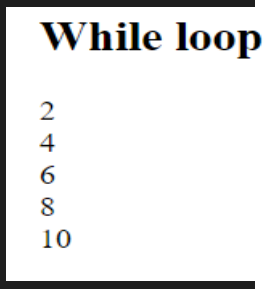
```
<h2>Do While loop</h2>

<script>
    var i = 2;
    do {
        if (i % 2 !== 0)
            document.write(i + "</br>");
        i++;
    } while (i <= 10);
</script>
```

Do While loop

3
5
7
9

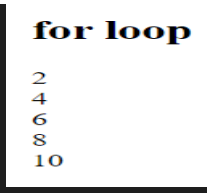
WHILE LOOP

```
<h2>While loop</h2>
   <script>
       var i = 1;
       while (i <= 10) {
           if (i % 2 == 0)

               document.write(i + "</br>");
           i++;
       }
   </script>
```



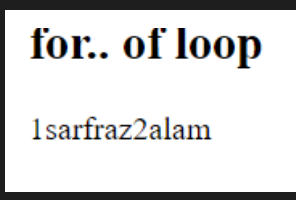
FOR LOOP

```
   <h2>for loop</h2>
   <script>
       for (var i = 1; i <= 10; i++) {
           if (i % 2 == 0)
               document.write(i + "</br>");
              }
   </script>

 <h2>for loop to iterate array</h2>
   <script>
       let array1 = [1, "sarfraz", 2, "alam"]
       for (i = 0; i < array1.length; i++) {
           document.write(array1[i] + "</br>");
       }
   </script>
```



FOR .. OF LOOP

```
   <h2>for.. of loop </h2>
   <script>
       let array = [1, "sarfraz", 2, "alam"]
       for (var value of array) {

           document.write(value);
           console.log(value);
       }
   </script>

   <h2>for.. of loop to print string </h2>
   <script>
       let str = "I am sarfraz";
        for (let char of str) {
           document.write(char);                          // IT WILL PRINT I  A M  S A F R A Z IN ROWWISE.
           console.log(char);
          }
   </script>
```



ITERATOR

```
   <h2>iterator</h2>
   <script>
       let ar1 = [1, "sarfraz", "dil", 3]
       let atrtr = ar1[Symbol.iterator];
       console.log(atrtr);     // It will return function
       let atrtr1 = ar1[Symbol.iterator]();  // Syntax of iterator
       console.log(atrtr1);  // It will return iterator
       console.log(atrtr1.next());   // It will display {value: 1, done: false}
       console.log(atrtr1.next());  // It will display value: 'sarfraz', done: false}
       console.log(atrtr1.next()); // It will display {value: 'dil', done: false}
       console.log(atrtr1.next()); // It will display   {value: 3, done: false}
       console.log(atrtr1.next());  // It will display {value: undefined, done: true}
   </script>
   <h2>iterator to print at one go </h2>
   <script>
       let ar2 = [1, "sarfraz", "dil", 3]
      let atrtr2 = ar1[Symbol.iterator]();  // Syntax of iterator
```

```
        let x = atrtr2.next();
        while (!x.done) {
            console.log(x.value);
            x = atrtr2.next();
        }
```

```
            //  for( let value of ar2){
            //     console.log(value);                  // It will return 1 sarfraz dil 3 in row
            //  }
 </script>
```

```
   <script>
      let array = [10, 20, 30];

      let x = array[Symbol.iterator];   // function
      let y = array[Symbol.iterator]();  // Object
      let k =  y.next();
      let l =  y.next();
      let m =  y.next();
      let n =  y.next();
     let z = typeof(y);
      console.log(k);  //Object { value: 10, done: false }
      console.log(l); // Object { value: 20, done: false }
      console.log(m); // Object { value: 30, done: false }
      console.log(n); // Object { value: undefined, done: true }
      console.log(k.value);  // 10
      console.log(l.value);  // 10
      console.log(m.value);  // 10
```

```
   </script>
```

GENERATOR

It is an advance version of iterator. If we take a function and assign three variables inside it and it will print all the variables as soon as the function is called. So, there is no control on it but generator allows us to control the variables to print. We need to add * before function to make generator function and as soon as we add * before function name it will not print anything when function is called and to print all the variables we need to add .next();

```
   <script>
            function *abc(){
                  console.log("1st");
                  console.log("2st");
                  console.log("3nd");
            }
            let g = abc();
                 g.next();           // 1st,2nd, 3rd will be printed all at once and to get control we will need yield, it will stop to print and to print we need to use g.next() one by one
   </script>
```

TO PRINT ONE BY ONE

```
   <script>
            function *abc(){
            yield console.log("1st");
            yield console.log("2st");
            yield console.log("3nd");

            }
            let g = abc();
            g.next();   // 1st
            g.next();   // 2nd
            g.next();   // 3rd
   </script>
```

What are generator functions?

Generator functions are declared with a special class of functions and keywords using function*. It does not execute the code, however, it returns a generator object and handles the execution.

NESTED LOOP

```
    <script>
        for (a = 1; a <= 4; a++) {
            for (b = 1; b <= 5; b++) {

                document.write("1 ");
            }
            document.write("<br>")
        }
    </script>

    <h2>Second Loop</h2>
    <script>
        for (a = 1; a <= 5; a++) {//This is for number of rows
            for (b = a; b <= 4; b++) {//This is for number of columns
                document.write("b ");

            }
            document.write("<br>")
        }
    </script>

<h2>Third Loop</h2>
    <script>
        for (i = 1; i <= 5; i++) {
            for (j = 1; j <= 4; j++) {

                document.write(j);
            }
            document.write("<br>");
        }
    </script>


<h2>Even Number between 1 to 10</h2>
    <script>
 for (i = 1; i <= 10; i++) {
            if (i % 2 == 0)
                document.write(i + "<br>");
        }
    </script>
```

1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1

Second Loop

b b b b
b b b
b b
b

Third Loop

1234
1234
1234
1234
1234

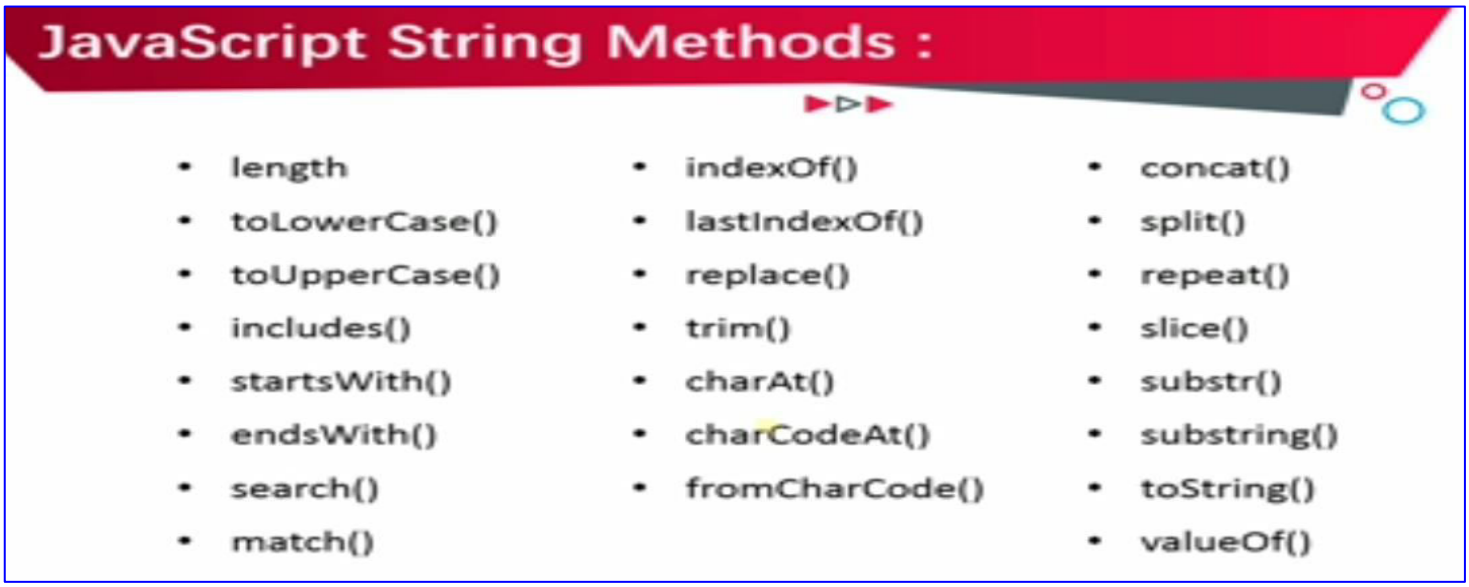
Fourth Loop

1234
234
34
4

forEach()

```
        <script>
            let array = ["sarfraz", 52, "alam"]

            array.forEach(function(value, index){
                document.write(`${index}  :  ${value} <br>`);
            });
        </script>
```

11. STRING METHOD



length property of string method, It will count the number of all index starting from one in the word.

toLowerCase() & toUpperCase() to make all the string in lower and upper case.

includes() function is used to search a word in the string and return the values "yes" if that word is available in the string or else "no". It is a case sensitive.

startsWith() and endsWith() are also used to search word. startsWith() is used to search the starting word like Sarfraz, Sar but not arf etc and endsWith is used to search the ending word like boy. is true but just boy will be false else it will check the ending word and comma too. ending word should be there then it will be true else false. like is a good boy. will be true but is a good bo will be false.

search() is just like includes() but it gives index if that word available you are finding. It will return -1 if not found in the string.

match() It is just like search that searches the word you find and return in array like is, is if 'is' available twice in the sentence(Sarfraz is a goog boy. is not he?) Here, we will write s.match(/is/g); 'is' written under forward slash to seach it and g means globally means in the entire string.

indexOf() and lastIndexOf() will return the index from beginning and lastIndexOf() will return index of the word you find from last.

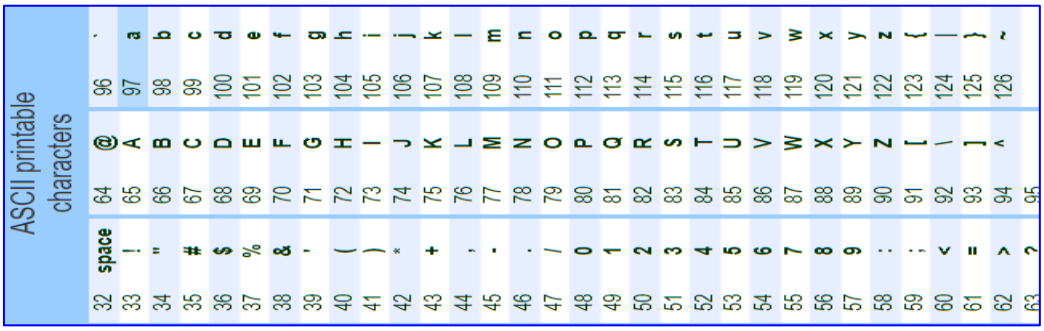
replace() It will replace the word you want and print the string with the replaced sentence. You first need to give the first word you want to replace and then second word from which you want to replace. You can also use /is/g to find as it will replace both 'is'

trim() it is used to remove extra gap if we write like " sarfraz". It is usually used in input in form to remove the extra gap/space.

charAt() It give the word/charactor available at the index you give like charAt(2). No word will be written if the index is more than the charactor lenght like charAt(300)

Note:-Sky code/charector code/Uni-code is a code for each button available on the key board.

charCodeAt(1) means returning sky code of a(if string="jarfraz") and it will be 97.
You will get this vaule when you seach the sky code of 'a' in the google.



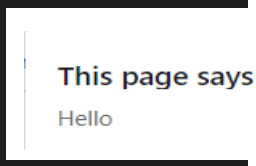
fromCharCode() It will give the code value like 'A' if you write document.write(String.fromCharCode(65)); as 65 is the code of capital'A' that can checked in the sky code in the google.

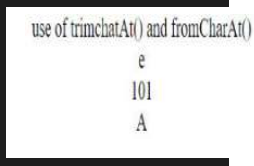
split() means remove the letter or space you select like split(" "). Here all spaces will be split or removed from the string and add comma where spaces were available and split("i") means i will be removed or split from i and coma will be added so make it array.

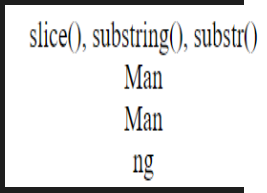
substr(3,5) in Javascript will return ascri as 3rd index is a and from a it will count 5 so ascri will be returned.

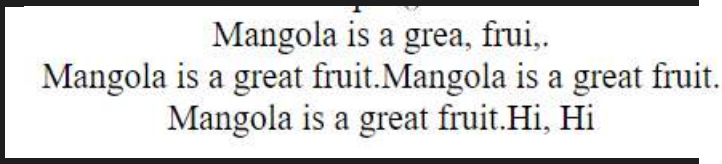
```
<script>
    var str = "Sar A";
    var a = str.length;
    document.write(a + "<br>");           // 5 It also counts space or comma or any charactor
    var str = "Sarfraz is a good boy.";
    var b = str.length;                   // 22
    document.write(b + "<br>");


    var d= "SArFRAZ"
    document.write(d.toLowerCase() + "<br>");         //sarfraz
    var e= "sarfrZ"
    document.write(e.toUpperCase() + "<br>");         //SARFRAZ
    document.write("sarfraz".toUpperCase());          //SARFRAZ


      var x = "sasrfraz is a good boy";
      document.write(x.includes("is"));       //true


    document.write("<h5>startsWith and endsWith</h5>");
    var sw = "Love you boy."
    var j= sw.endsWith("boy.");
    document.write(j+ "<br>");     //true
    document.write(sw.startsWith("l") + "<br>");    // false because it is case sensetive

    var srch = "He is a boy."
    var f = srch.search("is");
    document.write(f + "<br>");      // I will return 6 the index of a and 3 for index of is. It counts from 0



    var s = "Sarfraz is a goog boy. is not he?"
    mat = s.match(/is/g);
    document.write(mat + "<br>");     //return is is

    mat = s.indexOf('is');
    document.write(mat + "<br>");               // 8
    mat = s.lastIndexOf('is');
    document.write(mat + "<br>");               //23



    mat = s.replace('Sarfraz', 'Alam');
    document.write(mat + "<br>");               //Alam is a goog boy. is not he?
    mat = s.replace(/is/g, 'Alam');
```

```
            document.write(mat + "<br>");             //"Sarfraz Alam a goog boy. Alam not he?"

            document.write("use of trim");
            var trm= "   Hello";
            document.write(alert(trm));  // Here you can see the extra space taken in trm so when
                                    we alert it shows the extra space but trim will remove that space.
            var k = (trm.trim());

            alert(k);

            document.write("chatAt() and fromCharAt() <br>");
            var ch= "Hello";
            document.write(ch.charAt(1) + "<br>");  // It will return the charactor at 1 index
            document.write(ch.charCodeAt(1)+ "<br>"); //It will return the charactor code 101 at 1 index i.e
'e'
            document.write(String.fromCharCode(65)+ "<br>"); // It does not need any string. We need to add
            // String before 'fromCharcode' 65 is code and it will return the charactor available at 65.


            document.write("slice(), substring(), substr()<br>");
            var ch= "Mangola";
            document.write(ch.slice(0, 3) + "<br>");  // It will take the charctor from 0 index and before 3
index
            document.write(ch.substring(0, 3)+ "<br>"); //It will be work like slice but here values can be
            //swapped like (3, 0) will have same result but in slice(3,0) will not print anything
            document.write(ch.substr(2, 2)+ "<br>"); // It will count the charactor from second index for fist
            // parameter and for second one two more charctor from index 2 including 2nd index.


          document.write("split() <br>");
            var ch= "Mangola is a great fruit.";
            document.write(ch.split("t") + "<br>");  // It will replace all 't' in the string to comma or if (" ") the all
spaces to comma
            document.write(ch.repeat(2) + "<br>");  // It will repeat twice
            document.write(ch.concat("Hi") + "<br>");  // It will add


        </script>
```









C:\Users\pione\OneDrive\Desktop\FED\JAVASCRIPT\8. JS String Methods, functions

ARRAY N ITS METHODS

C:\Users\pione\OneDrive\Desktop\FED\JAVASCRIPT\6. Array

C:\Users\pione\OneDrive\Desktop\FED\JAVASCRIPT\7. Array Methods or Functions

NUMBER METHODS

What is the use of a Number object in JavaScript?

The JavaScript we use number object when we have to add, modify or do some mathematical operations with numbers.

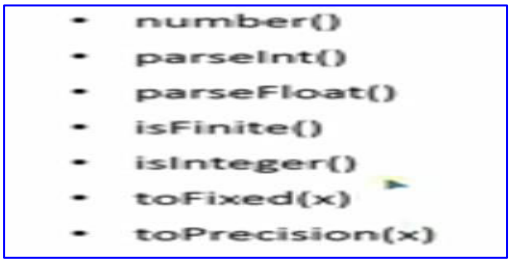
To make a number object we use constructor as

let no= new Number(25);

Or let num = 25;

```
<script>
      function display() {
         var x = 102;   //integer value
         var y = 102.7;   //floating point value
```

```
        var z = 13e4;          //exponent value, output: 130000
        var n = new Number(16);        //integer value by number object
        document.write(x + " " + y + " " + z + " " + n);
     }
     display();
  </script>
```



Number():- It will convert the string into a number. The string should be a number then like var a="25" it is a string with number. We can also add and do other mathematical operations. If the number like "25 36" is taken then it will return NaN(not a number) as it is not a number.

parseInt() & parsefloat() ParseInt will convert the decimal value into Integer(It will convert the first value without space or point like 99 56.25 you will get 99 because there is space after 99 if it is 56.23 then you willl get 56)[Integer means without point] and

parseFloat will give number whether that is 10 or 10.50(you will get 10 and 10.50) but if var a = 10 years then you willl get 10 only[float means with point].

isFinite() & isInteger() Number.isFinite is used to check if the number is finite(can be counted

or not), it will return true if it is a finite or else false. Similarly, isInteger is used to check if the value is integer or not. We will use Number.isInteger becuase it checks if the given number is integer or not. 99 will be true but "99" will be false as in integer becase "99" is a string value which is not integer. 99.50 is not integer because it has point.

toFixed(3) will fix the value after 3 digits of points and in the last value one will be added like 2.666555 to fixed(3) to 2.667 and 6.5656 to fixed(2) 6.57

Refer C:\Users\pione\OneDrive\Desktop\FED\JAVASCRIPT\9. Number Method()

```
 <script>
          //smallest and greates number javaScript can work
          document.write(Number.MAX_VALUE);  // 1.7976931348623157e+308
          document.write(Number.MIN_VALUE);  //  5e-324
  </script>
```

```
<script>
          document.write("<mark>Use of Number()</mark>" + "<br>")
    <script>
       var x = "25";
       document.write(typeof(x) + "<br>" );  // string
       var y = Number(x);
       document.write(y + "<br>");           // 25
       document.write(typeof(y) + "<br>" );   // number
    </script>

         document.write("<mark>Use of parseInt()  & parseFloat</mark>" + "<br>")
           var str = "99 65.56";
           var a = parseInt(str);
```

```
        document.write(a + "<br>");  //99
        var str = "65.56";
        var a = parseInt(str);
        document.write(a + "<br>");   // 65
        var str = "0.99";
        var a = parseFloat(str);
        document.write(a + "<br>");   // 0.99
```

```
        var str = "10 years";
        var a = parseFloat(str);
        document.write(a + "<br>");     // 10
        var str = "sarfraz 10";
        var a = parseFloat(str);
        document.write(a + "<br>");       //NaN
        var str = "10 sarfraz";
        var a = parseFloat(str);
        document.write(a + "<br>");    //10
```

```
        document.write("<mark>Use of isFinite()  & isInteger</mark>" + "<br>")
        document.write(Number.isFinite(2.10) + "<br>" );// true which can be counted
        document.write(Number.isInteger(2.10) + "<br>" );
        // false integer means positive value without point or only zero after point like 2.0
        var str = "99 65.56";
        var a = Number.isFinite(str);
        document.write(a + "<br>");           // false
        var str = 99;
        var a = Number.isFinite(str);
        document.write(a + "<br>");           //true
        var str = "65.56";
        var a = Number.isFinite(str);          //false
        document.write(a + "<br>");
        var str = "9.9";
        var a = Number.isInteger(str);         //false becuause it is a string not an integer.
        document.write(a + "<br>");
        var str = "99";
        var a = Number.isInteger(str);
        document.write(a + "<br>");           //false becuause it is a string not an integer.
        var str = 99;
        var a = Number.isInteger(str);
        document.write(a + "<br>");            // ture
        var str = 99.66;
        var a = Number.isInteger(str);
        document.write(a + "<br>");            // false

        document.write("<mark>Use of toFixed(x)  & toPrecision(x)</mark>" + "<br>")
        var str = 99.456;
        var a = str.toFixed(2);                   // 99.46
        document.write(a + "<br>");
        var str = 99.4565;
        var a = str.toFixed(3);                   //99.457
        document.write(a + "<br>");
        var str = 0.456;
        var a = str.toFixed(2);
        document.write(a + "<br>");               // 0.46
```

```
        var str = 0.456;
        var a = str.toFixed(3);
        document.write(a + "<br>");            //0.456
        var str = 0.45666;
        var a = str.toFixed(3);
        document.write(a + "<br>");           // 0.457

      document.write("<mark>Use of toPrecision(x)</mark>" + "<br>")
<script>
  var str =  56.456;
  var a = str.toPrecision(2);    // 56
  var b = str.toFixed(2);        // 56.46
```

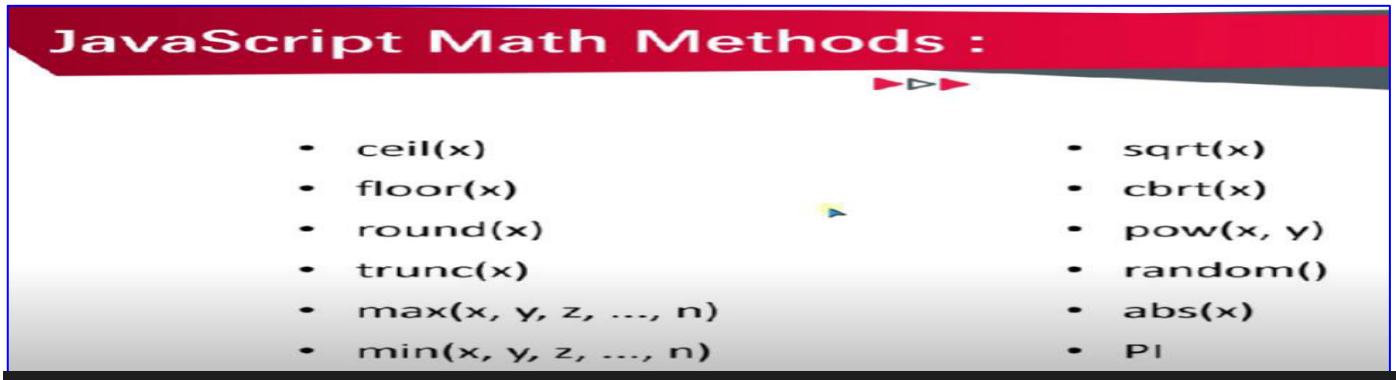
```
        document.write(a + "<br>");
        document.write(b + "<br>");
      </script>


          var str = 0.456;
          var a = str.toPrecision(2);
          document.write(a + "<br>"); //0.46
          var str = 56.456;
          var a = str.toPrecision(2);
          document.write(a + "<br>");
          var str = 56.456;
          var a = str.toPrecision(3);
          document.write(a + "<br>");    // 56.5
          document.write("round <br>")
          var r = 10.555;
          document.write(Math.round(r)+"<br>");   // return 11 as it is more than 50 after point
          var k = 10.29;
          document.write(Math.round(k));   // return 10 as it is less than 50 after point
        </script>
```

Math Methods

What is the use of Math object in JavaScript?

The JavaScript math object provides several constants and methods to perform a mathematical operation. Unlike date object, it doesn't have constructors. For example:

Math methods are usually not used in website but used in animation, video games development or complex accounting web applications. Math method is used with number data type only. There is only property pi and the rest is methods used to perform mathematical operation.




```
 <script>

    document.write(Math.ceil(5.2)+"<br>"); //It will return upper value of 5.2 which is 6
    document.write(Math.ceil(-5.2)+"<br>"); //It will return  lower value of 5.2 in minus which is -5
    document.write(Math.floor(5.2)+"<br>"); //It is just opposite to ceil & will return  lower value of 5.2 in which is 5
    document.write(Math.floor(-5.2)+"<br>"); //It will return lower value of 5.2 in minus but seems like upper because -5.2 is
greater than -6 which is -6
    document.write(Math.trunc(-5.2)+"<br>"); //It will remove the number after integer or dot -5
    document.write(Math.cbrt(125)+"<br>"); //It will return 5
    document.write(Math.sqrt(25)+"<br>"); //It will return 5
    document.write(Math.max(25,1,2,125)+"<br>"); //It will return maximum number value 125 in the following numbers
    document.write(Math.min(25,1,2,125)+"<br>"); //It will return maximum number value 1 in the following numbers
    document.write(Math.pow(2,3)+"<br>"); //It will return 2*2*2 =8 2 is base and 3 is exponential
    document.write(Math.random()+"<br>"); //It will return the random number between 0 to 1 like 0.124563 or 0.12545 etc.
    document.write((Math.random()*10) +1 +"<br>"); //It will return the random number between 1 to 10 like 0.124563 or 0.12545
etc.
    document.write(Math.round(Math.random()*10) +1 +"<br>"); //It will return the random number between 1 to 10 but that will be
in round figure.
    document.write((Math.abs(-10.280)) +"<br>"); //It will return the absolute or positive value 10.2 not negative or zero at
the end
    document.write((Math.abs(10.20)) +"<br>"); //It will return the absolute or positive value. 10.2
```

```
--    document.write(Math.PI); //It will return the PI value 3.14
    document.write("round <br>")

           var r = 10.555;
           document.write(Math.round(r)+"<br>");   // return 11 as it is more than 50 after point
           var k = 10.29;
           document.write(Math.round(k));   // return 10 as it is less than 50 after point
</script>
```

https://www.youtube.com/watch?v=zKVXQ_RgRJA

11. What are some of the built-in methods in JavaScript?

Built-in Method	Values
Date()	Returns the present date and time
concat()	Joins two strings and returns the new string
push()	Adds an item to an array
pop()	Removes and also returns the last element of an array
round()	Rounds of the value to the nearest integer and then returns it
length	Returns the length of a string

11. What are the scopes of a variable in JavaScript?

T

he scope of a variable means where the variable has been declared or defined in a JavaScript program. There are two scopes of a variable:

Global Scope

Global variables, having global scope are available everywhere in a JavaScript code. Global variable can be accessed from anywhere in the a JavaScript code even inside the curly braces(even within the a function in which it has been defined). Here is the example:-

```
<script>
       var a = "I am variable outside the function"
       function sarfraz() {
           document.write(a);
                   }
       sarfraz();   // return "I am variable outside the function"
<script>
```

Here, “var a” has been declared outside the function and can be accessed from inside the function via document.write(a);

If we write document.write(a); even outside the function the value of var a can be accessed because var a is global scope because it is outside the function.Here is the example:-

```
  <script>
      var a = "I am variable outside the function"
      function sarfraz() {

                      }
       document.write(a);
      sarfraz();    // It will return "I am variable outside the function"
   </script>
```

Local Scope

Local variables are accessible only within a function in which they are defined.

```
  <script>
      function sarfraz() {
                 var a = "I am variable inside the function and can be accessed from inside as I am local scope or local variable"
                       document.write(a);
```

```
                    }
      sarfraz();    // It will return I am variable inside the function and can be accessed from inside as I am local scope or local
variable
  </script>
```

Here, var a has been declared inside the function. So, can be accessed inside the function only because var a is local scope now.

19. Explain Scope and Scope Chain in JavaScript.

Scope in JS tells the range of variables and functions.

In general terms, the scope will define the range of variables and functions we can or cannot access.

There are three types of scopes in JS:

1. Global Scope
2. Local or Function Scope
3. Block Scope

Global Scope: In global scope, functions and variables can be accessed from anywhere inside the code.

Function Scope: In functional scope, functions and variables can be accessed inside a function, but not from outside.

Block Scope: Block scope is related to the variables declared using let and const. Variables declared with var do not have block scope. Block scope tells us that any variable declared inside a block { }, can be accessed only inside that block and cannot be accessed outside of it.

```
  <script>
     {let x = 45;}
     console.log(x);
     // Gives reference error since x cannot be accessed outside of the block
  </script>

  <script>
      for(let i=0; i<2; i++){
          // do something
             }
     console.log(i);
     // Gives reference error since i cannot be accessed outside of the for loop block
  </script>
```

Scope Chain: JavaScript engine also uses Scope to find variables. Let's understand that using an example:

```
 <script>
     var y = 24;
     function abc() {
         var x = 667;
         function abcd () {
             document.write(x + "<br>");
             document.write(y);
               }
               abcd()
             }
             abc();
  </script>
      var z = 20;
      function abc() {
          var y = 15;
          {
              let x = 10;
              document.write(x); //  block scope
              document.write(y); // local spope/ functional scoope
              document.write(z); // globe spope
          }
          document.write(z); // globe spope
      }
      document.write(z); // globe spope
      abc();
```

```
</script>
```

As you can see in the code above, if the JavaScript engine does not find the variable in local scope, it tries to check for the variable in the outer scope. If the variable does not exist in the outer scope, it tries to find the variable in the global scope.

If the variable is not found in the global space as well, a reference error is thrown.

12. What is the 'this' keyword in JavaScript?

The 'this' keyword in JavaScript refers to the currently calling object. It is commonly used in constructors to assign values to object properties.

The "this" keyword refers to different objects depending on how it is used.

In object method, this refers to the object.

Alone and in a function , it refers to the global object but in a function, in strict mode it refers to undefined.

In methods like call(), bind() and apply(), "this" keyword refers to any object.

In any event, "this" keyword refers to the event that received the event.

```
<script>
    var obj = {
        name: "vivek",
        getName: function() {
            console.log(this.name);
        }
    }
    obj.getName();     //the output will be vivek and it refers to the object 'obj'.
</script>
```

Here, the function is invoked in the context of global. So, it is a global object property.

Therefore, the output of the above code will be **the global object.** Since we ran the above code inside the browser, the global object is **the window object.**

Alone, this keyword refers to the global object because it is also invoked in the context of global.

```
<script>
    console.log(this);
</script>
```

```
▶ Window {window: Window, self: Window, document: document, name: '', location: Location, …}
```

In function, it refers to the global object

```
<script>
    function sar() {
        console.log(this);
    }
    sar();
</script>
```

```
▶ Window {window: Window, self: Window, document: document, name: '', location: Location, …}
```

Good To Remember The Output

The silly way to understand the "this" keyword is, whenever the function is invoked, we should check the object before the dot. The value of this . keyword will always be the object before the dot.

If there is no object before the dot-like in example1, the value of this keyword will be the global object.

Example 4:

```
<script>
    var obj1 = {
        address: "Mumbai,India",
        getAddress: function () {
            console.log(this.address);
        }
    }
    var getAddress = obj1.getAddress;
    var obj2 = {name: "akshay"};
    obj2.getAddress();
</script>
```

Can you guess the output?

The output will be an error.

Although in the code above, this keyword refers to the object obj2, obj2 does not have the property "address"', hence the getAddress function throws an error like not a function.

13. What are the conventions of naming a variable in JavaScript?

Following are the naming conventions for a variable in JavaScript:

- variable names can not be taken with Reserved keywords,
- variable names can not start with numeric value
- variable names can not have any space between the the charactors
- Variable names are case-sensitive.
-

For example, var, let, const, etc.

var let = "sarfra" // Wrong Way, never declare variables like this

Var var =32; //Wrong Way, never declare variables like this

- Variable names cannot begin with a numeric value. They must only begin with a letter or an underscore character.

99firstName //Wrong Way, never declare variables like this

- Leaving space between the character of variables.

firt name //Wrong Way, never declare variables like this

- Variable names are case-sensitive.

Correct Way to write variables.

Sarfarz99, first_name, first-name, firstname, firstName

13. Define callback in JavaScript?

In JavaScript, functions are objects and therefore, functions can take other functions as arguments and can also be returned by other functions.

A callback function is a function that is passed as an argument to another function, to be "called back" at a later time. A function that accepts other function as arguments is called higher-order function. callbacks are not asynchronous by nature, but can be used for asynchronous purposes. In JavaScript codes are executed synchronously which means 2nd code will be executed once first code is executed but if you want to execute the codes together like asynchronously then we will use callbacks and promise.

```
<script>
    function prmtr() {
        console.log(" I will be used as an argument")
    }
    function main(a, b, callback) {
        callback();
        console.log(a + b);
    }
    main(10, 20, prmtr);
</script>

<script>
        function de(){
            document.write("Hello")
```

```
I will be used as an argument                    valueTest.html:15
30                                               valueTest.html:19
```

```
        }
        function abc (a, b, callback){
            callback();
            document.write(`, I scored ${a+b} marks.` );
        }
        abc(100, 40, de);    // Hello, I scored 140 marks.
</script>
```

First of all we will make a function, name can be anything and passed parameter a, b, callback then a and b have been added in console and third parameter has been called.

We know that in callback function, another function is taken as an argument so prmtr(which is function has been defined above) has been taken.

Here, callback() will control the outer function prmtr().

Since callback() was called first and “I will be used an an argument” printed first and then console.log(a+b). So, result will be “I will be used an an argument” 30.

Main is called high order function and prmtr is called callback function

There are two types of callback functions. asynchronous and synchronous

Q14. Promises in JavaScript

Promises are used to handle asynchronous operations in JavaScript.

Before promises, callbacks were used to handle asynchronous operations. But due to the limited functionality of callbacks, using multiple callbacks to handle asynchronous code can lead to unmanageable code or call back hell. So, Promise was introduced in ES6 (ECMAScript 2015).

Promise object has four states -

- Pending - Initial state of promise. This state represents that the promise has neither been fulfilled nor been rejected, it is in the pending state.
- Fulfilled - This state represents that the promise has been fulfilled, meaning the async operation is completed.
- Rejected - This state represents that the promise has been rejected for some reason, meaning the async operation has failed.
- Settled - This state represents that the promise has been either rejected or fulfilled.
- A promise is created using the Promise constructor which takes in a callback function with two parameters, resolve and reject respectively.
- resolve is a function that will be called when the async operation has been successfully completed.

  reject is a function that will be called, when the async operation fails or if some error occurs.

```
<script>
    // step 1
    var pomise = new Promise(); // Promise constructor used.

    // step 2
    var promise = new Promise(function (resolve, reject) { });   // callback function taken with two parameter

    // step 3
    var promise = new Promise(function (resolve, reject) { if (condition) { resolve(); } else { reject(); } });

    // step 4
    var promise = new Promise(function (resolve, reject) { if (condition) { resolve(); } else { reject(); } });

    promise.then(OnFulfilment);    // then call back function will be called when condition is fulfilled
    promise.catch(OnRejection);     // catch call back function will be called when condition is fulfilled



    let OnFulfilment = function (result) { console.log(result) }
                                or
    let OnFulfilment = (result) => { console.log(result) }  // arrow function used
    let OnRejection = (reject) => { console.log(reject) }   // arrow function used

</script>
```

```
<h2>EXAMPLE 1</h2>
    <script>
        let a = 10;
        let prom = new Promise(function (resolve, reject) {
            if (a != 10) {
                resolve("Promise kept");
            } else {
                reject("Promise not kept");
            }
        });
        console.log(prom); // Just to test to see the result
        // var truee = (result)=> console.log(result);
        // var falses = (error)=> console.log(error);

        prom.then(function abc(result) {
            console.log(result);
        });
        prom.catch(function def(error) {
            console.log(error);
        });
    <h2>EXAMPLE 2</h2>

        function pro(a) {
                return new Promise(function (resolve, reject) {
                console.log("Promise is being initialised");
                setTimeout(() => {
                    if (a = 10) { resolve("Promise kept"); }
                    else { reject("Promise not kept"); }
                }, 3000);
            }); }
        pro(10).then((result) => { console.log(result) }).catch((error) => { console.log(error) });
    </script>
```

Using Normal function

```
  <script>
     let a=10;;
     let prom = new Promise(function (resolve, reject) {
        if (a ==10) { resolve("Promise was kept/Async was executed"); } else { reject("Promise was not kept/Async was not
               executed"); }
     })
     prom.then((result)=>{console.log(result)}).catch((error)=>{console.log(error)}) // Promise was kept/Async was executed
  </script>
```

Using Call back function

```
  <script>
     function prom(a) {
        return new Promise(function (resolve, reject) {
```

```
            if (a == 10) { resolve("Promise was kept/Async was executed"); } else { reject("Promise was not kept/Async was not
                executed"); }
        })
    }
    prom(10).then((result) => { console.log(result) }).catch((error) => { console.log(error) }) // Promise was kept/Async was
executed
  </script>
```

Q15. PROMISE.ALL

If all the promise is true then then call back function will execute and if any of the promises is false then catch call back will execute

```
  <script>
    let p1 = new Promise(function (resolve, reject) {
      setTimeout(() => {
        console.log("My first promise was kept")
        resolve(10)
      }, 1* 1000)
    })

    let p2 = new Promise(function (resolve, reject) {
      setTimeout(() => {
        console.log("My Second promise was kept")
        resolve(20)
      }, 2* 1000)
    })
    let p3 = new Promise(function (resolve, reject) {
      setTimeout(() => {
        console.log("My third promise was kept")
        resolve(30)
      }, 3* 1000)
    })
    Promise.all([p1, p2, p3]).then((result) => { console.log(result) }).catch((error) => { console.log(error) });
  </script>
```

```
My first promise was kept                                   test.html:19:21
My Second promise was kept                                  test.html:26:21
My third promise was kept                                   test.html:34:21
▼ Array(3) [ 10, 20, 30 ]                                   test.html:42:60
    0: 10
    1: 20
    2: 30
    length: 3
  ▶ <prototype>: Array []
```

Q16. Async & Await

Async allows us to write promise based codes and a word “Async” before a function means one simple thing that function will always returns a promise and it is an Async function and work in asynchronous mode in the background.

The keyword “await” before a function makes function await for a promise. It can be used only with Async function.

Async is the latest version of Promise which means we can do all the things that we were doing in Promise and the best part of Async is that we can use one more feature with Async called await. Await is used inside the Async function. Async was introduced in ES8 in 2017 in which we don’t require to declare resolve and reject every time . Now, we can simply make a function and add Async just before the function declaration and it will return a promise. So, here we will use then and catch methods without defining resolve and reject which will reduce the code length and also complexity that we had when using in promises.

```
  <script>
        async  function klm(){
            return "Listen"
          }

          console.log(klm()); // Promise { <state>: "fulfilled", <value>: "Listen" }
  </script>
  <script>
              async function abc(a){
                 if(!a==10){return "Promise fulfilled"} else{return "Promise fulfilled"};
              }

              abc(10).then((result)=>{console.log(result)}).catch((error)=>{console.log(error)})  // Promise fulfilled
```

```
</script>
```

Async & Await Example

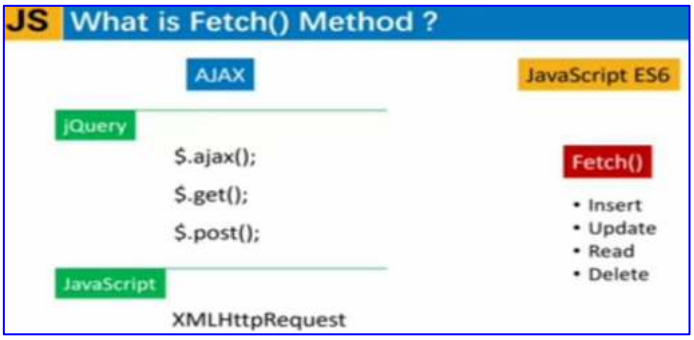
```
<script>
            async function abc(a){
                                console.log("Fist Promise");
                          await console.log("Second Promise");
                                console.log("Third Promise"); // It will be printed after Fourth
            }
            abc();
            console.log("Fourth Promise");
</script>
```

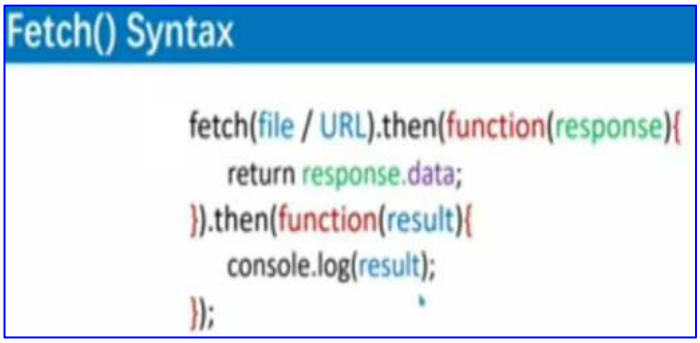
Fetching Data using Async & Await

```
 <script>
     async function test() {
           return (await fetch("content/readme.txt")).text();   // I love you so much and I love you too will be printed
         // return (await fetch("content/studentData.json")).json();
         // return (await fetch("https://jsonplaceholder.typicode.com/users/1/posts")).json();
     }
     test().then((result) => { console.log(result) }).catch((error) => { console.log(error) })
 </script>
```

Q17. FETCH()

Fetch() method was introduced in ECMAScript 2015(ES6). It is the advance version of AJAX which allows to crude with server such as reading and fetching the data from server, updating, inserting and deleting the data from server. We can do the same with AJAX in jQuery and JavaScript . When we use AJAX in jQuery using $.ajax(); , $.get(); , $.post(); methods . It is easy to use due to less coding but we need to add to 50 to 100 KB file to use it as all the codes are inside this file and without it jQuery will not work, which loads every time when we run the jQuery codes which makes the application slow and when we use AJAX in JavaScript using XMLHTTPReuest method, we have to write too many codes. So, to overcome these issues fetch() method was introduced.





Fetch() Syntax:-
1. Fetch();
2. Fetch(file path or URL); file or URL can be text, ajax, php. Fetch returns promise so we will use then() call back function considering the promise is fulfilled.
3. Fetch(file/URL).then(); When we use then() call back function, we usually need to take a function to do the next work but here we will not do so as then() returns promise so we will take then()
4. Fetch(file/URL).then().then()
5. Feth(file/URL).then(function(response){return r esponse.data;}).then(function(result){console.log(result);})

```
<div id="demo"></div>
<body>
   <script>
      fetch("data.txt").then(function (response) {
         return (response.text());
      }).then(function (result) {
         console.log(result);
         document.getElementById("demo").innerHTML = result;    // Oupt put Hello, Sarfraz
      });
   </script>
```

```
</body>
```

Shortcut Way Using Arrow Function

```
<script>
   fetch("content/readme.txt").then((response) => {
      return (response.text());
   }).then((result) => {
      console.log(result);
      document.getElementById("demo").innerHTML = result;
   });
</script>
```

Printing Local PC Data

```
<script>
    fetch("jsonData.json").then(function (response) {
        return (response.json());
    }).then(function (result) {
        console.log(result);
        document.getElementById("demo").innerHTML = result;  // It will show the data in console but not in document
        for (var i in result) {
          //  document.write(result[i].name + " : " + result[i].class +  "<br>")
            document.write(`${result[i].name} - ${result[i].class} <br>`);  //     Sarfraz Alam - BCA Pammi -B.Com
        }
    });
</script>
```

Printing Online Data from fake JSON(Here, we will get the catch error if response is not received, we can get 'can't fetch the data' if we misspell 'response' to 'response.

```
<script>
    fetch("https://jsonplaceholder.typicode.com/users/1/posts").then(function (response) {
        return (response.json());
    }).then(function (result) {
        console.log(result);
        document.getElementById("demo").innerHTML = result;  // It will show the data in console but not in document
        for (var i in result) {
          //  document.write(result[i].name + " : " + result[i].class +  "<br>")
            document.write(`${result[i].userId} - ${result[i].title} <br>`);
        }
    }).catch((error)=>{document.write("can't fetch the data")});;
</script>
```

Q17. INSERT, UPDATE, DELETE AND READ DATA FROM THE SERVER USING FETCH METHOD

To do the above operation we first write fetch(first parameter will be file/URL path, {second parameter will properties and its values to insert, update, remove data from server under curly braces})

We use different properties and its values inside curly braces like an object.

First Property is method used to send(in the server) and retrieve(from the server) the data in json.

We have different methods such as

POST to insert the data

PUT to update the server data,

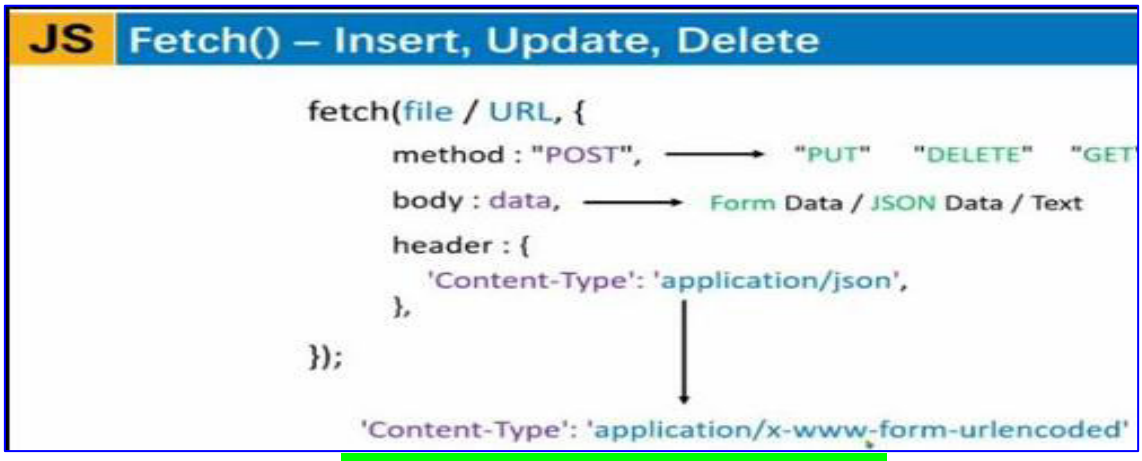
DELETE to delete the server data and

GET to read the server data and it is a default value. We can read the data even if we don't use it as it is default.

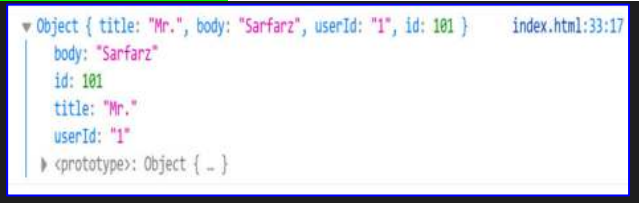
Second Property is body in which we pass the data that we want to save in the server . Data can be of different types such form data, json data or text data.

Third Property is header which refers to the formate of the data written under content type:- there are two conent type data . First is application /json refers to the json data and Second is application/x-www-form-urlencoded refers to the form data. We send these formats of data to the server through body.

NOTE:- We convert the object or data into json formate because this formate is universal formate and can be read and supported by all programming language. The data we take from users is JavaScript object and an object can not be sent to the server directly. So, we convert it into json using JavaScript function called JSON.stringify



INSERTING THE DATA TO THE SERVER USING POST




```
<script>
    fetch('https://jsonplaceholder.typicode.com/users/1/posts', {
        method: 'POST',               // To insert the data in the server
        body: JSON.stringify({
            title: 'Mr.',
            body: 'Sarfarz',
            userId: 10,          // We have a total of 9 entries of userId in the given URL, So, the data we updated will be
have userId: 10
        }),
        headers: {
            'content-type': 'application/json; charset= UTF-8',          // Since it is json formate so application/json used
        },
    }).then(function (response) { return (response.json()); }).then(function (result) {
        console.log(result);
        }).catch(function (error) { document.write("can't fetch the data") });
</script>
```

TO UPDATE

```
<script>
    fetch('https://jsonplaceholder.typicode.com/users/1/posts/1', {
        method: 'PUT', // To update the data in the server, here in the URL. we have added 1 like posts/1 so the the update at
user id:1
        body: JSON.stringify({
            title: 'Mrs.',
            body: 'Pammi Tarannum',
            userId: 10,          // We have a total of 9 entries of userId in the given URL, So, the data we updated will be
have userId: 10
        }),
        headers: {
            'content-type': 'application/json; charset= UTF-8',          // Since it is json formate so application/json used
        },
    }).then(function (response) { return (response.json()); }).then(function (result) {
        console.log(result);

    }).catch(function (error) { document.write("can't fetch the data") });
</script>
```

TO DEETE 2ND FROM 100 POST

```
fetch('https://jsonplaceholder.typicode.com/posts/2',{
    method:'DELETE',
});
```

TO READ

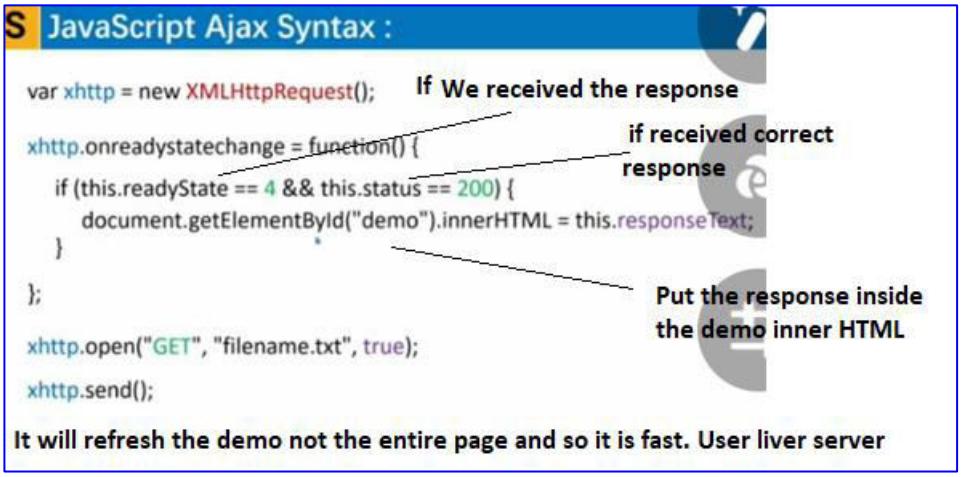
```
<script>
    fetch('https://jsonplaceholder.typicode.com/posts/1')
        .then((response) => response.json())
        .then((json) => console.log(json));

</script>
```

Define AJAX

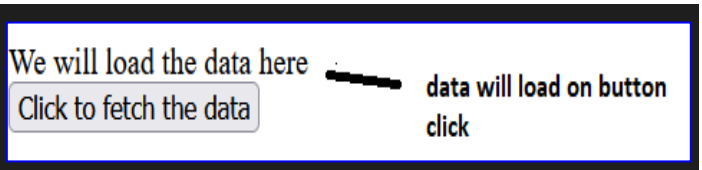
Ajax is a technique for creating fast and dynamic web page. It stands for **Asynchronous JavaScript And XML.**
XML is a format of data and Asynchronous means doing multiple work together or don't wait for another work until first work is done. It only loads the object you make changes rather than updating the entire objects of the DOM and this is why it is fast and loads quickly.

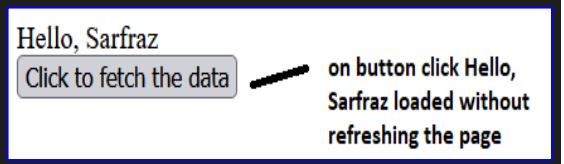
How AJAX Work

When a user requests for something to the server it goes to the server and server gives response to the user in return but in AJAX when we send any request to the server it does not go to the server directly, it goes behind the server and from there a JavaScript class called
XMLHttpRequest takes the user request and collect the response from the server and returns to the user and this process takes into five steps called readyState .**0 means neutral, we don't send any request, 1 means connection has been established between the user PC and the server, 2 means request has been received by the server, 3 means request is being processed and 4 means responses returned to the user**.
The response or content we receive from the server can be of three formats such as **text file, XML data or JSON data.**
When we receive response from the server, we get status which refers the response we have got from the server is correct or not or whether it is the same response of my request or not. If we get status code : 200 then it fine and it means we have got the response we requested for
403 means server is not responding or something is wrong, 404 means request file is not available or page not found.
We need to make object construtor for the syntax.
Let xhttp = new XMLHttpRequest();
Here, we have two methods open and send, open is used for accessing the data or content from the server, it has three parameters
1. Method:- like "GET" or "POST"
2. File Name:- filename.txt,
3. Async mode:- true means allow the Asynchronus mode, keep it true not false
and second method is used to send the request to the server.
We also use onReadyStateChange Method to check the status of the server and once received set that into HTML




```
<div id="demo">We will load the data here</div>
<button onclick="loadData()">Click to fetch the data</button>
<script>
    function loadData() {
        var ajax = new XMLHttpRequest();
        ajax.onreadystatechange = function () {
            if (this.readyState == 4 && this.status == 200) {
                console.log(this.responseText);
                //alert(this.responseText);
                document.getElementById("demo").innerHTML = this.responseText;
            }
        };
        ajax.open('GET', "data.txt", true);
        ajax.send();
    };
</script>
```





14. Define debug a JavaScript code?

Or

Requirement of debugging in JavaScript Or

Use of debug keyword

JavaScript don't show any error message in a browser. However, these mistakes can affect the output. The best practice to find out the error is to debug the code. All modern web browsers like Chrome, Firefox, etc. have an inbuilt debugger that can be accessed anytime by pressing the relevant key, usually the F12 key.

To perform debugging, we can use any of the following approaches:

Using console.log() method or using F12

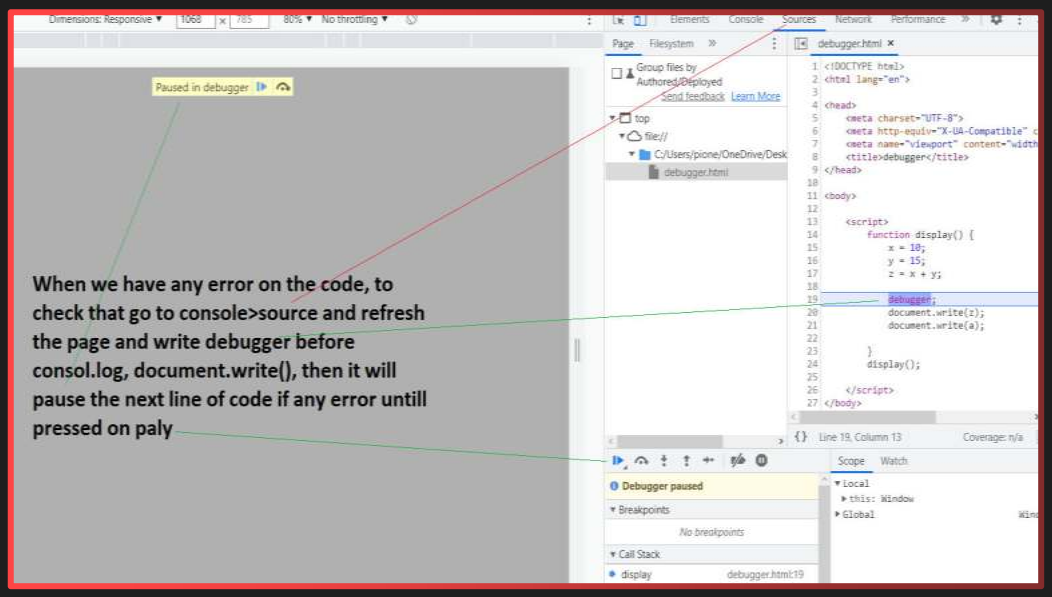
Using debugger keyword

Installing debugger tool in the editor tools like VS by going to the extension find debugger for chrome,Atom, Sublime Text, etc. etc.

Way to use debugger keyword in the JavaScript Code Using debugger Keyword

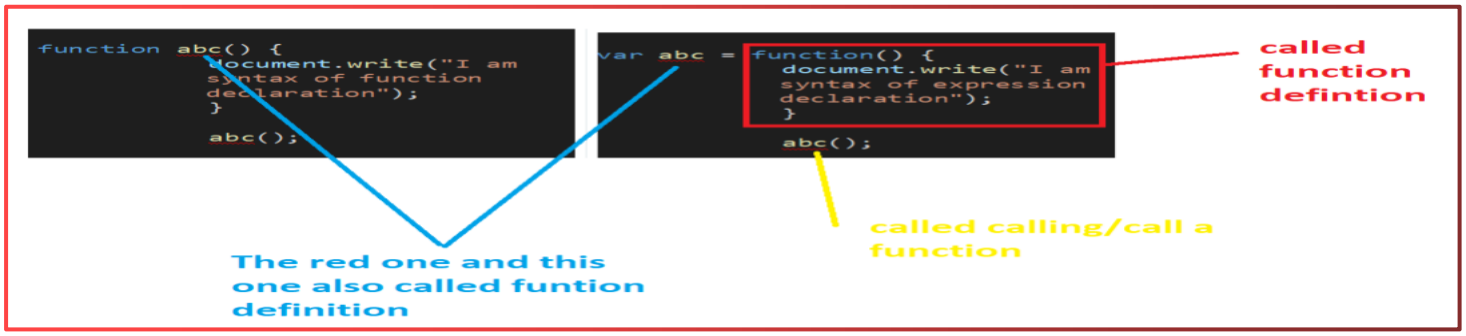
The debugger stops the execution if any error occurs and we need to start the flow of execution manually.

```
<script>


  function display() {

      x = 10;
      y = 15;

      z = x + y;

      debugger;

      document.write(z);

      document.write(a);
  }


  display();


</script>
```



15. Define function declaration and function expression (Difference)

Please understand the term before understanding the difference



Syntax:	Syntax:
`<script> function abc() { document.write("I am syntax of function declaration"); } abc(); </script>`	`<script> var abc = function() { document.write("I am syntax of function expression"); } abc(); </script>`
function definition (abc) begins with the function keyword.	**Here, abc is called function definition. So, in function expression begins with the variable.**
Function declaration is hoisted because when we call the function abc() before the function definition in function declaration, it will print the value and this is why function can be called before function definition in function declaration. `<script> abc(); function abc() { document.write("I am syntax of function declaration"); } </script>` Output :- I am syntax of function declaration **Offers better code readability and better code organization**	**Function expression is not hoisted because when we call the function abc() before the function definition in function expression, it will not print the value. It will give typeError that abc() is not a function. This is why function can't be called before function definition in function expression.** `<script> abc(); var abc = function() { document.write("I am syntax of function declaration"); } </script>` **Used when there is a need for a conditional declaration of a function**

16. What are the ways of adding JavaScript code in an HTML file?

There are primarily two ways of embedding JavaScript code:

1. Inpage JavaScript:- Using .js in the HTML page.

We can use <script> </script> before the end of the </head> tag or before the end of the </body> tag. It is good to use before the end of the </body> tag between opening and closing tag of script like

<script> JavaScipt Code </script>

So that web-side loads quickly. This is suitable when we need just a few lines of scripting within a web page

1. External JavaScript:-

We can make external file where can write all the JavaScript codes and link that file to the HTML page inside the head tag before the end of the head tag. Like

<head> <script src="my.js"> </script> <head>

Make sure, we save the my.js with js extenstion like my.js then save and also on the same file else need to change the directory

For multiple files:-

<head> <script src="my.js"> </script> <head> and <head> <script src="script.js"> </script> <head>

17 . Define web storage API OR Define html

HTML API (v)/STORAGE API

It is an API to store the web-side data. We have an API of HTML5 that we use to store the data of the web.

It has two types LOCAL STORATE and SESSION STORAGE.

Note:- It will work on live server only and use chrome for this.

LOCAL STORAGE VS COOKES

LOCAL STORATE:- HTML5 local storage API is used to store the data locally in user's browsers or user's computer/laptop. It is an alternate and better way than cookies. It is not related to the server.

When we make any website using JavaScript or JavaScript framework or library, the execution of the code takes place in user's browser or on client side(browser) and if we store the data locally means in the user's browsers or pc, the codes will execute quickly so the web page will load quickly.

But when we use any server side script we should use cookies.

Local storage data never save the data in the server with HTTP request means whenever we send data from local storage to server using HTTP request then that data will not save in the server. So, the response will be faster but cookies data will save in the server when we send data via HTTP request so it can be a bit slow in the response from the server.

We can save 5GB data in the local storage and

Local storage data never expires but we can clear data using JavaScript methods but cookies data will expire with the time you set and if you don't set any time it will expire with session.

Limitation of local storage:-

We can store data as string only, not boolean, number, etc. up to 5MB

Point to Remember

1. Local storage is a property of window object.
2. It stores data in the machine or web browser.
3. It does not have expiry date
4. It does not get saved to the server but in cookies it gets sent.

Methods:- (Both local & Session storage have the same methods)

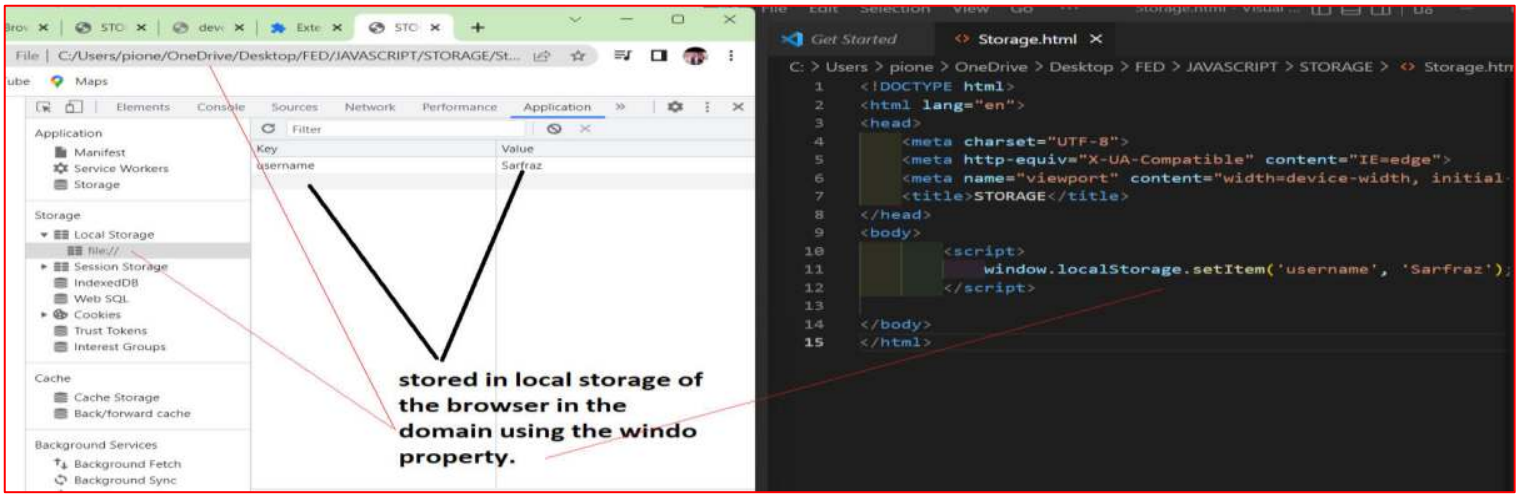
setItem(key, value) - To store or add the key value pair. It will overirde the old value if already added/saved.

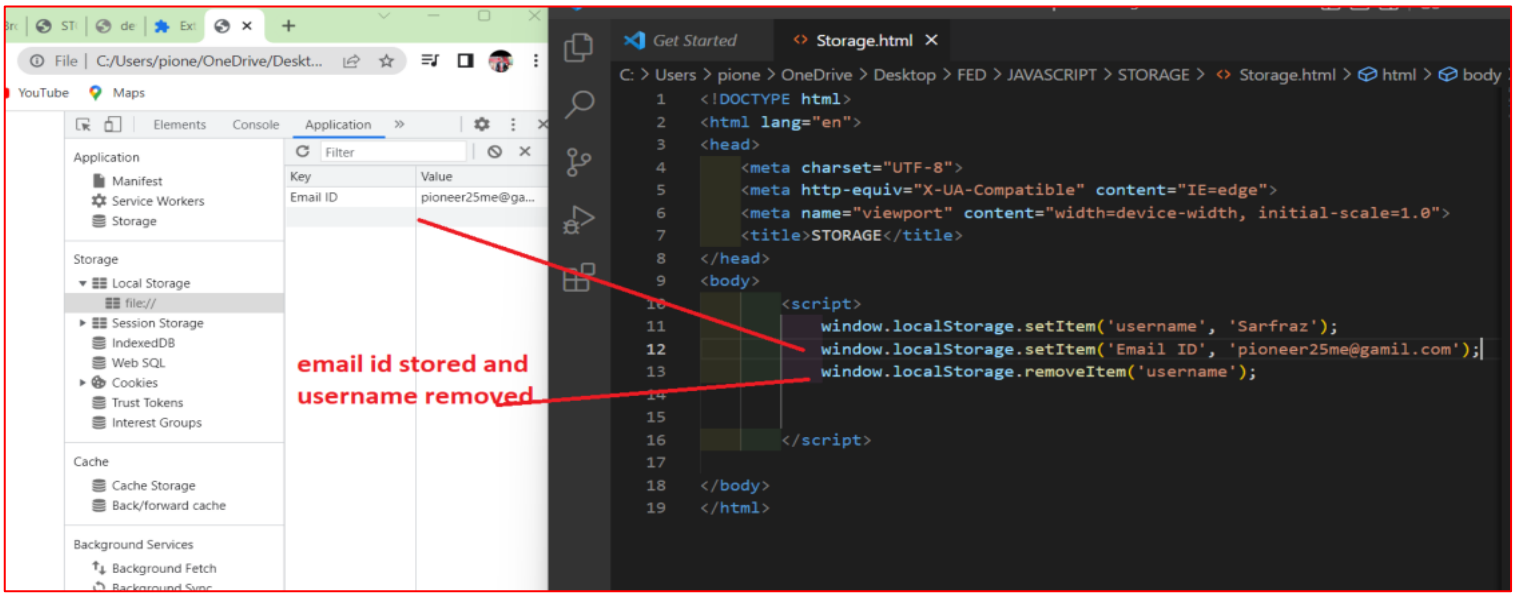
getItem(key) - To get the value aleady set or stored.

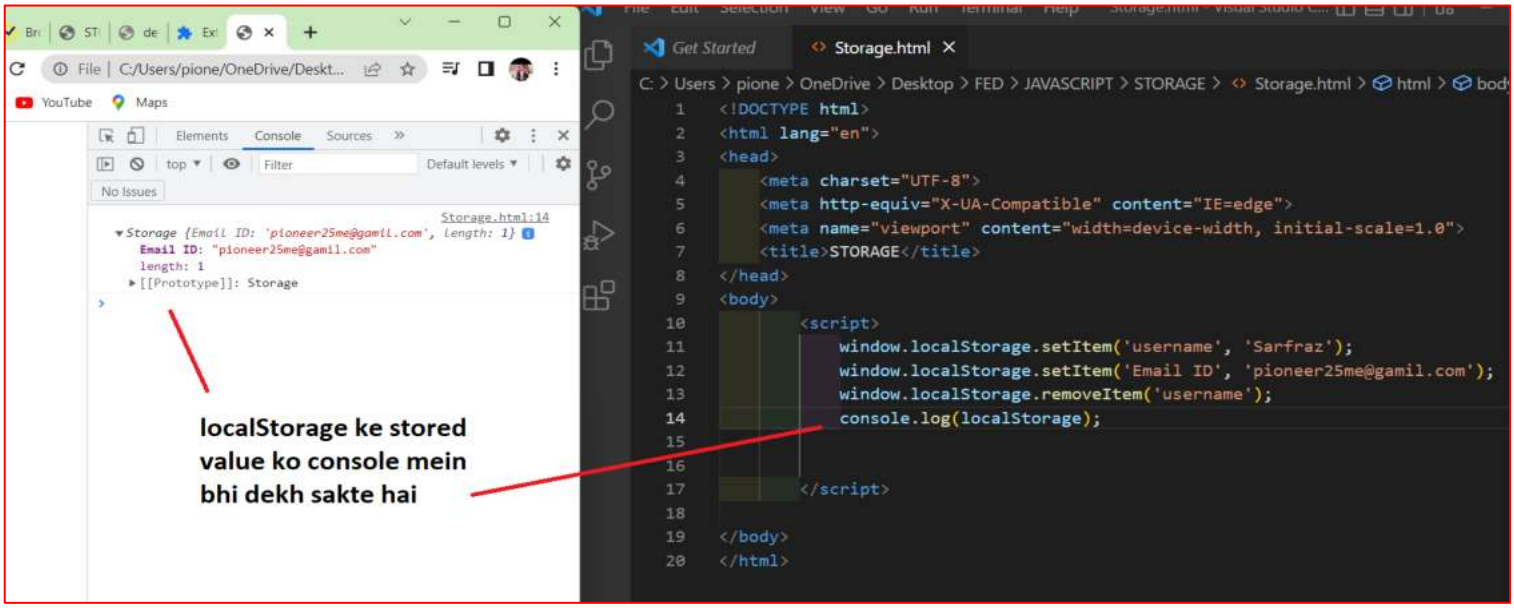
key(n) - It return the index of the value.

removeItem(key)- To remove the value stored

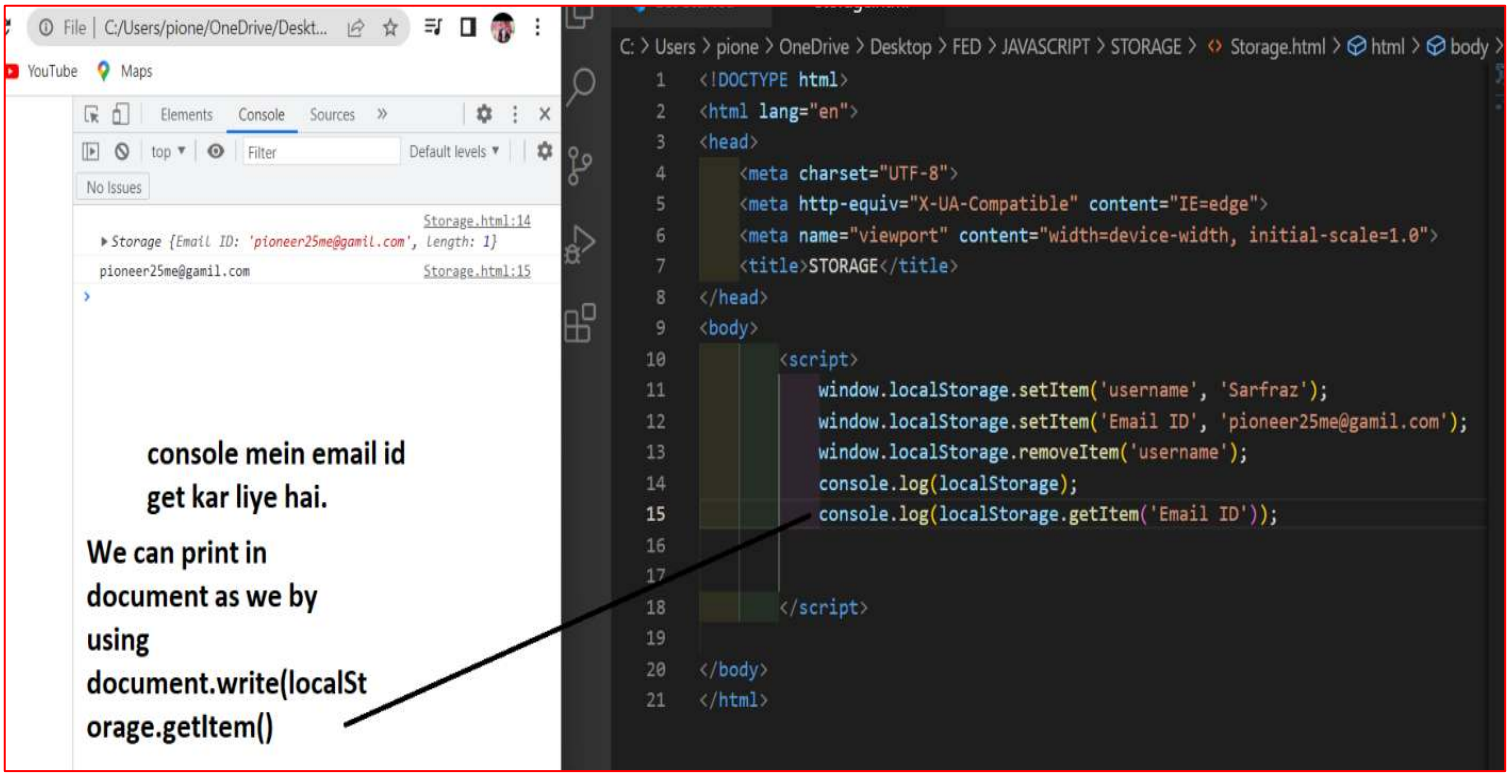
Here is the screen-shot and steps to use the above methods.

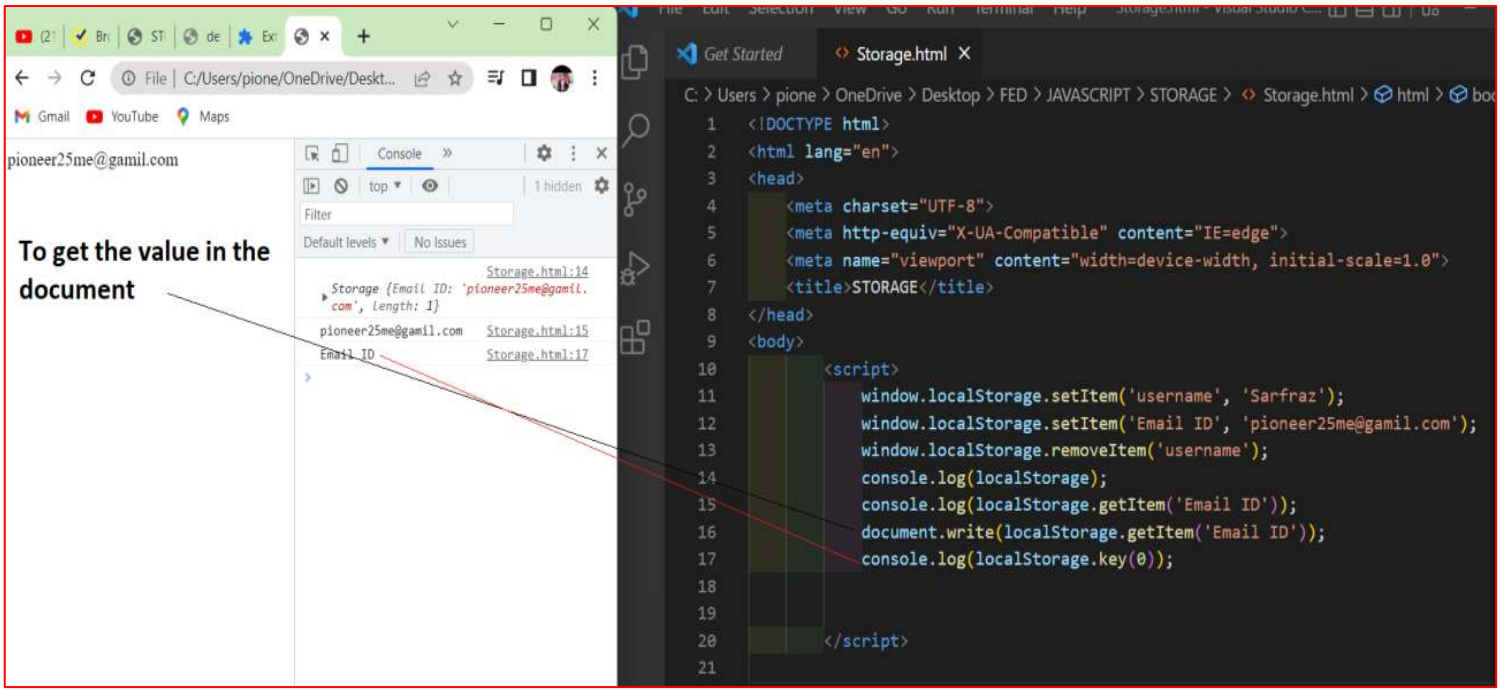






We also have window.localStorage.clear() to remove all the store items





Define cookies?

You might have noticed whenever you visit any website, an window pops up saying accept cookies. Suppose I have visited any e-commerce website in which I have not made any account and tried to purchase something and added to the cart but didn't make the payment and close the browser or shut down the system. Since I have not made any account. So, my details like user ID & password will not save in the database but add to cart information on your browser will be saved via cookies and this is why when you visit the website again, the cart information is available there.

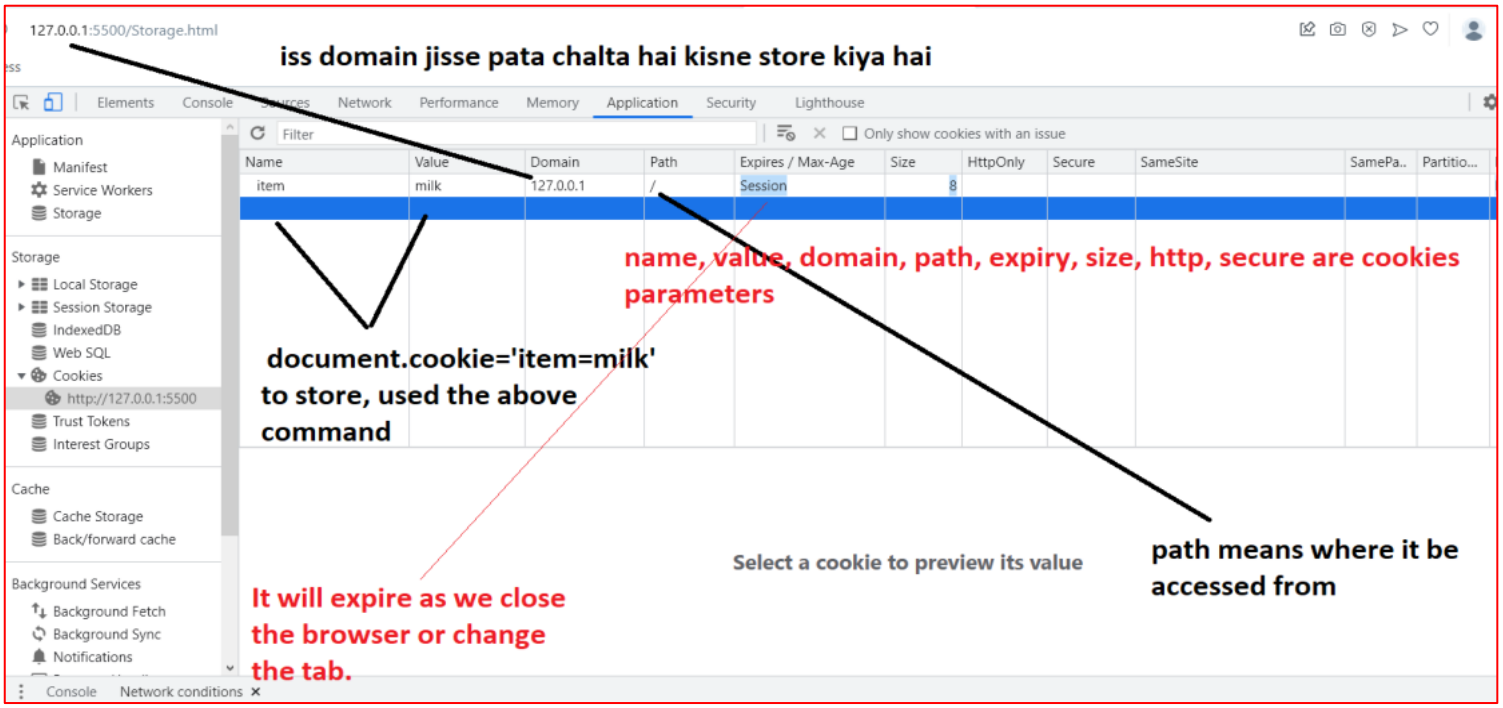
Any website that stores your data on the browser then that data is called cookies.

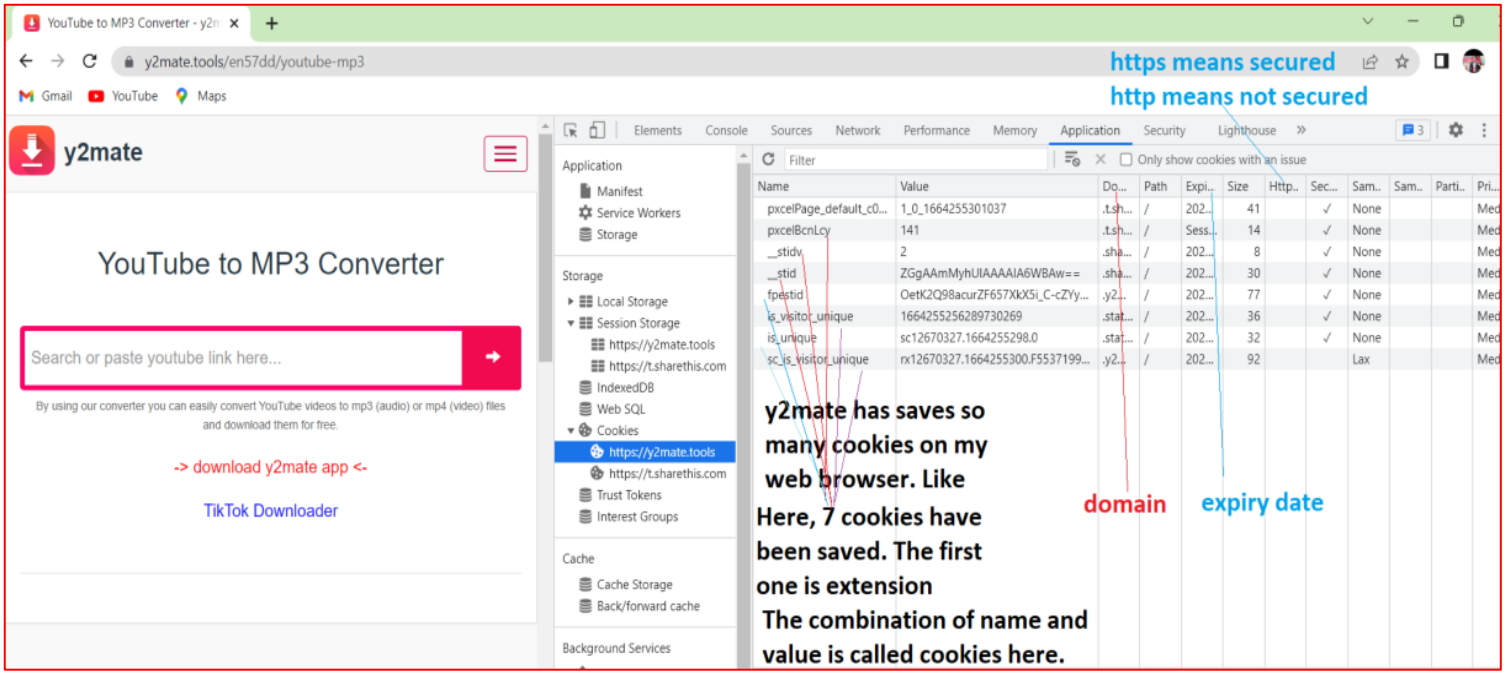
When we write any code in VS code and see the result by clicking on the page, we can see the output of the code but we will not be able to store, read and delete cookies by clicking on the browser. We need a local server and it can be installed in the VS code called live server that we show the output itself as soon as you save the code after writing.

We need to enable the cookies in the browser in case it is not enabled. We will not be able to save or see the cookies if not enabled.

We need to go to settings>privacy & security >cookies and other data sites>allow all cookies to store the cookies in local machine.

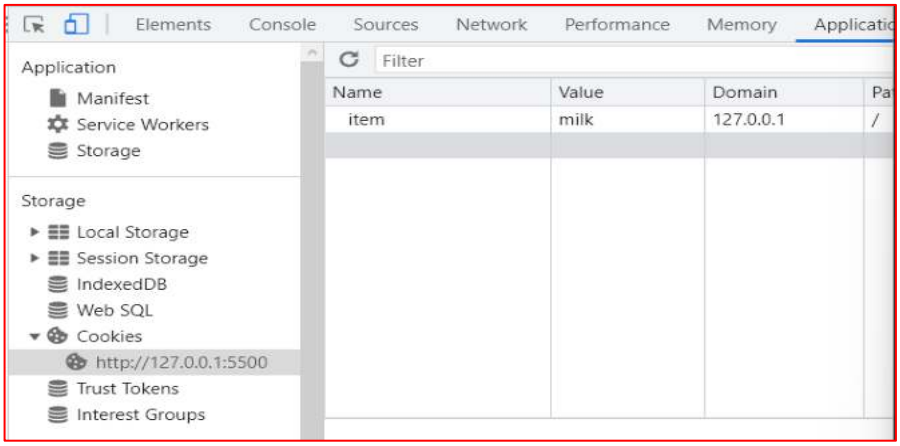
name, value, domain, path, expiry, size, http, secure are cookies parameters as you can see in the image.





In another word, a cookie is generally a small data that is sent from a website and stored on the user's machine by a web browser that was used to access the website. Cookies are used to track browser's activity done by the user.

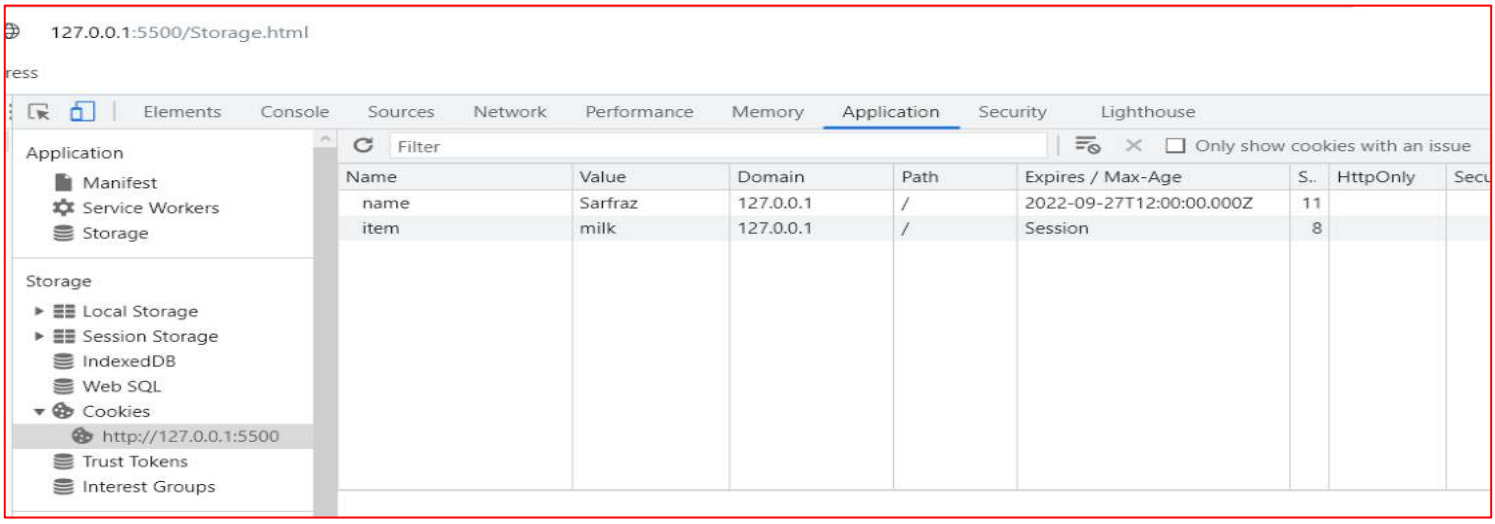
```
</script>

<h2>cookies</h2>
<script>
    document.cookie='item=milk'

</script>
```



Cookies are stored just below the local and session storage as you see in the image above.

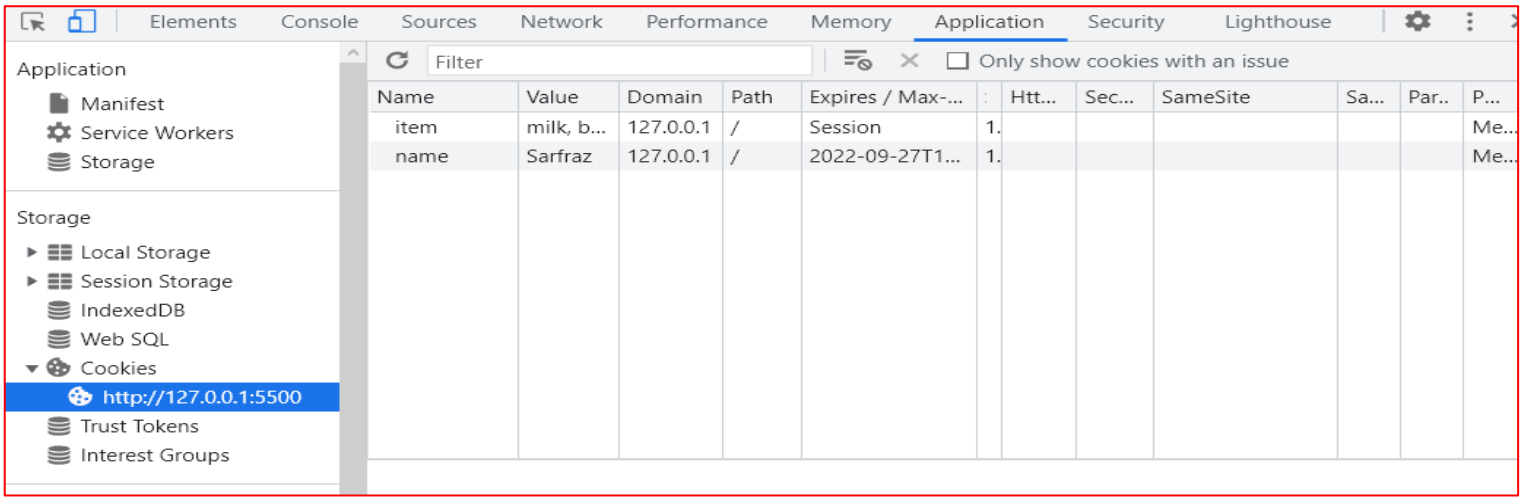
WAY TO CREATE COOKIE AND EXPIRY

```
<script>
    document.cookie="name=Sarfraz; expires=Tue, 27 Sep 2022 12:00:00 UTC";
    document.cookie='item=milk';
```



TO CHANGE OR EDIT THE COOKE

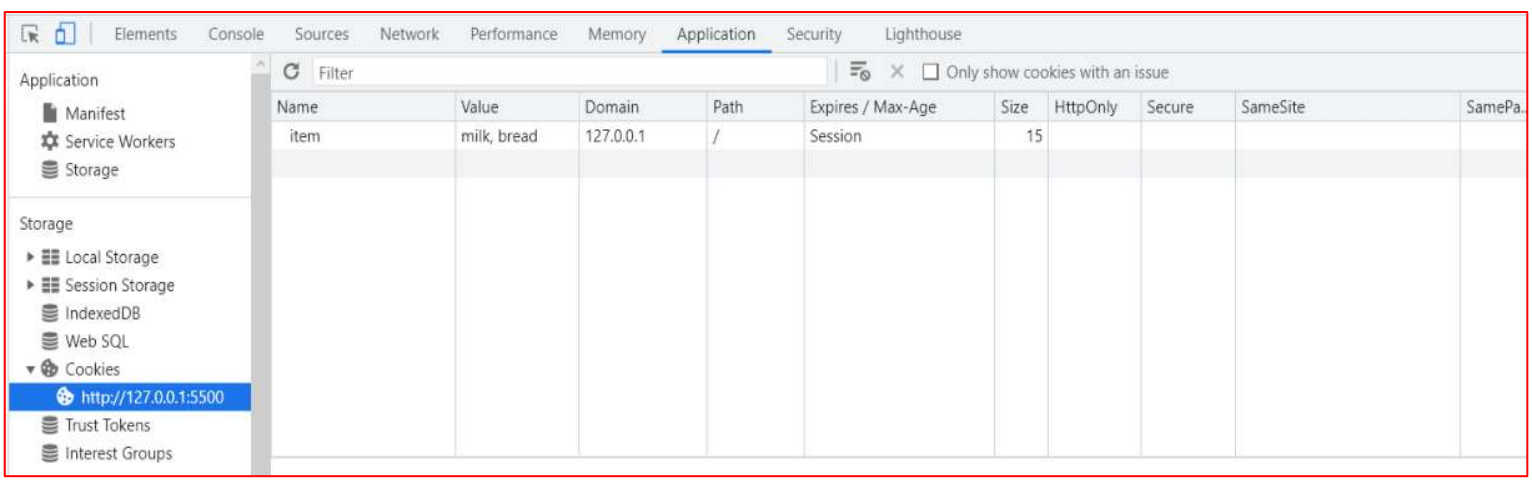
```
document.cookie='item=milk, bread';    // To change or edit
```



ITEM HAS BEEN CHANGED FROM MILK TO MILK AND BREAD

TO DELETE THE COOKE

```
document.cookie="name=Sarfraz; expires=Tue, 26 Sep 2022 12:00:00 UTC";
// We need to give past date to delete the cookie. It 27/09/2022 today and I have set 26/09/2022 So, it will delete now
```



You can see name= sarfraz has been deleted.

TO READ OR PRINT THE COOKE

```
let x =document.cookie;
document.write(x);      // To read cookie
```

name=Sarfraz; item=milk, bread

All the cookies update or saved are displaying here on the document.

We can also use the split() method to break the cookie value into keys and values.

document.cookie = "username=; expires=Fri, 07 Jun 2019 00:00:00 UTC; path=/;";

Note: You should define the cookie path option to ensure that you delete the right cookie. Some browsers doesn't allow to delete a cookie unless you specify a path parameter.

What are the differences between cookie, local storage and session storage

Below are some of the differences between cookie, local storage and session storage,

Feature	Cookie	Local storage	Session storage
Accessed on client or server side	Both server-side & client-side	client-side only	client-side only
Lifetime	As configured using Expires option	until deleted	until tab is closed
SSL support	Supported	Not supported	Not supported
Maximum data size	4KB	5 MB	5MB

https://www.youtube.com/watch?v=rxunlf6oEUw

18. What's the difference between let, const and var?

All the three are used for the variable and used to declare the method in JavaScript.

You can list out the differences in a tabular format

var

It is been available from the beginning of JavaScript

var can defined twice and assigned once.

var x; // It is declaration or it is declared or undefined

x ="Gullu" // It is called assigned//initialized.

Example:-

var x ="Sarfraz";

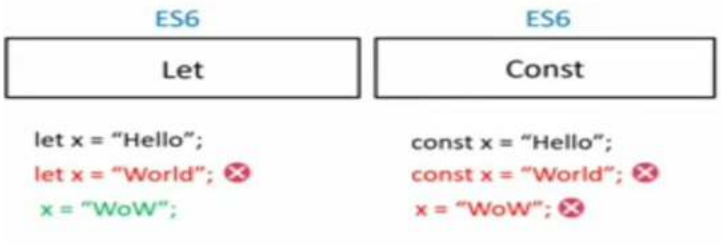
var x = "Alam";

y= "Pintu"; (= is assigned operator)

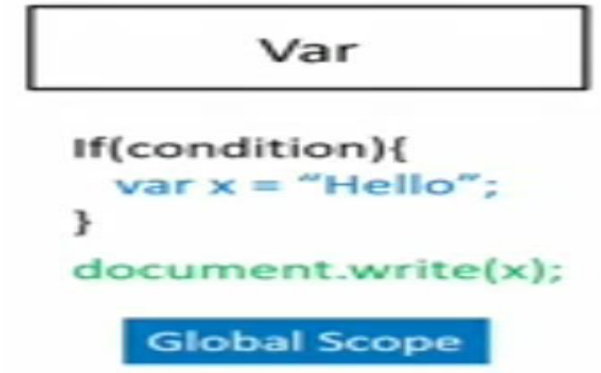
let & const

Introduced as part of ES6

let can declared once and assigned once but const can be declared once only, const can not be assigned.

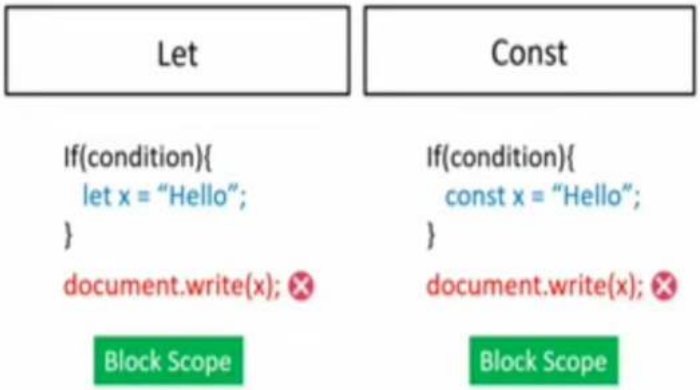


has function scope. Here, function scope mean document.write(x); will be accessed within the function only.




```
<script>
if (a=10){
var a;
document.write("Hello" + a + "<br>")
}
document.write("Hell" + a)
</script>
// Both will be printed
```

It has block scope. Here block scope means variables let and const declaration will be accessed if document.write(x); will be inside the curly braces




```
<script>
if (a=10){
let a;
document.write("Hello" + a + "<br>")
}
document.write("Hell" + a)
</script>
// Hellow will be printed
```

Variables will be hoisted

Here, variables will be hoisted means if we define/ declare variable using var keyword after document.write(), console.log() then JavaScript compiler will take the var a; (variable declaration using var keyword on the top within the scope and also define if not defined and return the value undefined;

```
<script>
document.write(a);
var a = 10;
</script>
// It will return undefined
```

`

Since hoisting will take up to the var keyword declaration

Like var a;

and JavaScript compiler is so smart that it will assign undefined value like var a=undifined; So, the output will be undefined

Hoisted but not initialized

In the case of let and const it will be hoisted but not initialized and this is why it will return an error not initialized.

```
<script>
document.write(a);
let a = 10;
</script>
// It will return reference error: can't access 'a' before initialization
```

Since In the case of let & const, hoisting will take both up but not initialize and this is why you will get an error as given below:-

Uncaught ReferenceError: Cannot access 'a' before initialization

Explantion

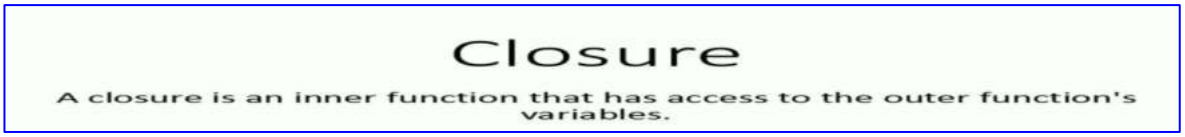
Point to remember:- The var keyword is both function and block scope and hoisted only declaration but The let and const keyword are just block scope and hoisted so return undefined but not initialized so return reference error (can't access 'a' before initialize)

```
<script>
const a = 10;  //var a = 10;  // let a =10;  Global variable will be accessed by both of them
function abc() {
const a = 10;  //var a = 10;  // let a =10; Local Variable will be accessed by document.write("Hello" + a);
document.write("Hello" + a);
}
document.write("Hell" + a);
abc();
</script>
```

```
10
11  <body>
12
13      <script>
14          function abc() {
15              if (1 == 1) {
16                  var a = "Sarfraz"
17                  document.write(a); // It will return Sarfraz as block scope starting from line 15 and ends at 19,
18                                     // if we take let or const keyword here, it will pint as both are block scope
19              }
20              document.write(a);     // It will return Sarfraz as function scope
21          }
22          document.write(a);     // It will return a is not defined  as var keyword is function scope and block scope and here line 22
23                                 //is outside the function scope becuase function abc() scope starts from line 14 and ends at 21
24
25          abc();
26
27      </script>
```

18. Define Closure? (v)

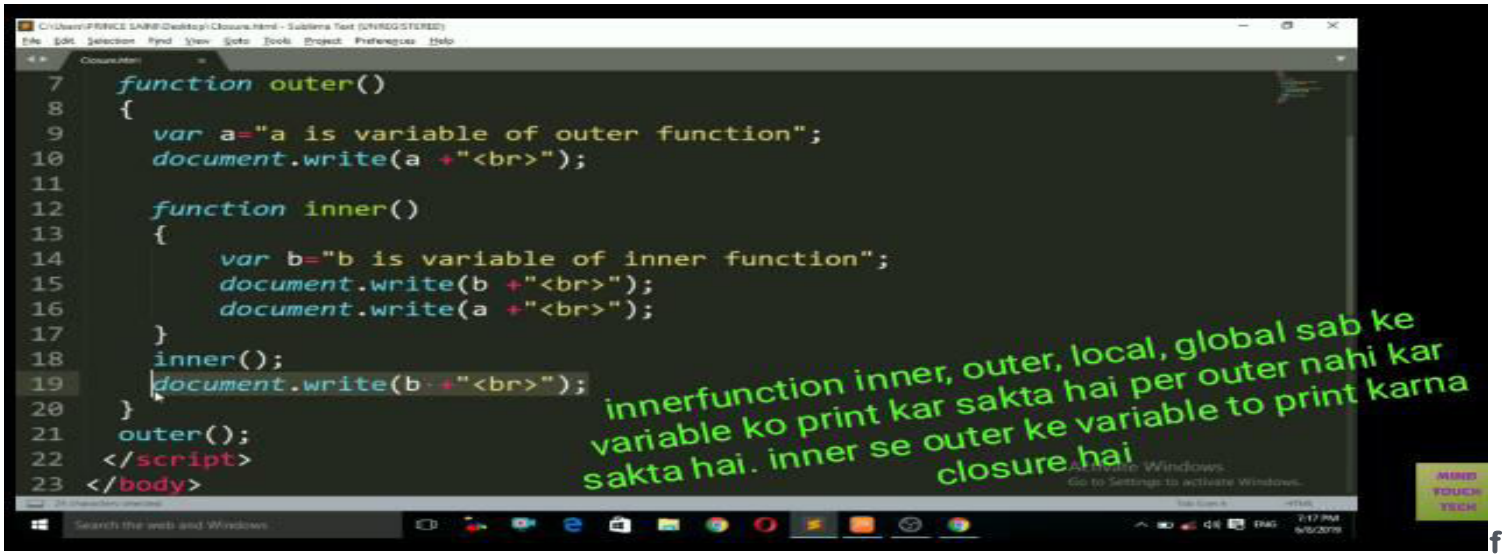
Closures provide a better, and concise way of writing JavaScript code for the developers and programmers. Closures are created whenever a variable that is defined outside the current scope is accessed within the current scope.



```
Closure.html
 4  <body>
 5   <script>
 6    function outer()
 7    {
 8       var a="a is variable of outer function";
 9       document.write(a +"<br>");
10
11       function inner()
12       {
13            var b="b is variable of inner function";
14            document.write(b +"<br>");
15            document.write(a +"<br>");
16       }
17       inner();
18    }
19    outer();
20   </script>
```

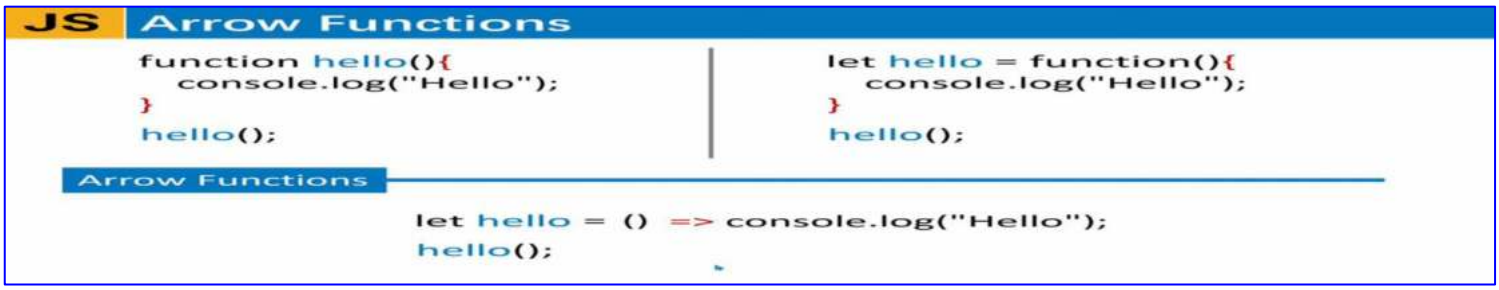
```
a is variable of outer function
b is variable of inner function
a is variable of outer function
```

```
Closure.html
 7    function outer()
 8    {
 9       var a="a is variable of outer function";
10       document.write(a +"<br>");
11
12       function inner()
13       {
14            var b="b is variable of inner function";
15            document.write(b +"<br>");
16            document.write(a +"<br>");
17       }
18       inner();
19       document.write(b +"<br>");
20    }
21    outer();
22   </script>
23  </body>
```

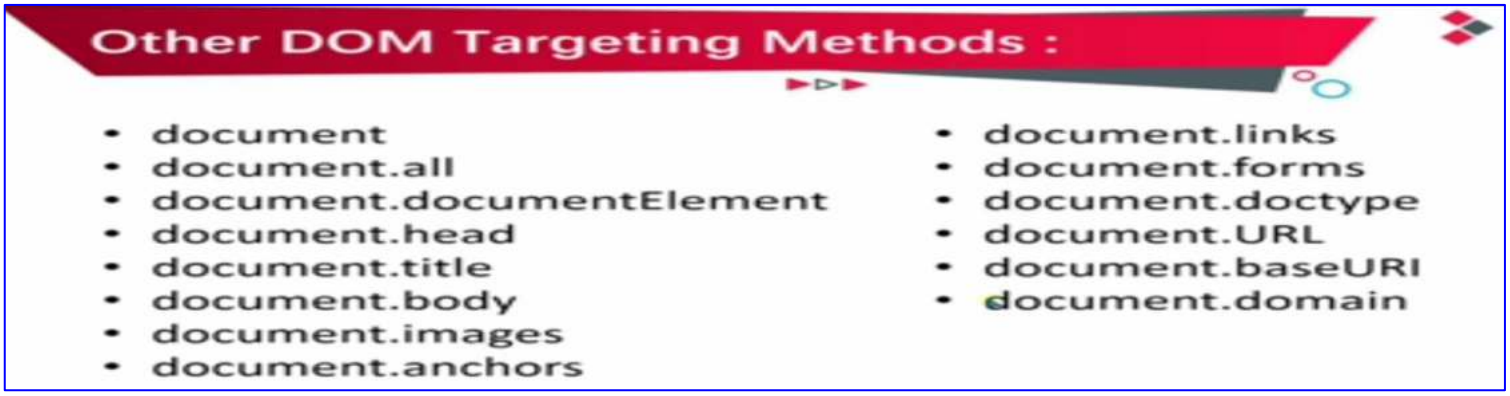


22. Define arrow function. (v)

Arrow functions are a short and concise way of writing functions in JavaScript. The general syntax of an arrow function is as below:



19. Define dom/DOM.



Here are the ways an HTML element can be accessed in a JavaScript code:

- **getElementByClass('classname'): Gets all the HTML elements that have the specified classname.**
- **getElementById('idname'): Gets an HTML element by its ID name.**
- **getElementbyTagName('tagname'): Gets all the HTML elements that have the specified tagname.**
- **querySelector(): Takes CSS style selector and returns the first selected HTML element.**

```
<body>
    <ul class="list">
        <span>Hellow</span>
    </ul>
    <div id="sarf_id">I am id </div>
    <p>I am paragraph tag</p>

    <script>
           var y = document.getElementById("sarf_id").innerText;     //I am id
            console.log(y);
            var x =document.getElementsByClassName("list");   // will target list
            console.log(x);
            var z =document.getElementsByName("p");  // will target paragarph
            console.log(z);
    </script>
```

```
</body>
```

DOM get Methods(We can target HTML, attribute and text using following targeting methods.

1. innerText
2. innerHTML
3. getAttribute
4. getAttributeNode
5. Attributes

```
    <div id="sarf_id" class="hello">
```

```
            <p>I am paragraph tag</p>
            <p>I am paragraph 2</p>
            <span class="dil">kjdajfdkaj</span>
            <b>4</b>
      </div>

      <script>

            var x = document.getElementById("sarf_id").innerText;
            var y = document.getElementById("sarf_id").innerHTML
            var z = document.getElementById("sarf_id").attributes; // It will return the attributes available in "sarf_id" only
like 0:id, 1:class so to attributes
            var k = document.getElementById("sarf_id").innerText;
            console.log("innerTest is : ", x);
            console.log("innerHTML is : ", y);
            console.log("getattribute is : ",z);
            console.log(document.getElementsByClassName("dil")[0].innerText);// It will return sarfraz because 0 index has
sarfaraz and to access innerTest used



            console.log(document.getElementsByTagName("span"));  //
```

```
▼HTMLCollection [span.dil]
  ▶0: span.dil
    length: 1
  ▶[[Prototype]]: HTMLCollection
```

```
            console.log(document.querySelector("#sarf_id").getAttribute("class"));  // hello
            console.log(document.querySelector("#sarf_id").getAttributeNode("class"));  // class="hello"
            console.log(document.querySelector("#sarf_id").getAttributeNode("class").value);

// It will return hello same as getAttribute("class")

   </script>
```

DOM set Methods

1. innerText
2. innerHTML
3. setAttribute
4. removeAttribute
5. Attribute

```
   <div id="sarf_id" class="hello">
       <p>I am paragraph tag</p>
       <p>I am paragraph 2</p>
       <span class="dil">sarfraz</span>
       <b>4</b>
    </div>
    <script>
       var x = document.getElementById("sarf_id").innerText = "Hello World";
       console.log("innerTest is : ", x);
```

```
innerTest is :  Hello World
```

```
         var y = document.getElementById("sarf_id").innerHTML = "<h1>I am changed</h1>"
       console.log("innerHTML is : ", y);
```

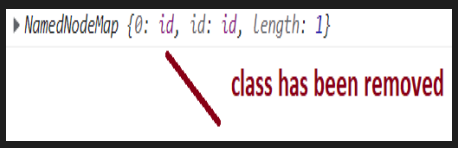
```
innerHTML is :  <h1>I am changed</h1>
```

```
      var ll = document.getElementById("sarf_id").setAttribute("class", "sar");  // first is attribute and 2nd is value.New
class was set.
       kk = document.getElementById("sarf_id").getAttribute("class");  // Here new class was accessed
       console.log("setattribute is : ", kk);
```

```
setattribute is :  sar
```

```
      <style>
```

```
      .sar h1{color: blue;}   // Used to style the new class .sar h1 used because now id “sarf_id” has h1 as set innerHTML
    </style>
```




```
    console.log(document.getElementById("sarf_id").attributes);    // SHOW ALL THE ATTRIBUTES
     console.log(document.getElementById("sarf_id").attributes[1]);  // TO TARGET INDEX ONE ATTRIBUTE
     console.log(document.getElementById("sarf_id").attributes[1].value="new_class"); // TO SET NEW CLASS IN INDEX ONE
     console.log(document.getElementById("sarf_id").removeAttribute("class"));  // CLASS HAS BEEN REMOVED.

</script>
```

In stead of targeting the element by class name/tag name or id, we can simply **use querySelector or querySelectorAll**

- document.querySelector(css selector)

  It will select the first element only if we target mutiple elements.
  document.querySelector(".sarfraz") if targeting class="sarfraz"
  document.querySelector("#sarfraz") if targeting id="sarfraz"
  document.querySelector("ul") if targeting by class name ul. It will target only the first ul tag if more that one ul tag in the document but document.querySelectorAll will target all the ul tags available in the document.

- document.querySelectorAll(css selector) : It will select all the elements we target.

DOM CSS STYLING METHOD

There are three ways to style the web content through DOM CSS styling method. We can both get and set the value.

- Style

  document.querySelector(#header).style.color;

- className

  className will help to set and get className in any id or tag name. It will return the class in string whereas classList will return the number of class in the form of Array.

- classList

  ClassList also work as className and it is another method to write className

STLYLING USING STYLE METHOD

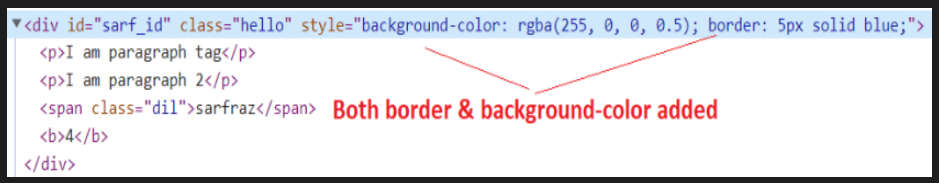
We learn how to access style attribute using JavaScript style method(Make sure that style method is already added in so that we can access else we will not be able to access.

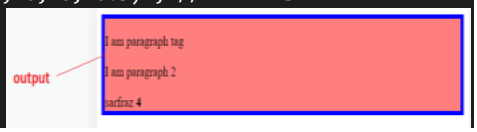
```
<div id="sarf_id" class="hello" style="border:1px solid red ; color: blue;">
    <p>I am paragraph tag</p>
    <p>I am paragraph 2</p>
    <span class="dil">sarfraz</span>
    <b>4</b>
</div>
<script>
    // To access or get the style value
```

```
        var target = document.querySelector("#sarf_id").style.border;     // It will return 1px solid red so we have accessed it
        console.log(target);
</script>
```

Now, we will change or set new style using JavaScript. It is not necessary that style is already used in the id or class. Here we will add the CSS styling using style method.

```
    <div id="sarf_id" class="hello">            // NO STYLE IS ADDED BUT IT WILL BE ADDED VIA STYLE
        <p>I am paragraph tag</p>
        <p>I am paragraph 2</p>
        <span class="dil">sarfraz</span>
        <b>4</b>
    </div>
    <script>
        // To set or add the style
        var target = document.querySelector("#sarf_id").style.backgroundColor = "rgba(255, 0, 0, 0.5)"; // ADDED STYLE
        var target = document.querySelector("#sarf_id").style.border = "5px solid blue";
        console.log(target);
 </script>
```

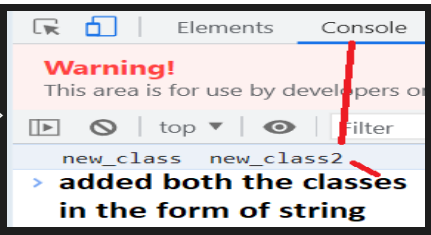


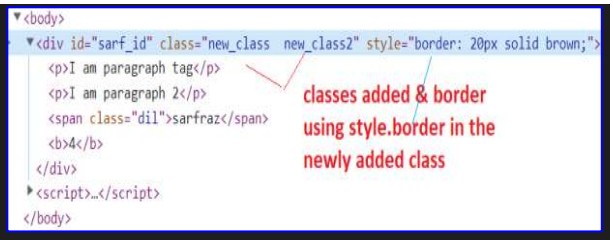


ClassName

Adding style using className will add multiple classes and display in console in a string but classList will display in an array

```
    <div id="sarf_id" class="hello">
        <p>I am paragraph tag</p>
        <p>I am paragraph 2</p>
        <span class="dil">sarfraz</span>
        <b>4</b>
    </div>
    <script>

        var nc = document.querySelector(".hello").className = "new_class  new_class2";
        console.log(nc);            // added two classes
        document.querySelector(".new_class").style.border = "20px solid brown"; // It will change the border of the newly added
class

</script>
```

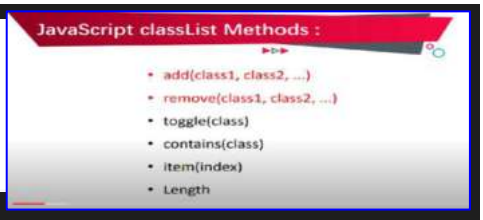


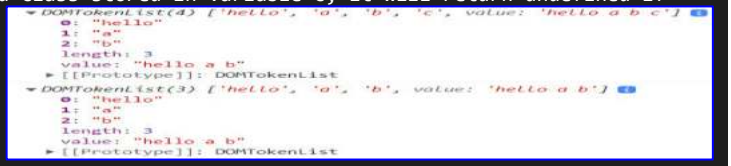


ClassList()

You can see classList returns an array of all the classes added. It can do the same what className can. It has different methods.We have used classList.add(); here to add the class and remove to remove the class.

```
 <body>
    <div id="sarf_id" class="hello">
        <p>I am paragraph tag</p>
        <p>I am paragraph 2</p>
        <span class="dil">sarfraz</span>
    </div>
    <script>
        //Using classList// class list has various methods. class list will return an array of all the class added but className
will return string of the classes added
        document.querySelector(".hello").classList.add("a", "b", "c");
        var c = document.querySelector(".hello").classList;  //added class stored in variable c, it will return undefined if
don't do so or if we store line no. 21 in any variable.
        console.log(c);
        document.querySelector(".hello").classList.remove("c");
        var d = document.querySelector(".hello").classList;
        console.log(d);
    </script>
```





```
</body>
```

Use of toggle and length(toggle is used to hide and show something on click or to add or remove a class and length will give the length of the classes added

```
<body>
    <div id="sarf_id" class="hello">
        <p>I am paragraph tag</p>
        <p>I am paragraph 2</p>
    </div>
    <script>
        var c = document.querySelector("#sarf_id").classList.length;  // It is a property of classList. It will return 1 bcoz there is only one class in html
        console.log(c);
        console.log(document.querySelector("#sarf_id").classList); // It will return an array of the all classes
```

```
1
DOMTokenList ['hello', value: 'hello']
    0: "hello"
    length: 1
    value: "hello"
  ▶ [[Prototype]]: DOMTokenList
```

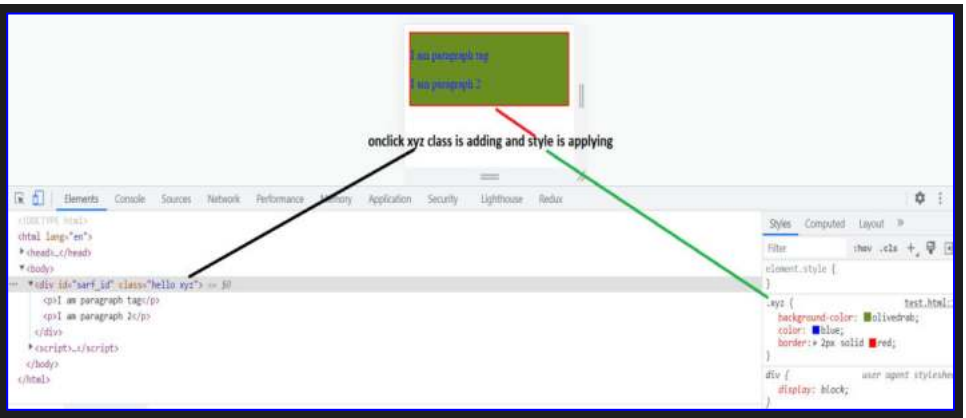
will give legth

given an array of classes

classList.toggle()

We need to put classList.toggle inside the function that we will call on button click

```
<head>
    <style>
        .xyz{
            background-color:olivedrab;
            color: blue;
            border: 2px solid red;}
    </style>
</head>
<body>
    <div id="sarf_id" class="hello">
        <p>I am paragraph tag</p>
        <p>I am paragraph 2</p>
    </div>
    <script>
        document.querySelector("#sarf_id").addEventListener("click", abc); // abc function will be called on click of div which as id="sarf_id"

        // The new class xyz will be added on click of the div.  We have styled xyz class as you see above under head tag.
        function abc(){
            document.querySelector("#sarf_id").classList.toggle("xyz");
            var t = document.querySelector("#sarf_id").classList;
            console.log(t);}
   </script>

</body>
```

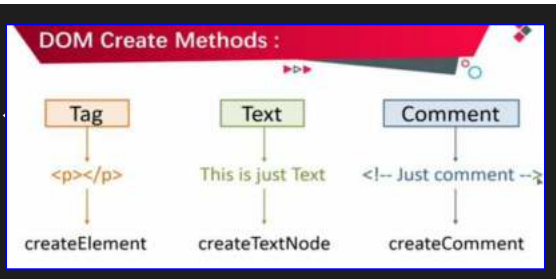


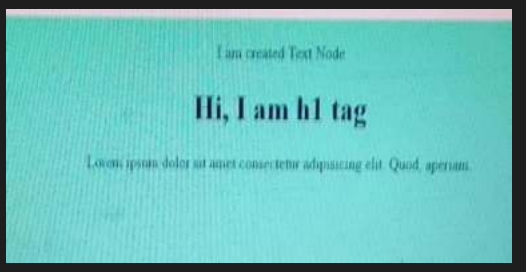
contains() and item(index)

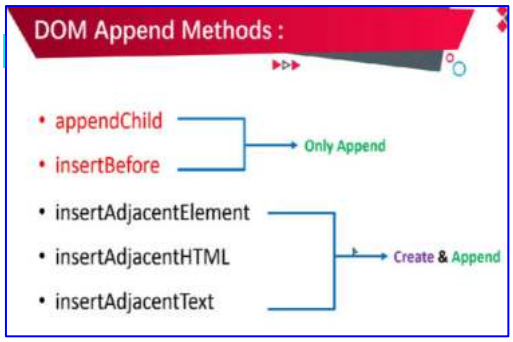
```
<body>
    <div id="sarf_id" class="hello">
        <p>I am paragraph tag</p>
        <p>I am paragraph 2</p>
    </div>
    <script>
        var c = document.querySelector("#sarf_id").classList.contains("hello");  // It will return true as hello class is there
        console.log(c);
        console.log(document.querySelector("#sarf_id").classList.item(0)); // It will return hello becoz 0 index has hello class

    </script>
</body>
```

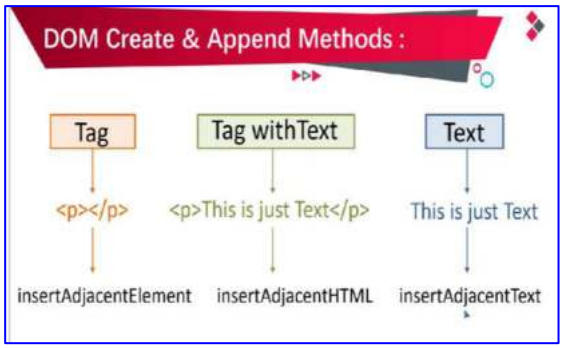
DOM CREATE & APPEND METHODS

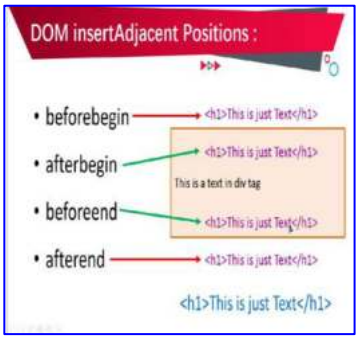
```
<body>
    <div id="ce">
        <h1>Hi, I am h1 tag</h1>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod, aperiam.
    </div>
<script>
    //document.write("Example of createElement");
    var a = document.createElement("h2");

    //document.write("Example of createTextNode");
    var b = document.createTextNode(" I am created Text Node");
    var c = a.appendChild(b);

    //document.write("Example of appendChild");
    document.querySelector("#ce").appendChild(c);


    //document.write("Example of insertBefore");
   var d = document.querySelector("#ce");
   d.insertBefore(c, d.childNodes[0]);    (first parameter whom to add, where to add
</script>
```





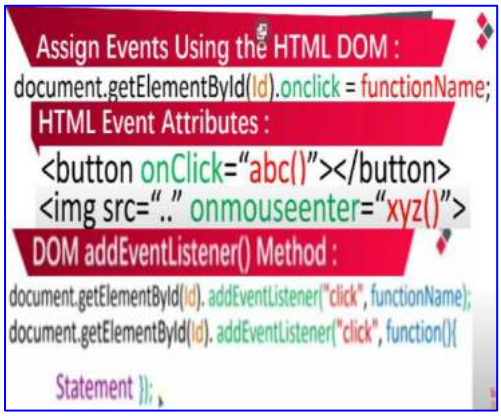


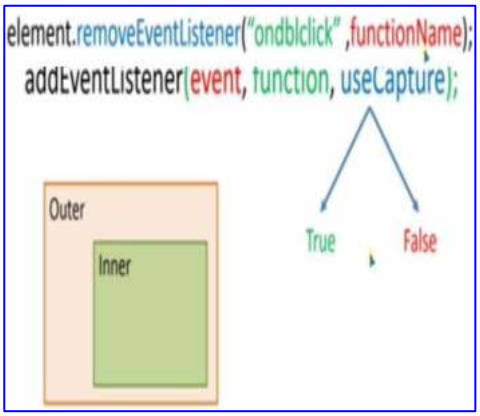





```
<body>
    <div id="cei">
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod, aperiam.</p>
    </div>
<script>
    document.write("Example of insertAdjacentElement element means adding any tag and it will not display if not var b and var c
created");


    var a = document.createElement("h2");
    var target = document.getElementById("cei");
    target.insertAdjacentElement("afterbegin", a);
    var b= "<h2> I am loving it </h2>";
    var target = document.querySelector("#cei");
    target.insertAdjacentHTML("beforebegin", b);
   var b= "I am loving it again and again";
    var target = document.querySelector("#cei");
    target.insertAdjacentText("beforebegin", b);

</script>
```

HTML EVENT ATTRIBUTES






```
<!-- Use of onclick method-->
        <button type="text" onclick="myname()">Click on me to know the creater's name</button>// ON CLICKING ON THE BUTTON,
myname() function will be called.
        <button type="text" onclick="myfame()">Click to see alert</button>
        <br><br>
        <script>
            function myname() {
                document.write("The creater is Sarfraz.");
            }
        </script>

        <!-- Use of alert method-->
        <script>
            function myfame() {
                alert("The creater is Sarfraz.");
            }
        </script>
```

DOM addEventListener()Method

Another way to write event and it is called DOM addEventListener()Method. We can also write document.getElementByTagName(). also create funciton inside the addEventListener as mentioned in 95. We will not write "on" like onClick/onmouseenter etc. We will simply write Click/DoubleClick etc. The best part of this addEventListener is that we can call multiple events as shown from 219 to 221 in my.js. addEventListener has three parameters like addEventListener(event, funciton, useCapture); useCapture willl be either true or false.

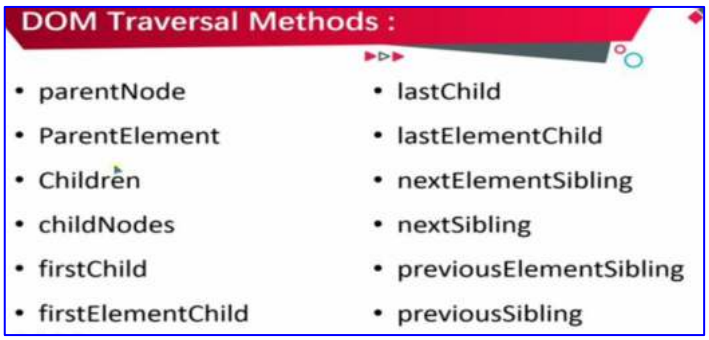
document.getElementById("id").addEventListener("Click", funcitonName);
document.getElementByClass("class").addEventListener("Click", funciton(){statement});

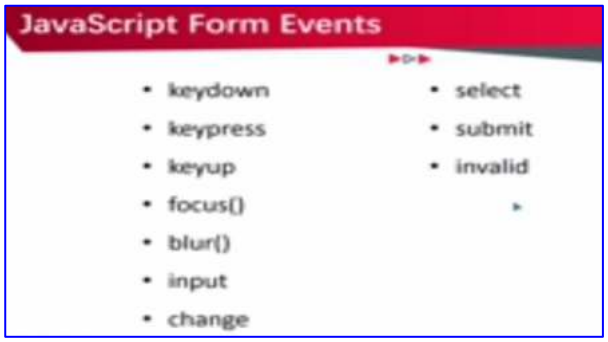
DOM removeEventListener()Method

It is used to remove any addEventListener without entering into the coding. Here is the way to do it.

element.removeEventListener("click", functionName); You can see the example in my.js in 256. Note- element=getElementById,getElementByTagName,getElementByClassName

Q. Traversal/transversing method.





BOM

C:\Users\pione\OneDrive\Desktop\FED\JAVASCRIPT\15. BOM methods

20. What are the ways of defining a variable in JavaScript?

There are three ways of defining a variable in JavaScript:

Var

This is used to declare a variable and the value can be changed at a later time within the JavaScript code.

Const

We can also use this to declare/define a variable but the value, as the name implies, is constant throughout the JavaScript program and cannot be modified at a later time.

Let

This mostly implies that the values can be changed at a later time within the JavaScript code.

21. What are Import and Export in JavaScript (Module)?

Imports and exports help in writing modular code for our JavaScript applications. With the help of imports and exports, we can split a JavaScript code into multiple files in a project. Since we separate the codes into different files or components so it is easy to maintain and organize the code which improves the code readability.

Refer to notes

21. What is the difference between Document and Window in JavaScript?

Document	Window
The document comes under the windows object and can also be considered as its property.	Window in JavaScript is a global object that holds the structure like variables, functions, location, history, etc.

23 . What are some of the JavaScript frameworks and their uses?

JavaScript has a collection of many frameworks that aim towards fulfilling the different aspects of the web application development process. Some of the prominent frameworks are:

- Angular - Frontend development of a web application
- Node - Backend or server-side development of a web application

ARNV - Angular.js , React.js (library), Native.js, Vue.js (Web Development)

Electron.js (Desktop App Development)

24. What is the difference between Undefined and Undeclared in JavaScript?

Undefined	Undeclared
Undefined means a variable has been declared but a value has not yet been assigned to that variable. var a;	Variables that are not declared or that do not exist in a program or application.

25. What is the difference between Undefined and Null in JavaScript?

Undefined	Null
Undefined means a variable has been declared but a value has not yet been assigned to that variable. var a;	Null is an assignment value that we can assign to any variable that is meant to contain no value. var a=null;

25. What is the difference between Call and Apply? (explain in detail with examples)

```
<script>
    let obj5= {
        name:"Sarfraz",
        detail:function(country){
            //return "My Name is" + " " +this.name + " I live in" + " "+ country;
            return (`My Name is ${this.name} I live in ${country}`);
        }

        }

    var x = obj5.detail("India");
    console.log(x);
</script>
```

```
<script>
    let obj5= {
        name:"Sarfraz",
        detail:function(country){
            //return "My Name is" + " " +this.name + " I live in" + " "+ country;
            return (`My Name is ${this.name} I live in ${country}`);
        }

        }

    let obj6 = {
        name: "Alam",
        detail:function(country){
            //return "My Name is" + " " +this.name + " I live in" + " "+ country;
            return (`My Name is ${this.name} I live in ${country}`);
        }
    }

    var x = obj6.detail("India");
    console.log(x);
</script>
```

OutP **OutPut**

My Name is Sarfraz I live in India My Name is Alam I live in India

Use Of Call() Method

Here, you can see that function has been taken twice in obj5 and obj6 to print obj6. Now, we have freedom to take or use function once only in obj5 or outside and then borrow that function with the help of call(), apply() and bind() method.

```
<script>
        let obj5= {
            name:"Sarfraz",
            detail:function(country){
                //return "My Name is" + " " +this.name + " I live in"
                return (`My Name is ${this.name} I live in ${country}`
            }

            }

        let obj6 = {
            name: "Alam",
            // detail:function(country){
            //      //return "My Name is" + " " +this.name + " I live i
            //      return (`My Name is ${this.name} I live in ${countr
            // }

        }
        var x = obj5.detail.call(obj6, "India");
        console.log(x);

</script>
```

OutPut

My Name is Alam I live in India

Here, we have used call() method and you can see that we are getting the same result that we were getting using function in obj6 but this time function has not been taken or defined in obj6. However, we are getting the same output because we have borrowed function from obj5 using call() method. Here, we need to call the object that has function and then dot operator and then function and then call method (first parameter will be object in which we want to borrow the function, second parameter will arguments, in call method we use arguments individually separated by comma but apply method we take arguments in an array) and this the only difference between call and apply.

Use Of apply() Method

```
<script>
        let obj5= {
            name:"Sarfraz",
            detail:function(country){
                //return "My Name is" + " " +this.name + " I live in
                return (`My Name is ${this.name} I live in ${country
            }

            }

        let obj6 = {
            name: "Alam",
            // detail:function(country){
            //      //return "My Name is" + " " +this.name + " I live
            //      return (`My Name is ${this.name} I live in ${coun
            // }

        }
        var x = obj5.detail.apply(obj6, ["India"]);
        console.log(x);

</script>
```

OutPut

My Name is Alam I live in India

Here, we have got the same output but this time we have used apply method and we know that in apply the argument or arguments an taken as an array. So, we have taken it as Example:

var x = obj5.detail.apply(obj6, [“India”]); We can multiple arguments as well as shown below:-

```
<script>
        let obj5= {
            name:"Sarfraz",
            detail:function(country, state){
                //return "My Name is" + " " +this.name + " I live in" + " "+ cou
                return (`My Name is ${this.name} I live in ${country} and My sta
            }

            }

        let obj6 = {
            name: "Alam",
            // detail:function(country){
            //      //return "My Name is" + " " +this.name + " I live in" + " "+
            //      return (`My Name is ${this.name} I live in ${country}`);
            // }

        }
        var x = obj5.detail.apply(obj6, ["India", "Jharkhand"]);
        //var x = obj5.detail.call(obj6, "India", "Jharkhand");
        console.log(x);

</script>
```

Output

My Name is Alam I live in India and My state name is Jharkhand

We can use both apply and call to print the same value but in call we will have to take arguments individually separated by comma and in apply in an array as show above.

Use Of bind() Method

```
<script>
        let obj5= {
            name:"Sarfraz",
            detail:function(country, state){
                return (`My Name is ${this.name} I live in ${country} and My state
            }

            }

        let obj6 = {
            name: "Alam",
            // detail:function(country){
            //      //return "My Name is" + " " +this.name + " I live in" + " "+ co
            //      return (`My Name is ${this.name} I live in ${country}`);
            // }

        }
        var x = obj5.detail.bind(obj6, "India", "Jharkhand");
        console.log(x());


</script>
```

Output

My Name is Alam I live in India and My state name is Jharkhand

Here, we have used bind that gives the same output. The only difference is that it doesn't get invoked instantly like call and apply because it returns a new function and we need to call that function to the value printed whenever we need as you can see in the image that we have used console.log(x()); So, here we have used x() as it return a function. Here, arguments are taken individually like in call method.

```
</script>
<h4>call, bind & apply</h4>
<script>
    let detail = function(country, state) {
        return (`My Name is ${this.name} I live in ${country} and My state name is
    }

    let obj5 = {
        name: "Sarfraz",
    }

    let obj6 = {
        name: "Alam",
        // detail:function(country){
        //      //return "My Name is" + " " +this.name + " I live in" + " "+ country
        //      return (`My Name is ${this.name} I live in ${country}`);
        // }

    }
    var x = detail.bind(obj6, "India", "Jharkhand");
    console.log(x());


</script>
```

Using function Outside

```
</script>
<h4>call, bind & apply</h4>
<script>
    let detail = function(country, state) {
        return (`My Name is ${this.name} I live in ${country} and My state name
    }

    let obj5 = {
        name: "Sarfraz",
    }

    let obj6 = {
        name: "Alam",
        // detail:function(country){
        //      //return "My Name is" + " " +this.name + " I live in" + " "+ coun
        //      return (`My Name is ${this.name} I live in ${country}`);
        // }

    }
    var x = detail.bind(obj6, "India", "Jharkhand");
    console.log(x());


</script>
```

Output

My Name is Alam I live in India and My state name is Jharkhand

Here, we have used function outside. Since it is outside. So, we need to call the function name first to borrow and then we can use call, apply and bind and inside these method the first parameter will be object name inside which we want to use the function. Here, detail.bind(obj6, "India") means we are borrowing function in obj6.

Note:- We can get the same output form all the three methods, the only difference is that in call we us arguments individually, in apply as an array and bind it will not invoke instantly as return function so need to call the function to invoke.

26. Define Hoisting in JavaScript. (v)

Hoisting in JavaScript is the default process behavior of moving declaration of all the variables and functions on top of the scope where scope can be either local or global.

Example 1:

```
abc();
function abc(){
console.log(" Sarfraz ");
}
```

// " Sarfraz" will be the output because function declaration is taken on to the top by the compiler of JavaScript even if we call the abc() function before the function definition is declared but function expression is not hoisted as you read earlier.

Example 2:

```
a = 5;
console.log(a);
// outputs 5 though the variable is declared after it is initialized
var a;
```

```
<script>
      z = 10;
      debugger;
      document.write(z);
      var z; // it will return 10
      let z;     // no output as let is not hoisted
      const x;  // We will not assign const, we can only declare it once and this is why it is red
</script>
```

27. Difference between " == " and " === " operators (with examples)

1. "==" operator is a comparison operator that used to compare the values
2. "===" operator is also a comparison operator that is used to compare the values as well as types.

Example:

var x = 3;

var y = "3";

(x == y) // it returns true as the value of both x and y is the same

(x === y) // it returns false as the typeof x is "number" and typeof y is "string"

29. Is JavaScript a statically typed or a dynamically typed language?

Yes, JavaScript is a dynamically typed language and not statically.

Statically typed language is a language in which we have to define variable type or data type before initialization like var, let, or const when we assign any value like a =10;

In statically typed language if we write like a=10; the interpreter/compiler/translator will throw an error as they compiler is not so advanced but in dynamically typed language the interpreter will add the variable type if we miss as it is very advanced.

Since, the interpreter of JavaScript add the variable type before the code execution so we don't get any error but programming languages like c, c++, java are statically typed language.

30. Passed by value and passed by reference

https://www.youtube.com/watch?v=pslr6SWXFjQ

Passed By Values Are Primitive Data Types.

When we work with primitive data types, we call passed by value and when we work with non primitive data types like array and object then we call passed by reference.

We know that primitive data types can store one value.

Like var a = 10; // It is primitive data type

Now, we will pass the value 10 of a to some other variable like

Var b = a;

Now, the value of a has been passed to b and this is called passed by values that we can check in the

console.log(a); // 5

console.log(b); // 5

Both are returning the same because a value has been passed to b.

Now both are independent because when we add 10 in b like var b = a+10;

Console.log(a) /// 5

Console.log(b) // 15

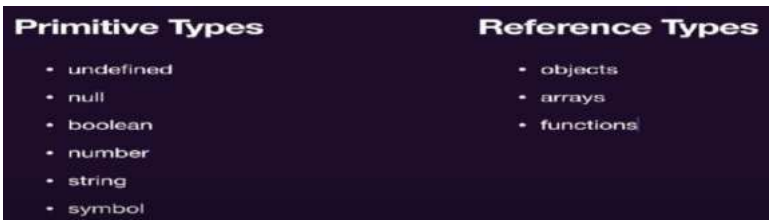
So, both are not dependant.

In short, when we pass the value both will be independent and work separately.

- **Passed by References Are Non-primitive Data Types.**

Consider the following example:

The reference of the 1st variable object i.e. 'var obj' is passed through the location of another variable i.e. 'var obj2' with the help of an assigned operator.



Example:

var obj = { name: "Raj", surname: "Sharma" };

var obj2 = obj;

Similarly, obj and obj2 are not connected to each other. They are independent and work separately.

31. Immediately Invoked Function in JavaScript (v)

An Immediately Invoked Function also abbreviated as IIFE (Immediately Invoked Function Expression) or IIFY runs as soon as it is defined. To run the function, it needs to be invoked otherwise the declaration of the function is returned.

Syntax

Step1 () ();

Step2 (enter the function)();

Step3 (function(){document.write("Sarfraz")})();

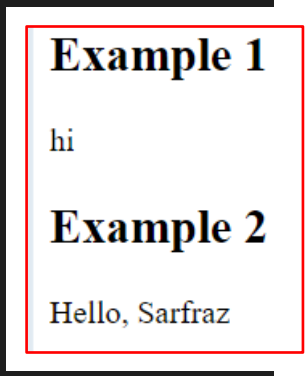
(function()

{ // Do something; })

();

It is also used to secure the data as the data inside the function and it can be accessed from the function block only not from the outside as you can see below:-

```
<body>
    <h2>Example 1</h2>
    <script>
        (function () {
            document.write("hi");
        })();
    </script>
    <h2>Example 2</h2>
    <script>
        (function () {
            var a = "Hello, Sarfraz"
             document.write(a);
        })();
        // document.write(a);
        //It will not print n this is why it is secure
    </script>
</body>
```



32. Define strict mode and also write the characteristics of JavaScript strict-mode

Strict mode was introduced in ES 5 which is a feature to check the errors. It helps to debug the code easier and less chances of making errors if we use Strict Mode. It tells the browser to check the code strictly. It is supported by all the browsers.

We need to write "use strict" before the code to apply it as shown below:-

```
"use strict";
    x=10;
console.log(x);
```

- Strict mode does not allow duplicate arguments and global variables.

```
<script>
"use strict"
function abc(a, b, b) {
```

```
var total = a + b + c;
document.write(total);
}
abc(10, 20, 10);
</script>
```

It will throw error like duplicate parameter is not allowed because b, b taken twice and if remove “use strict” it will give an error like c is not defined.

```
<script>
"strict mode"
var a = "I am variable outside the function"
function sarfraz() {  document.write(a);}
document.write(a);
sarfraz();
</script>
```

When we use global variable and use “strict mode” it does not make any difference in the output but error can be seen in the console like a is not defined.

- One cannot use JavaScript keywords as a parameter or function name in strict mode.
- All browsers support strict mode.
- Strict mode can be defined at the start of the script with the help of the keyword ‘use strict’.

33. Higher Order Functions (with examples)

Higher-order functions are the functions that take functions as arguments and return them by operating on other functions

Refer to the notes above(Q.N -12)

34. difference between exec (), match(), search(), replace() and test () methods (Regular Expression /pattern/g

exec()

It is an expression that finds a match in a string and returns the fist match. We need to make a pattern like Ga is pattern “is” also a pattern and then find that in the string. In exec() and test() first pattern is taken then dot exec()/test(“jisme search karna hai”)

```
<script>
let schoolName=  "My school name is Gandhi Ga";
let pattern = /Ga/g
// It is called regular expression used to search the pattern, g means find globally, "i" means checking/matching even if case-sensitive
document.write(pattern.exec(schoolName));
// exec() will search Ga in the string and return Ga if available else null

document.write(pattern.test(schoolName));
// test() will search Ga in the string and return boolean True if  available else false
document.write(schoolName.match("Ga"));
//It will return Ga once as regular expression not used
```

```
document.write(schoolName.match(/Ga/g)+ "<br>");
//match will do the same thing that exec does but if we use regular expression and search globally then it
        will return all Ga,
//Since Ga is available twice in the string so two Ga, Ga will be returned
document.write(schoolName.search("is"));
// search() will search "is" in the string and return position or index if available else -1
document.write(schoolName.replace("is", "are"));
//// replace() will search "is" in the string and replace with are if not available return -1
</script>
```

```
<script>
     document.write(/is/g.exec("He is a good boy. is not he?"));  // is
     document.write(/is/g.test("He is a good boy. is not he?"));  // true
     document.write("He is a good boy. is not he?".match(/is/g));  // is, is
     document.write("He is a good boy. is not he?".search(/is/g));  // 3
     document.write("He is a good boy. is not he?".replace(/is/g, "are"));  // He are a good boy. are not he?
</script>
```

- test ()
- It is an expression method in JavaScript that is also used to search a string with a specific pattern or text and return true if availble else false.

35. Define currying (v)

In JavaScript, currying is function that takes one argument at a time and return a new function expecting the next argument.

It is a conversion of a function which has parameters a and b like f(a,b) into f(a)(b).

It is created using closure function that calls the outer variable from inner function.

Why currying:-

To avoid passing the same variable again and again

To create higher order function

To make the function pure and error free.

```
  <script>
function curry(a) {
return function (b) {
return a + b;
}
}
//  let x = curry(10) // It will return a function
let x = curry(10)(20) // It will return 30
console.log(x);
  </script>
```

Write a code to get sum(10)(20)(30)

Refer the examples as shown in the image.

Question:- How will you achieve the given code output using currying?

evaluate(“sum”)(4)(2) => 6 evaluate(“multiply”)(4)(2) => 8 evaluate(“divide”)(4)(2) => 2 evaluate(“subtract”)(4)(2) => 2

```
<script>
function evaluate(operation) {
return function (a) {
return function (b) {
if (operation == "sum") { return a + b }
else if (operation == "multiply") { return a * b }
else if (operation == "divide") { return a / b }
else return "Enter Valid Data"
}
}
}
var x = evaluate("multiply")(10)(2);
console.log(x);
document.write(x);
      </script>
```

Code to get infinite argument

```
    <script>
function add(a) {
return function (b) {
if (b) return add(a + b);
return a;
}
}
var p = add(1)(2)(3)();
document.write(p);
      </script>
```

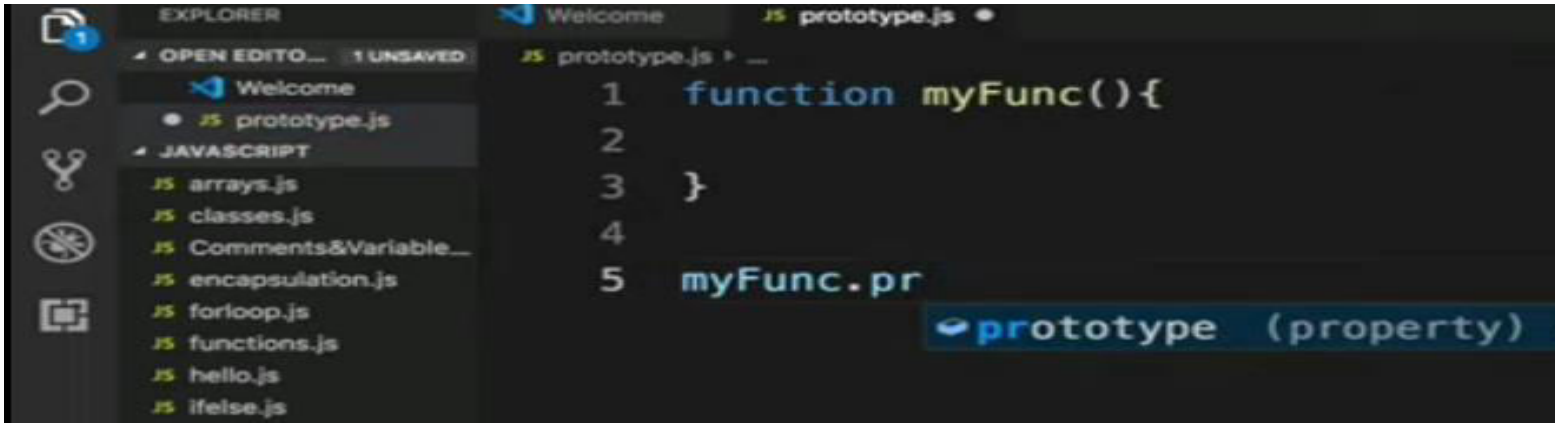
37. Advantages of using External JavaScript

- External JavaScript
- It allows web designers and developers to collaborate(kL.burate) on HTML and JavaScript files.
- It also enables you to reuse the code, code readability is simple.

38. What are object prototypes or prototype or prototype chain?

Every function has a property called prototype.

By default, this property is empty. We can add properties and methods to it. You can see here that when we use dot operator after myFunc which is function, we will get prototype which is property.



Everything in JavaScript is prototype.

A prototype is a blueprint of an object. The prototype allows us to use properties and methods on an object even if the properties and methods do not exist on the current object.

All JavaScript objects inherit properties from a prototype. For example,

Date objects inherit properties from the Date prototype

Math objects inherit properties from the Math prototype

Array objects inherit properties from the Array prototype.

Every prototype inherits properties and methods from the Object.prototype.

Let's see prototypes help us use methods and properties.

```
<script>
var arr = [];
arr.push(2);
console.log(arr); // Outputs [2]
</script>
```

In the code above, as one can see, we have not defined any property or method called push on the array "arr" but the JavaScript engine does not throw an error.

The reason is the use of prototypes. As we discussed before, Array objects inherit properties from the Array prototype.

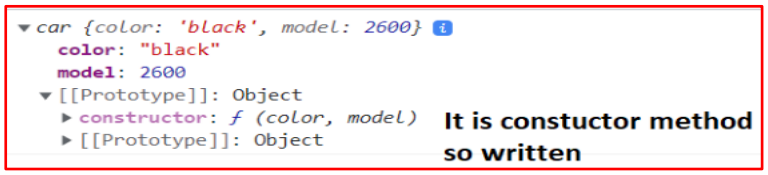
The JavaScript engine sees that the method push does not exist on the current array object and therefore, looks for the method push inside the Array prototype and it finds the method.

Whenever the property or method is not found on the current object, the JavaScript engine will always try to look in its prototype and if it still does not exist, it looks inside the prototype's prototype and so on.

Here, we have made a class name "car" which has two parameters color and model and taken two properties for color and model.

We have also made an object Maruti that has been consoled. We know that we add properties and methods it (in car class). JavaScript gives us freedom to add properties and methods to the class, Car.

```
<script>
let car = function (color, model) {
this.color = color;
this.model = model;
}
let Maruti = new car("black", 2600)
console.log(Maruti);
```



```
</script>
```

We have added a property price to the object Maruti and Suzuki.

```
<script>
let car = function (color, model) {
this.color = color;
this.model = model;
}
let Maruti = new car("black", 2600);
let Suzuki = new car("red", 2700)
Maruti.price= "12 lacs";
Suzuki.price= "19 lacs";
console.log(Maruti);
console.log(Suzuki);
</script>
```

You can see price has been added

car {color: 'black', model: 2600, price: '12 lacs'}

car {color: 'red', model: 2700, price: '19 lacs'}

```
/*Maruti.price= "12 lacs";
Suzuki.price= "19 lacs"; */
//The property using prototype is called prototype member
```

Instead of two lines code or using the same property price we have written

two lines of code for two objects. So, if we have 100 objects we will have to write 100 lines of codes but this can be done in one line using prototype

car.prototype.price= "12 lacs"Because prototype inherits all the objects or prototype is the parent of all the objects.

```
<script>
let car = function (color, model) {
//these properties using (this dot operator) are called instance member
this.color = color;
this.model = model;}
let Maruti = new car("black", 2600);
let Suzuki = new car("red", 2700)
//*Maruti.price= "12 lacs";
Suzuki.price= "19 lacs"; */
//The property using prototype is called prototype member
car.prototype.price= "12 lacs"
console.log(Maruti.price); / /12 lacs
console.log(Suzuki.price); / /12 lacs
</script>
```

40. What is the prototype design pattern?

The Prototype design Pattern is also known as a property or prototype pattern. It is a technique to copy the original object to a new object and then modify it according to our needs. The prototype design pattern comes under creation design pattern.

Where we use it

It is used to avoid extra cost, resource and time because prototype design pattern provides us ready made objects that can be modified later according to your convenience so that we don't have to start from scratch.

In another word, prototype design pattern is to create object that can be used as a blueprints for another objects that are created by constructors.

There are three types of design patterns or JavaScript Design Pattern.

Creation Design Pattern: It is used to create object which can be used according to the situation. It also allows us to copy the existing object or make single object for a class. This design pattern also allows us to create complex object if required. Prototype design pattern is an example of creation design pattern.

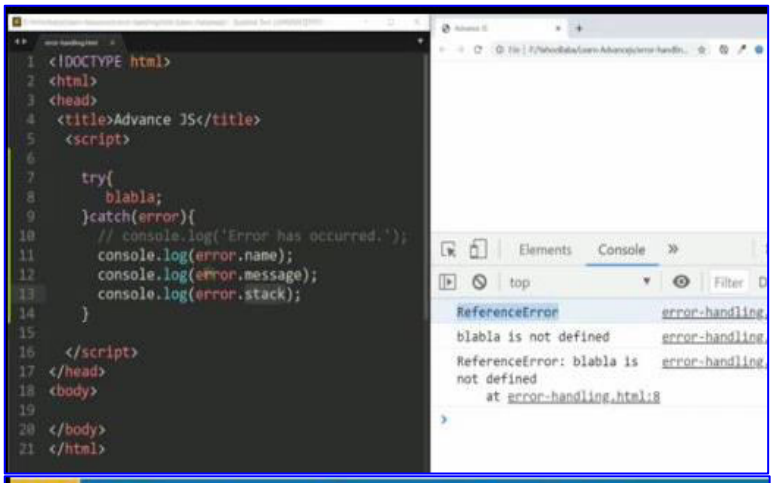
Structural Design Pattern: This design pattern focuses on class and object composition.

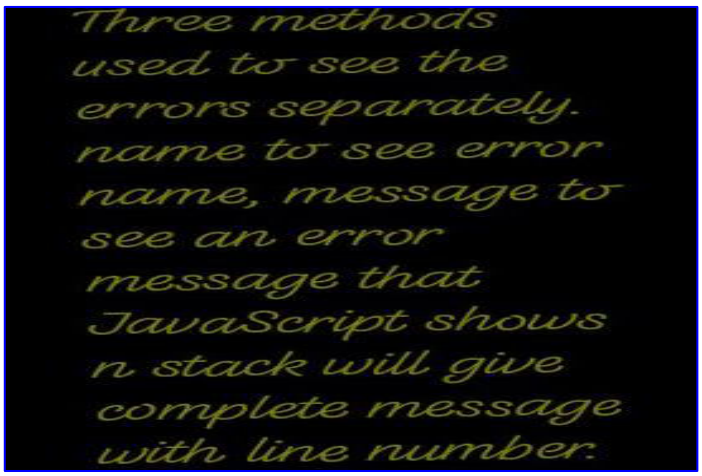
Behavioral Design Pattern: This design pattern is focused on the communication between objects.

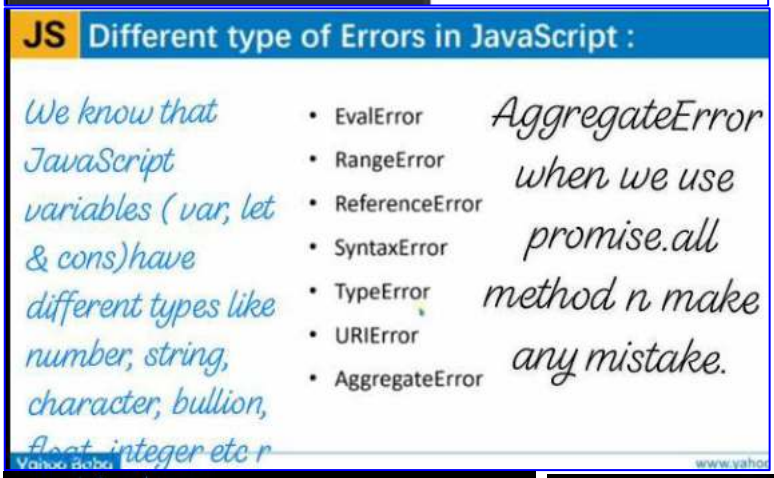
41. Types of errors in JavaScript

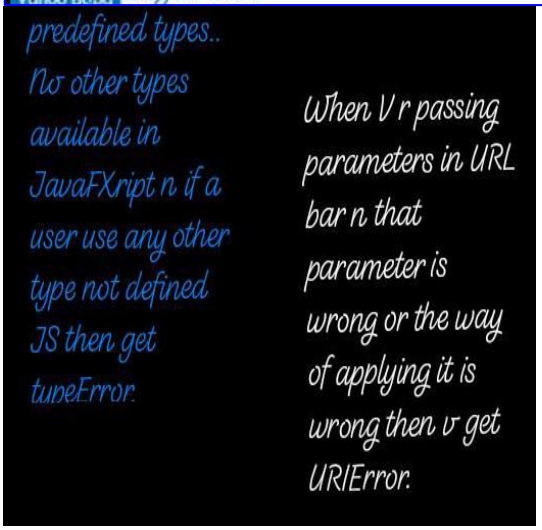
JavaScript has two types of errors, Syntax error, and Logical error.

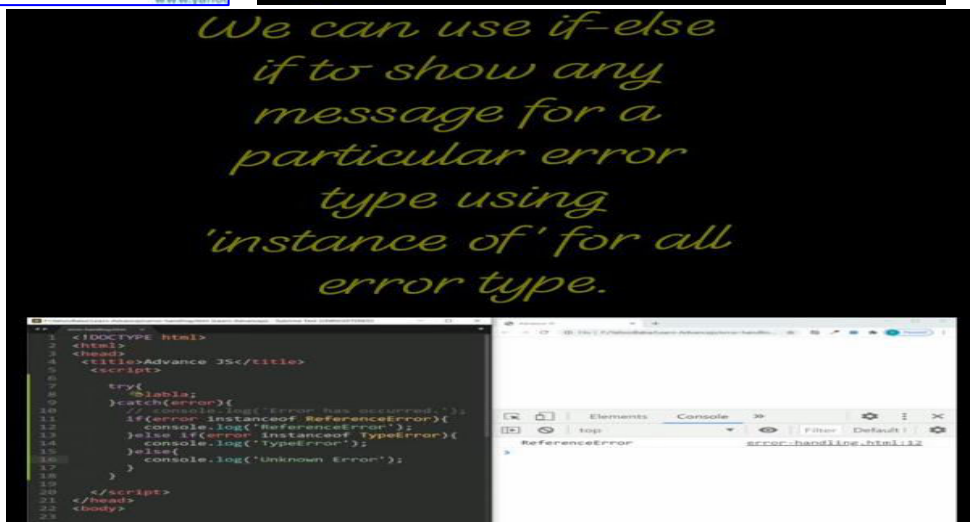
Refer to the notes.












```
          const x;
                    //    Syntax error because this is not the correct syntax of const variable becuase const can be defined
onece only
                    //    it can't be redefined nor assigned. Here we have assigned so getting syntax error.The correct syntax
will be
                    //    const x=10; as we can only define the const once.

            function abc {
                console.log('syntax error')
            }
            abc();
            // It will also return syntax error becuase () is missing after abc
                let x =10;
                let x =12;    // syntax error
                const x =10;
                 x =12;      // It will return type error because const is constand and its value can not be changed later.
            var x = ["a","b"];
          console.log(x.toFixed());   // It will return type error becuase toFixed is used with number data type but we have
used
            // array here
```

44. Use of a constructor function (with examples)

Constructor functions are used to create single objects or multiple objects with similar properties and methods. It is like a blueprint that can be reused for different objects. We don't have to create different function for different objects again and again. It saves the time and coding length.

Example:

```
function   Person(name,age,gender)
{
this.name = name;
this.age = age;
this.gender = gender;
}
var person1 = new Person("Vivek", 76, "male");
console.log(person1);
var person2 = new Person("Courtney", 34, "female");
console.log(person2);
```

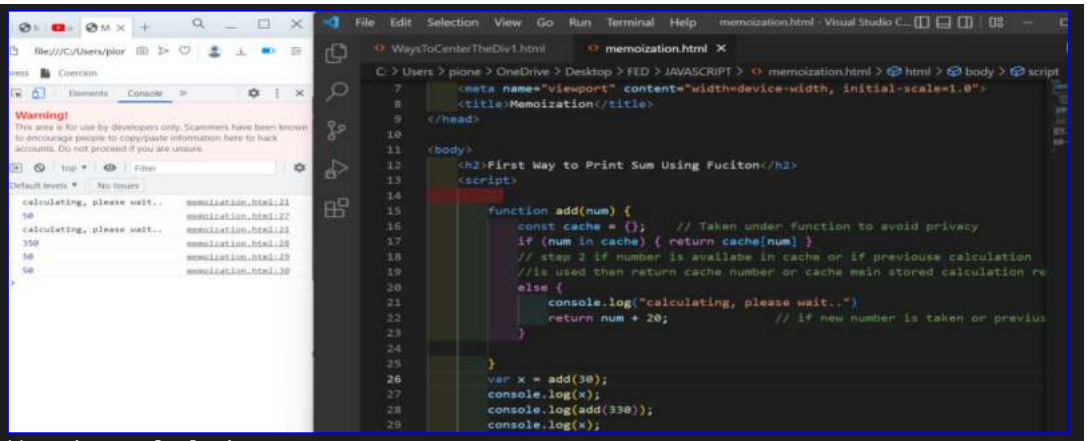
42. What is memoization?

In JavaScript, Memoization is an optimization technique that makes application more efficient and hence faster.

It used to reduce the time consuming calculations by saving the previous input to something called cache and returning the result from it.

Using Normal function To Add a Number

Here, in the given function, we have added a number to 30 and hence getting result 50. If we have to add 30 to 20 then it is okay, but what if we have to add the same 100 times, our compiler will take time to compute or calculate. So, here instead to calculating the number 100 times and increase the timing, we will use memoization(remember) that will store the output of num+20 in a place called cache and use later to give 50 value without calculating because we have saved in the memory of cache that num+20=30 so whenever we console num+20, it will return the desired vaule without calculating and hence save the calculation time.



```
<script>
    function add(num) {
        return num + 20;
    }
    var x = add(30);          //50
    console.log(x);           //50
    console.log(x);           //50
    console.log(x);           //50
    console.log(x);           //50
</script>

<script>

    const cache = {};
    //step 1 to store the repeated/previous calculation
    function add(num) {
         if (num in cache) { return cache[num] }
    // step 2 if number is available in cache or if previous calculation is
    //used then return cache number or cache mein  stored calculation result
        else {
            return num + 20;
    // if new number is taken or previuse one is not taked add number with 20
        }
    }
    var x = add(30);
    console.log(x);         // result will be calculated
    console.log(add(30));   // result will be calculated
    console.log(x);         //taken from the store as previous
    console.log(x);         //taken from the store as previous
</script>
```

```
So, calculation takes place twice only but printed four values so saved two calculation timing.
```

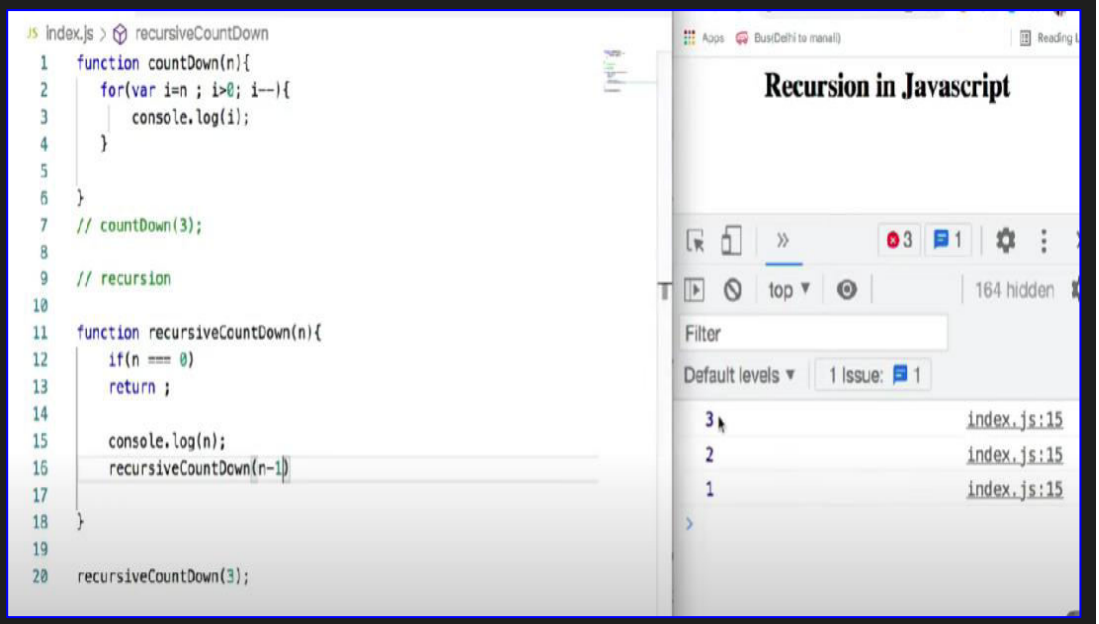
You can see first time second time values are being calculating from the third time value is being taken out from the cache as old value are being repeated. So, this is how we can avoid calculation time and make the application faster using memoization.

43. Recursion in a programming language Or How to flatten Array of Array?

Recursion is a process of calling itself and a function that calls itself is called recursive function.

Recursion is a technique in a programming language that is used to iterate over an operation whereas a function calls itself repeatedly until we get the result.

```
function test(){
                test();    // This is a recursive function becoz it is calling itself
                           // If we print this, it will print infinite times as no base case or terminating condition or exit
condition
                           // So, we need one terminating condition in recursive function else it will calls itself again and
again.
            }
            test();
            <script>
        //Recursion
        function test(n) {
            if (n == 0) {
                return;
            }
            console.log(n);
            test(n - 1);
        }
        test(3);
        // For loop
        for(i=3; i>0; i--){
            console.log(i);
        }
        //output for both the code
        3
        2
        1
    </script>
    <script>
        Recursion
        function test(n) {
            if (n > 10) {
                return;
            }
            console.log(n);
            test(n + 1);
        }
        test(1);
        i=1;
        while(i<=10){
            console.log(i);  // Both will print 1 to 10 in different rows
            i++
        }
```




```
    </script>
```

Using flat Method we can [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
    <script>
        arr = [1, 2, [3, 4], 5, [[6]], [7, 8], 9, 10];
        let x = arr.flat(2); // 2 shows depth like [[]] 2 depth used to get answer
        console.log(x);
        document.write(x);
    </script>
```

Second Way to Dynamically get the data using recursion

```
    <script>
```

```
    arr = [1, 2, [[[3, 4]]], 5, [[6]], [7, 8], 9, 10];
    var output = [];
    function flatten(arr) {
        for (let i = 0; i < arr.length; i++) {
            if (Array.isArray(arr[i])) {
                flatten(arr[i]);                // used recursion to call the function again & again until we get result
            } else {
                output += arr[i];
            }
        }
        return output;
    }
    document.write(flatten(arr));
</script>
```

https://m.youtube.com/watch?v=dBk5CJYSr5k&list=PLC8OkhrVTHNFITtT_TzfBbn8SohEU8Cpa

45. Which method is used to retrieve a character from a certain index?

We can retrieve a character from a certain index with the help of charAt() function method.

46. What is BOM?

BOM stands for browser object model, here browser means browser window. browsers can be any browsers like chrome, opera, edge, internet explorer, Mozilla Firefox etc. We will use browser object(name of browser) to make changes in/manipulate the browsers. We can get browser height and width by using window object or many things as mentioned in the picture. We can open and close the window etc. We can resize or move by one click not by using the mouse.

Please note that it will not count the height of tool bar in gray in color and URL bar if we talk about height of the window and in width, the width of scroll bar will not be added. It is used to target any element in HTML.

```
<script>
        document.write(`Inner height :  ${window.innerHeight}`);    // Will give inner height.
        console.log("Outer height " + window.outerHeight);
        console.log("Inner width " + window.innerWidth);
        console.log("Outer width " + window.outerHeight);
</script>
```

open & close methods are used to open & close a browser new widow. We can open and close a new window by clicking on any button or image etc. window.open & close are pop messages and you need to allow pop up blocked (settings>search for pop up and enable under site settings) once to see the new window to open. The unit of window open is always in px.

open().mehtod is written as window.open(URL, name, specs); name(It is optional & it will not make any changes, you can use it or not): has value like

_blank to open the window in new page. It is a default value,

_parent means on the same window,

_self means on the same frame if using frame.

_top means on the new frame window.

specs decides the width, height of the new window that will open and position of the new window like how far you want the new window from left, top.

Note :- height and width of newly open window will be small or big only if opened in the new window not in the same windo.

```
    <button onclick="openWindow()">Open the Window</button>
    <button onclick="closeWindow()">Close the Window</button>
<script>
    function openWindow() {
      var close =   window.open("https://www.google.com/","_blank", "width=400px, height=100px, left=500px, top=500px");
    }
    function closeWindow() {
```

```
            window.close(close);     // To close the opened window on button click
        }
    </script>
```

moveTo() & moveBy() methods **moveBy() method is used to move the window and it works in relative position like open & close methods are used to open & close a browser new widow. moveTo() method will not work with different URL, it will work only if you open your own website or if you want to show any content in pop up inside your website. It works on absolute position. moveTo will move from absolute which means form left & top corner. If we assign any values like left=100px, top=50px then it will move 100px form left corner and 50px down from top but moveBy will move relative wise which means if already given any left and top values then it will move after the given value. For example, if a window has been assigned the value of left=100px, top=50px then moveBy will move the window after 100px far from left and 50px far from top.**

scrollTo() & scrollBy() methods **these are two methods to scroll the window bar horizontally and vertically. If you want to scroll the window bar vertically, you will use X axis and sign -(minus) to scroll up and sign +(plus) to scroll down vertically. If you want to scroll the window horizontally, you will use Y axis and sign -(minus) to scroll and left side +(plus) to scroll right side horizontally. In short, (10 0) if the first value is more than 0 which means it will move horizontally. +10 or 10 means right had size movement and -10 means left hand side movement (0 10) if the second value is more than 0 which means it will move vertically. +10 or 10 means top to down movement and -10 means down to up movement scrollTo will take the scroll bar to the top/down or left/right if (0 0). scrollTo works on absolute position but scrollBy works on relative position**

```
    <button onclick="openWindow()">Open the Window</button>
    <button onclick="moveWindow()">Close the Window</button>
    <button onclick="resizeWindow()">Close the Window</button>
    <script>
        var myWindow;
        function openWindow() {
            myWindow = window.open("", "", "width=200px, height=100px, left=100px, top=100px");
            // We can also move the widow even if set left/top value and window will open 100px far from left and top
            myWindow.document.write("<p> I am Sarfraz </p>");
        }
        function moveWindow() {
            myWindow.moveTo(200, 200);//moveTo will move from relative position which means form left & top corner
            myWindow.focus();//
        }

        function resizeWindow() {
            myWindow.resizeBy(500, 500);
            myWindow.focus();//
        }
    </script>
```

Scroll bar

```
    <button onclick="scrollWindow()">scroll down window</button>

    <p style="width: 200px;">Lorem, ipsum dolor sit amet consectetur adipisicing elit.
        Temporibus dignissimos labore fuga distinctio quasi,doloribus consequatur ullam libero commodi, harum eveniet, saepe ad
sit est minus voluptas ipsa cumque numquam
        rerum porro nostrum facere alias. Repellat ratione dolorem officia ut omnis numquam aliquam maxime assumendapariatur
        facilis nulla consectetur saepe delectus rem cumque, sed nemo nobis laborum reiciendis incidunt quod!Error repellat ex
        asperiores, unde doloribus ea illum dicta facilis architecto rem. Atque beatae mollitia
        esse fuga, voluptatum commodicupiditatefugiat----------------------------------------------------------------------------
-------------------------------------------------   iure deserunt aliquid sequi enim fugit?Laboriosam esse officia libero
quas quidem repudiandae doloremque error recusandae eos reiciendis provident,
        numquam labore soluta deserunt dolores dolore ad non placeat obcaecati corrupti inventor e sit aut ratione
        Tat
    </p>
 <script>
       function scrollWindow() {
            window.scrollBy(0, 20);//It will scroll up from down by 20px on every click.
            window.scrollBy(20, 0);//It will scroll left to right from down by 20px on every click.
        }
```

```
</script>
```

48. Rest parameter and spread operator

- Rest Parameter(...)
- Rest parameter was introduced in ES6 which allows to handle or access multiple parameters of a function very easily in a form of an array. It is taken as a last parameter in any function which starts with three dots.
- THIS WAS THE ISSUES EARLIER AS SHOWN BELOW

```
<script>

            function abc(a, b){
               document.write(a+b + "<br>");
            }
            abc(10, 20);  // 30
            abc(10, 20, 30);  // 30 which is wrong
</script>
```

- THE ISSUE WAS RESOLVED BY REST PARAMETER OR **WRITE A PROGRAMMING TO ADD//MULTIPLY/DIVIDE MULTIPLE NUMBERS**

```
<script>
            function abc(...restOperator){
               let sum = 0;
               for( let i in restOperator){
                  sum=sum+restOperator[i];
               }
               document.write(sum + "<br>");
            }
            abc(10, 20);  // 30
            abc(10, 20, 30);  // 60 which is right
            abc(10, 20, 30, 40);  // 100 which is right
</script>
```

- BOTH STRING AND REST OPERATOR

```
<script>
            function abc(a, ...restOperator){
               document.write(`Hello ${a}`)
               let sum = 0;
               for( let i in restOperator){
                  sum=sum+restOperator[i];
               }
               document.write(`, your marks is ${sum} <br>`);
            }
            abc("Sarfraz", 10, 20);  // Hello Sarfraz, your marks is 30
</script>
```

- Spread Operator(...)

We know that rest operator is used to access/handle multiple parameters of a function very easily in a form of an array and Spread operator is used spread that array or any array . It is also denoted by 3 dots . It is used to spread the element or value of an array.

```
        let array = [10, 20, 30];
        console.log(array);  // Array(3) [ 10, 20, 30 ]
        console.log(...array);  // 10 20 30
```

```
<script>
        let obj = {name:"Sarfraz", age: 25}
        console.log(obj);  // Object { name: "Sarfraz", age: 25 }
        console.log(...obj);  // Uncaught TypeError: obj is not iterable
```

```
</script>
```

We can iterate even object but property/key and value will not spread

```
<script>
        let obj = {name:"Sarfraz", age: 25}
        console.log({...obj});  // Object { name: "Sarfraz", age: 25 }
</script>
```

50. Classes in JavaScript

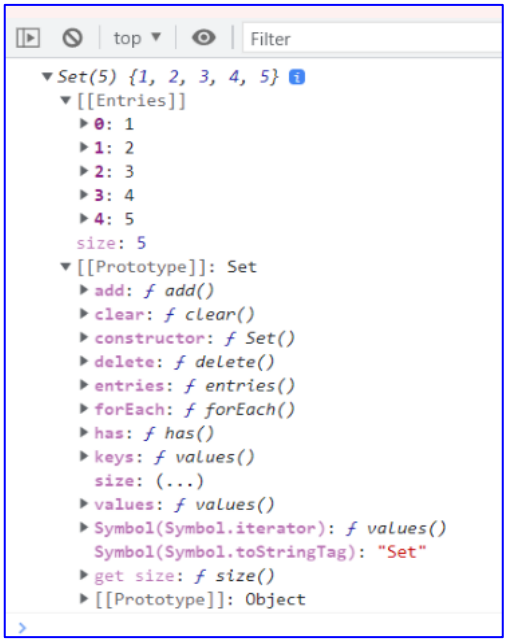
classes are syntactic sugars for constructor functions mentioned in the ES6 version of JavaScript. Classes are not hoisted-like Functions and can't be used before it is declared. Also, it can inherit properties and methods from other classes with the help of extended keywords. If the strict mode ('use strict') is not followed, an error will be shown.

52. Define Set, WeakSet, Map and Weakmap?

Set:-Set is a collection of unique data and need to make object constructor to use set.

For example:- let mySet = newSet();

We know that everything in JavaScript is prototype. We have prototype in Set as well as you can see below:-




```
<script>

    let ary=[1,2,3,2,4,5,4];

    mySet= new Set(ary);

    console.log(mySet);  // It is return an object of {1,2,3,4,5}
    console.log(mySet.size);   //The number of element/values available in array.
                               //Here, it will be 5 because there are 5 unique values in array.
    console.log(mySet.add(7)); // It is return an object of {1,2,3,4,5,7} or {0:1,1:2,2:3 etc}
    console.log(mySet.delete(7)); // It is return an object of {1,2,3,4,5,7} or {0:1,1:2,2:3 etc}
    // We can add anything like array,object, string, and integer etc.
    console.log(mySet.has(20)); // It will return yes if 20 is available in the given array else false.
    console.log(mySet.clear); // It will make the array empty.return yes if 20 is available in the given array else false.
</script>
```

Map:-In map function or method of JavaScript we store data in the form of key and value and need to make object constructor to use set.

For example:- let myMap = newMap();

We have stored the data in the form of key and value as given below:-

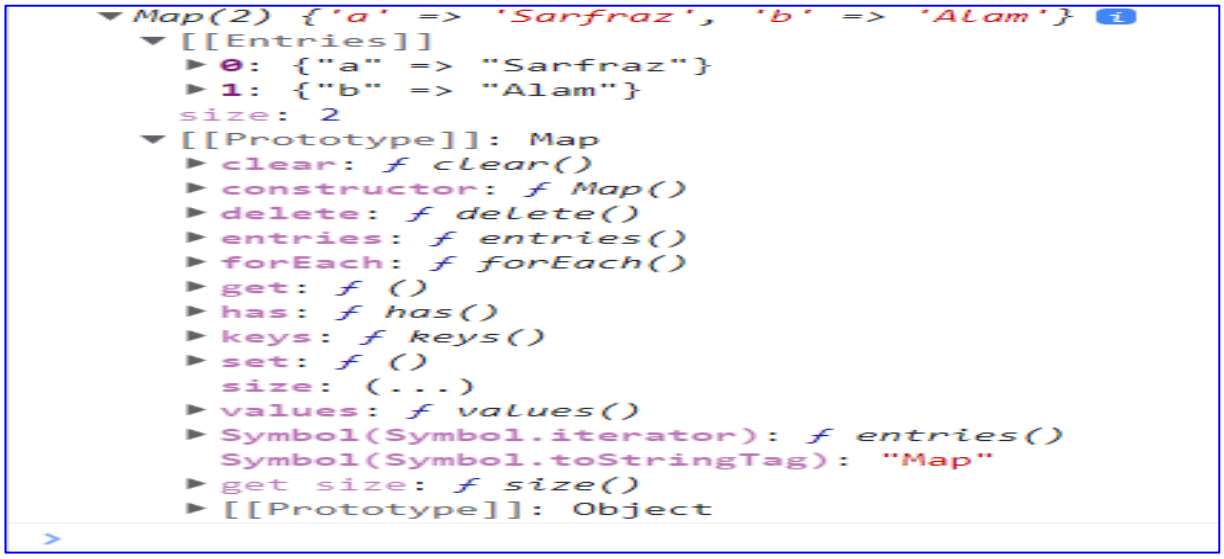
```
    let ary1=[["a","Sarfraz"],["b","Alam"]];
```

This is how we can console the map

```
    mySet= new Map(ary1);

      console.log(mySet);
```

We know that everything in JavaScript is prototype. We have prototype in Set as well as you can see below and we can use all the functions like set(add in set), get()delete in map as well.

OUTPUT AND PROTOTYPE OF MAP

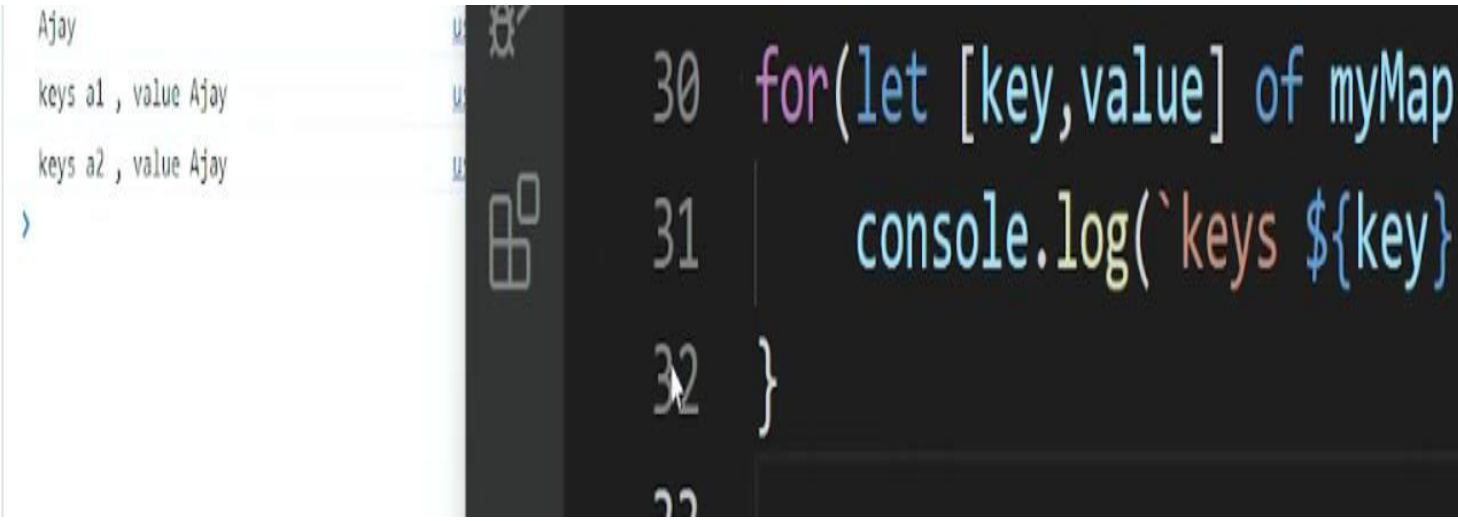



```
let ary1=[["a", "Sarfraz"],["b", "Alam"]];
myMap= new Map(ary1);
myMap.set("c", "Pammi")
console.log(myMap);
```

OUTPUT

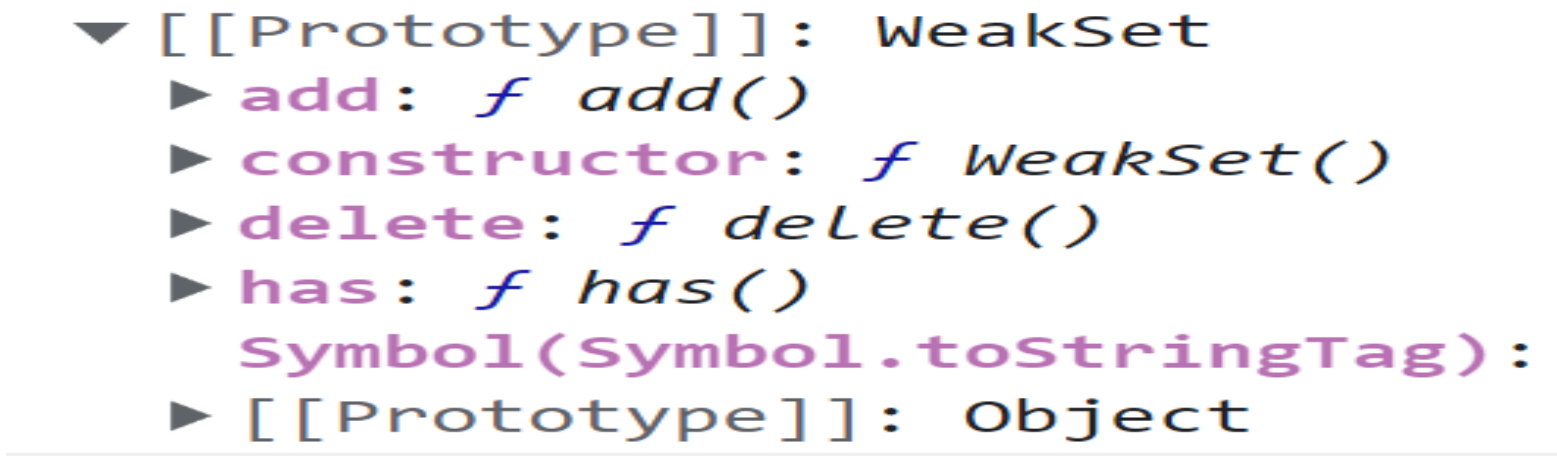
```
Map(3) {'a' => 'Sarfraz', 'b' => 'Alam', 'c' => 'Pammi'}
console.log(myMap.get("b")); // It will return Alam
```

We can also iterate the key and value using for of and forEach as shown below:- forEach shown above.



Weakset:-The JavaScript WeakSet object can store object only and we can't iterate its values using for of and for each loop.

WeakSet only has three methods, add() , delete() and has() as you can see below in the image.



Set Example

```
const newSet = new Set([4, 5, 6, 7]);
console.log(newSet);// Outputs Set {4,5,6,7}
```

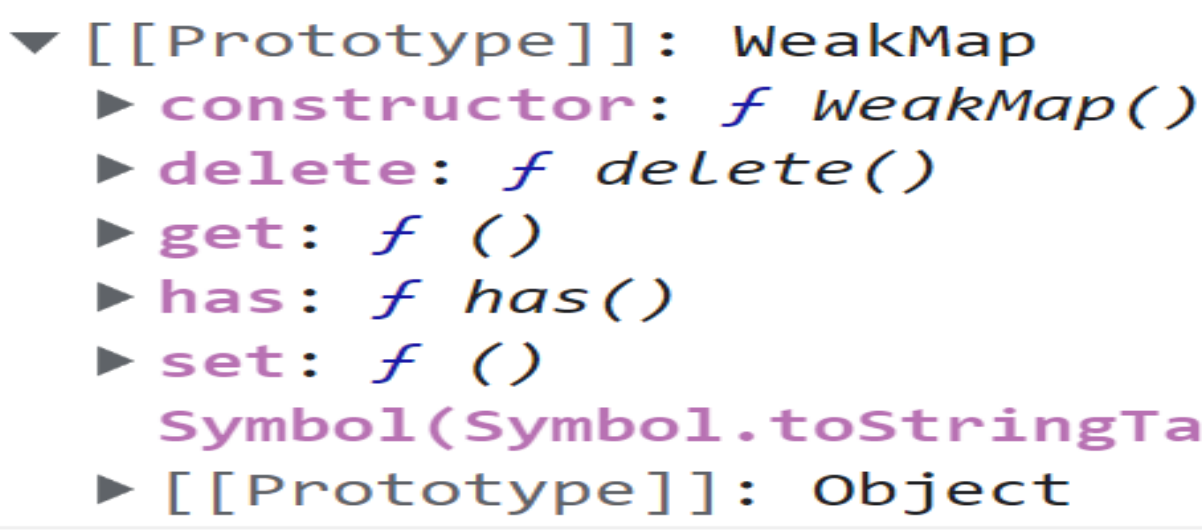
WeakSet Example

```
const newSet2 = new WeakSet([3, 4, 5]); //Throws an error
let obj1 = { message: "Hello world" };
const newSet3 = new WeakSet([obj1]);
console.log(newSet3.has(obj1)); // true
```

```
<script>

    let myWeakSet = new WeakSet();
    let ob1 = {"name":"sarfraz", "School":"sarfu" };
    myWeakSet.add(ob1); // We have stored object using add
    //myWeakSet.add(1); // We can't not add value as per definition and it wil
    console.log(myWeakSet.has(ob1)); // It will return true because ob1 is avai

    console.log(myWeakSet);

</script>
```

WeakMap: The JavaScript WeakMap object can store object only in the form of key and value and we can't iterate its values using for of and for each loop.The key-value pairs can be of both primitive and non-primitive types.

WeakSet only has methods like get(),set(),delete() and has() as you can see below in the image.



Map Example

```
const map1 = new Map();
map1.set('Value', 1);
```

WeakMap Example

```
const map2 = new WeakMap();
map2.set('Value', 2.3); // Throws an error
let obj = { name: "Vivek" };
const map3 = new WeakMap();
map3.set(obj, { age: 23 });
```

52. What is Object Destructuring? (with examples)

Object destructuring is a method to extract elements from an array or an object.

Example 1: Array Destructuring

```
const arr = [1, 2, 3];
const first = arr[0];
const second = arr[1];
const third = arr[2];
```

Example 2: Object Destructuring

```
const arr = [1, 2, 3];
const [first,second,third,fourth] = arr;
console.log(first); // Outputs 1
console.log(second); // Outputs 2
console.log(third); // Outputs 3
```

53. Prototypal vs Classical Inheritance

- Prototypal Inheritance
- Prototypal inheritance allows any object to be cloned via an object linking method and it serves as a template for those other objects, whether they extend the parent object or not.
- Classical Inheritance

- Classical inheritance is a class that inherits from the other remaining classes.

54. What is a Temporal Dead Zone?

Temporal Dead Zone is a behavior that occurs with variables declared using let and const keywords before they are initialized.

55. Difference between Async/Await and Generators

- Async/Await
- Async-await functions are executed sequentially one after another in an easier way.
- Async/Await function might throw an error when the value is returned.
- Generators
- Generator functions are executed with one output at a time by the generator's yield by yield.
- The 'value: X, done: Boolean' is the output result of the Generator function.

56. Role of deferred scripts

The Deferred scripts are used for the HTML parser to finish before executing it.

57. What is Lexical Scoping?

Lexical Scoping in JavaScript can be performed when the internal state of the JavaScript function object consists of the function's code as well as references concerning the current scope chain.

58. What is this [[[]]]?

This '[[[]]]' is a three-dimensional array.

```
var myArray = [[[]]];
```

60. How do you empty an array in JavaScript?

There are a few ways in which we can empty an array in JavaScript:

- By assigning array length to 0:

```
var arr = [1, 2, 3, 4];
arr.length = 0;
```

- By assigning an empty array:

```
var arr = [1, 2, 3, 4];
arr = [];
```

- By popping the elements of the array:

```
var arr = [1, 2, 3, 4];
while (arr.length > 0) {
arr.pop();
}
```

- By using the splice array function:

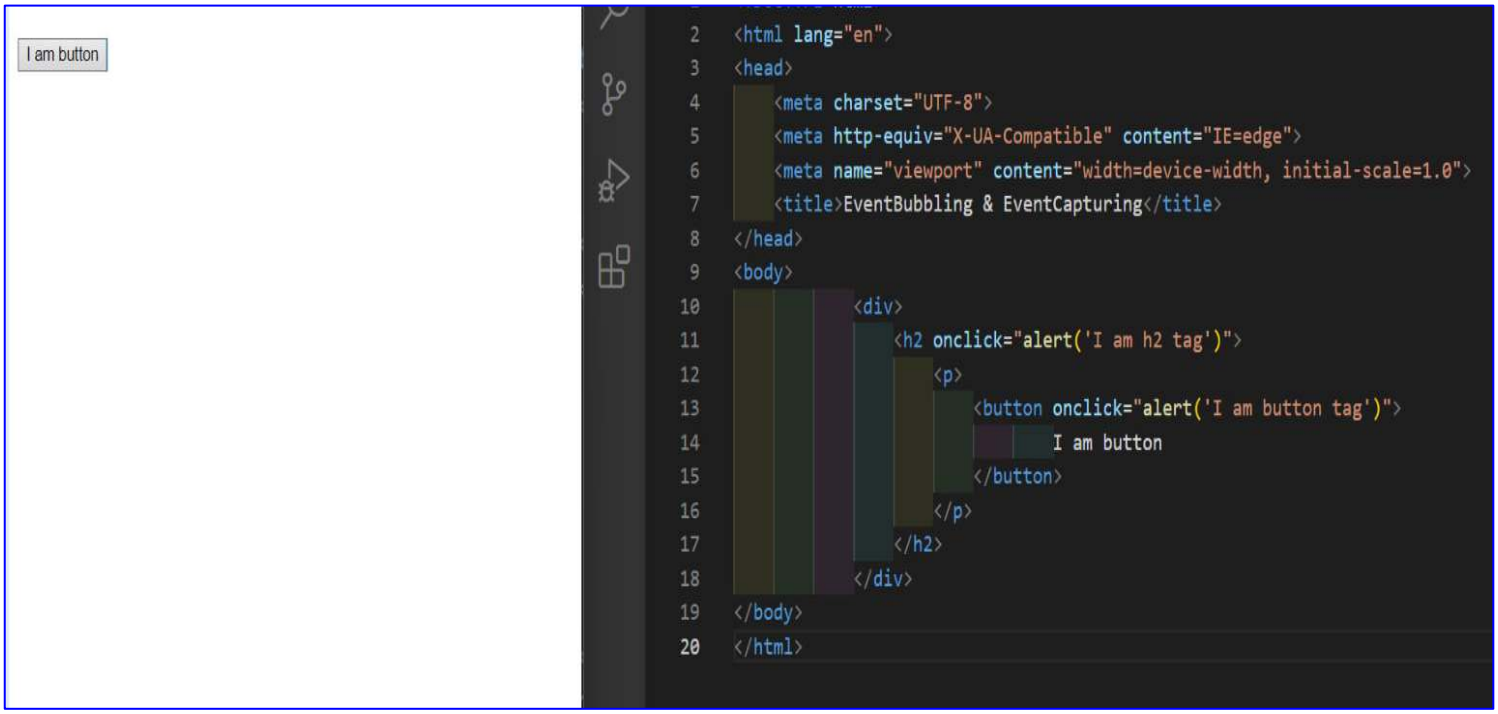
```
var arr = [1, 2, 3, 4];
arr.splice(0, arr.length);
```

61. What is the difference between Event Capturing and Event Bubbling?

Event is an action of JavaScript that calls any function on clicking on any button or anything.

Bubbling happens with all the events like on click, on hover, on change etc.

Suppose we have div and inside it we have h2 tag and inside it p tag and inside it we have button tag as you can see below:-

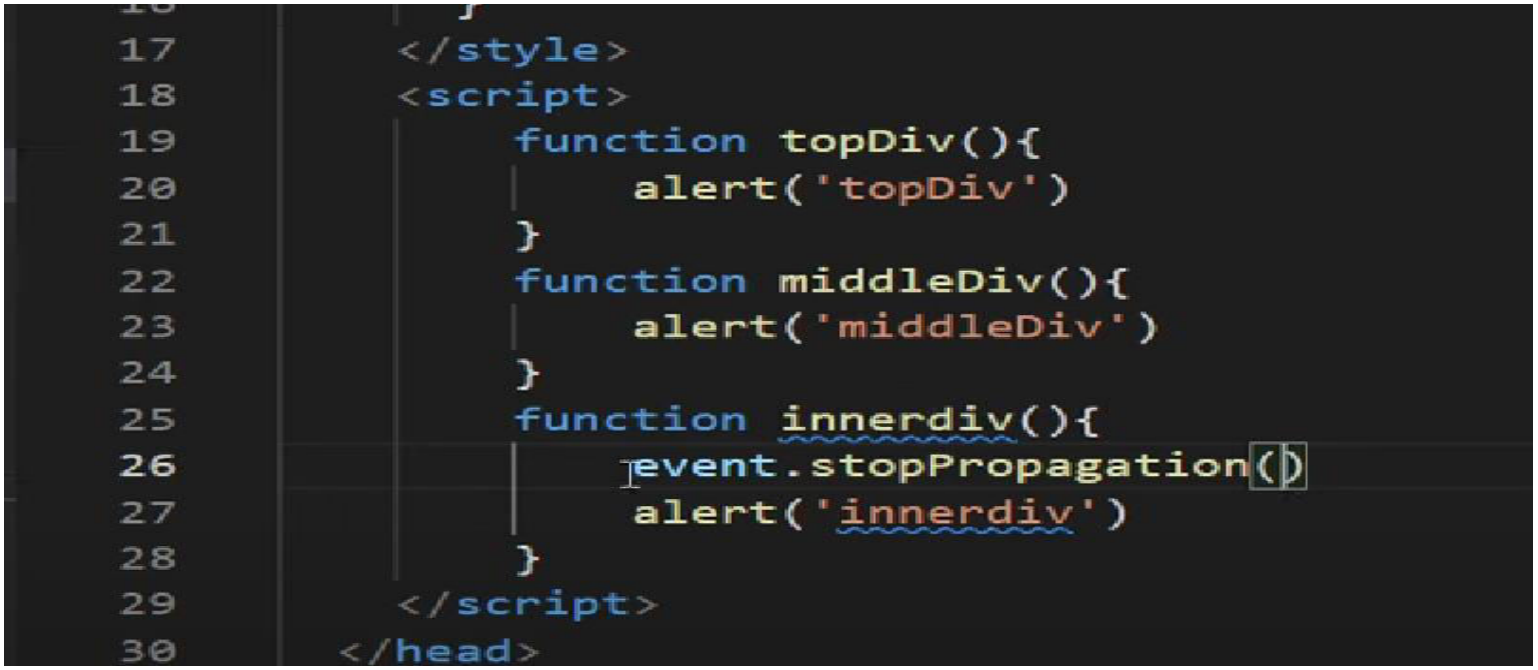


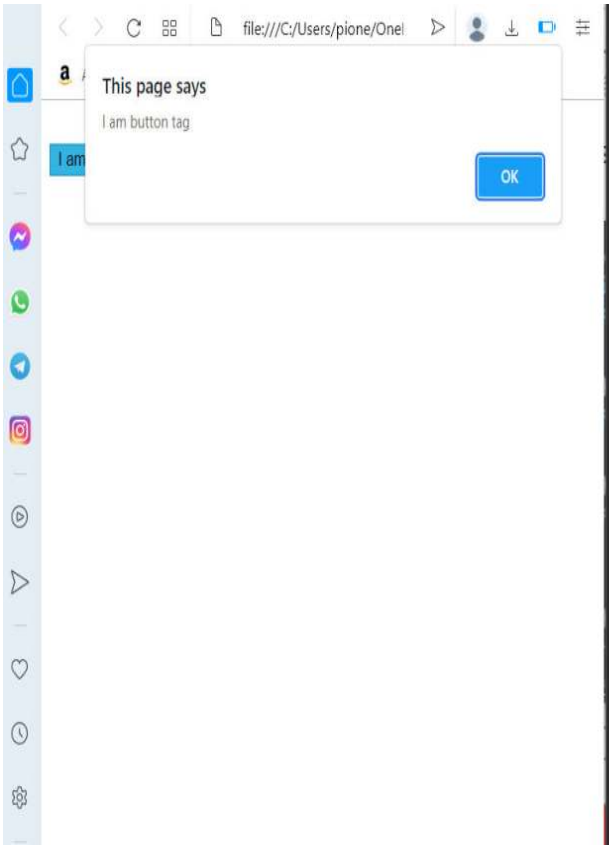
Here, when we click on the button tag onclick event " I am button tag" executes first and then onclick event of h2 tag " I am h2 tag" executes because both are under div and when we click on the button, the entire elements get clicked inside the div. The inner element event will call first and then outer element event will be called if two or more same events are applied in event bubbling. Event bubbling is a process in which inner element event is called and then outer and then outermost wherever event is applied. It is just like bubble that appears on the top of the glass from bottom.

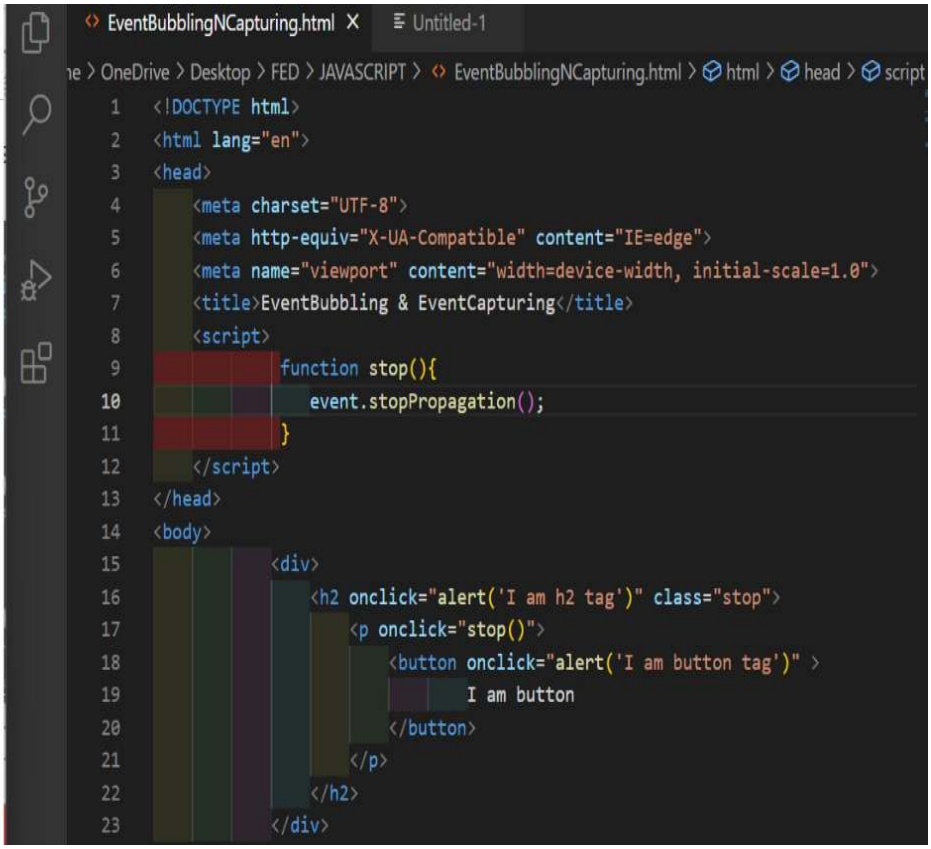
Note:- Event bubbling will be executed only if all the same events are applied on different elements.

We use it in a complex website wherein many event are used.

If we want that when we click on inner element event, it executes the inner element event function only then we need to use event.stopPropagation() as you can see below:-







Event Capturing:- It is just opposite to the event bubbling. It first call the outer element event and inner element event and so on.

63. What would be the output of the below JavaScript code?

```
var a = 10;
if (function abc(){})
{
a += typeof abc;
}
console.log(a);
```

The output of this JavaScript code will be 10undefined. The if condition statement in the code evaluates using eval. Hence, eval(function abc(){}) will return function abc(){}.

Inside the if statement, executing typeof abc returns undefined because the if statement code executes at run time while the statement inside the if the condition is being evaluated.

64. Can you write a JavaScript code for adding new elements in a dynamic manner?

18. What will be the output of the following code?

```
var Bar = Function Foo()
{
return 11;
};
typeof Foo();
```

The output would be a reference error since a function definition can only have a single reference variable as its name.

19. What will be the output of the following code?

```
var Student = {
college: "abc",
};
var stud1 = Object.create(Student);
delete stud1.college;
console.log(stud1.company);
```

This is essentially a simple example of object-oriented programming. Therefore, the output will be 'abc' as we are accessing the property of the student object.

20. How do you remove duplicates from a JavaScript array?

There are two ways in which we can remove duplicates from a JavaScript array:

By Using the Filter Method

To call the filter() method, three arguments are required. These are namely array, current element, and index of the current element.

By Using the For Loop

An empty array is used for storing all the repeating elements.

21. Can you draw a simple JavaScript DOM (Document Object Model)?

22. Logical operators

Logical operators in javascript, unlike operators in other programming languages, **do not return true or false. They always return one of the operands.**

OR (| |) operator - If the first value is truthy, then the first value is returned. Otherwise, always the second value gets returned.

AND (&&) operator - If both the values are truthy, always the second value is returned. If the first value is falsy then the first value is returned or if the second value is falsy then the second value is returned.

23. What is NaN property in JavaScript?

NaN property represents the **"Not-a-Number"** value. It indicates a value that is not a legal number.

typeof of NaN will return a **Number**.

To check if a value is NaN, we use the **isNaN()** function,

Note- isNaN() function converts the given value to a Number type, and then equates to NaN.

```
isNaN("Hello")  // Returns true
isNaN(345)   // Returns false
isNaN('1')  // Returns false, since '1' is converted to Number type which results in 0 ( a number) isNaN(true)
// Returns false, since true converted to Number type results in 1 ( a number)
isNaN(false) // Returns false
isNaN(undefined) // Returns true
```

24. Define Mobile First Approach

23. What has to be done in order to put Lexical Scoping into practice?

To support lexical scoping, a JavaScript function object's internal state must include not just the function's code but also a reference to the current scope chain.

24. In JavaScript, how do you turn an Object into an Array []?

```
<script>
     let obj = { id: "1", name: "user22", age: "26", work: "programmer" };
     console.log(Object.keys(obj)); // [ "id", "name", "age", "work" ]
     console.log(Object.values(obj)); //  [ "1", "user22", "26", "programmer" ]
     console.log(Object.entries(obj));
     //Array(4) [ (2) […], (2) […], (2) […], (2) […] ]
          0: Array [ "id", "1" ]
          1: Array [ "name", "user22" ]
          2: Array [ "age", "26" ]
          3: Array [ "work", "programmer" ]
  </script>
```

25. What is the output of the following code?

```
const b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
for (let i = 0; i < 10; i++) {
setTimeout(() => console.log(b[i]), 1000);
} // 1..9
for (var i = 0; i < 10; i++) {
setTimeout(() => console.log(b[i]), 1000);
}
```

Ans.

```
12345678910undefinedundefinedundefinedundefinedundefinedundefinedundefinedundefinedundefinedundefined
```

26) Who developed JavaScript, and what was the first name of JavaScript?

JavaScript was developed by Brendan Eich, who was a Netscape programmer. Brendan Eich developed this new scripting language in just ten days in the year September 1995. At the time of its launch, JavaScript was initially called Mocha. After that, it was called Live Script and later known as JavaScript.

27) If we want to return the character from a specific index which method is used?

```
<script>
   let x = "I am Sarfraz";
  let y = x.charAt(3);
  console.log(y); // m
</script>
```

28) What is the use of window object?

The window object is created automatically by the browser that represents a window of a browser. It is not an object of JavaScript. It is a browser object.

The window object is used to display the popup dialog box. Let's see with description.

Method	Description
alert()	displays the alert box containing the message with ok button.
confirm()	displays the confirm dialog box containing the message with ok and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.
setTimeout()	performs the action after specified time like calling function, evaluating expressions.

29) What is the use of history object?

The history object of a browser can be used to switch to history pages such as back and forward from the current page or another page. There are three methods of history object.

1. history.back() - It loads the previous page.
2. history.forward() - It loads the next page.
3. history.go(number) - The number may be positive for forward, negative for backward. It loads the given page number.

More details.

30) How to write a comment in JavaScript?

There are two types of comments in JavaScript.

1. Single Line Comment: It is represented by // (double forward slash)
2. Multi-Line Comment: Slash represents it with asterisk symbol as /* write comment here

31) How to write HTML code dynamically using JavaScript

The innerHTML property is used to write the HTML code using JavaScript dynamically. Let's see a simple example:

```
<body>
```

```
   <div id="mylocation"></div>
   <script>
      document.getElementById('mylocation').innerHTML = "<h2>This is Sarfraz</h2>";
   </script>
</body>
```

32) How to write normal text code using JavaScript dynamically?

The innerText property is used to write the simple text using JavaScript dynamically. Let's see a simple example:

```
<body>
   <div id="mylocation"></div>
   <script>
      document.getElementById('mylocation').innerText = "This is Sarfraz";
   </script>
</body>
```

35) In which location cookies are stored on the hard disk?

The storage of cookies on the hard disk depends on the OS and the browser.

The Netscape Navigator on Windows uses a cookies.txt file that contains all the cookies. The path is c:\Program Files\Netscape\Users\username\cookies.txt

The Internet Explorer stores the cookies on a file username@website.txt. The path is: c:\Windows\Cookies\username@Website.txt.

36) What's the difference between event.preventDefault() and event.stopPropagation() methods in JavaScript?

In JavaScript, the event.preventDefault() method is used to prevent the default behavior of an element.

For example: If you use it in a form element, it prevents it from submitting. If used in an anchor element, it prevents it from navigating. If used in a contextmenu, it prevents it from showing or displaying.

On the other hand, the event.stopPropagation() method is used to stop the propagation of an event or stop the event from occurring in the bubbling or capturing phase.

39) What is the real name of JavaScript?

The original name was **Mocha**, a name chosen by Marc Andreessen, founder of Netscape. In September of 1995, the name was changed to LiveScript. In December 1995, after receiving a trademark license from Sun, the name JavaScript was adopted.

40) How can you check if the event.preventDefault() method was used in an element?

When we use the event.defaultPrevent() method in the event object returns a Boolean indicating that the

event.preventDefault() was called in a particular element.

41) How to set the cursor to wait in JavaScript?

The cursor can be set to wait in JavaScript by using the property "cursor". The following example illustrates the usage:

```
   <script>
     let x = window.document.body.style.cursor = "wait";
     document.write(x);
     // When keep the cursor on wait on the document, we will se a moving round circle which
     //means it is waiting.
   </script>
```

42) What is negative infinity?

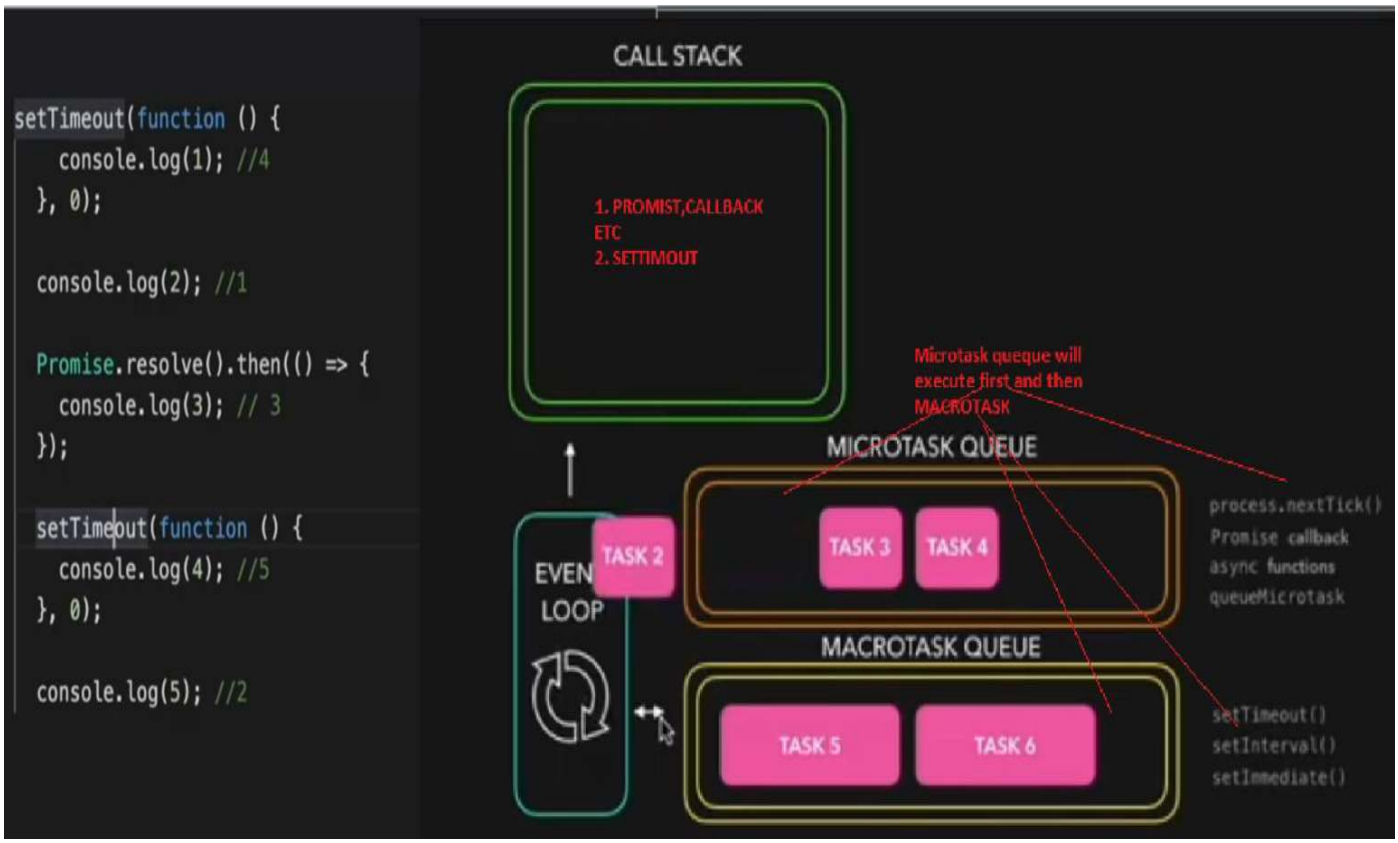
Negative Infinity is a number in JavaScript which can be derived by dividing the negative number by zero. For example:

```
<script>
   let a = 10;
   console.log(a/0); //Infinity
</script>
```

43) What is the difference between View state and Session state?

"View state" is specific to a page in a session whereas "Session state" is specific to a user or browser that can be accessed across all pages in the web application.

43) GUESS THE OUTPUT



44) What are the pop-up boxes available in JavaScript?

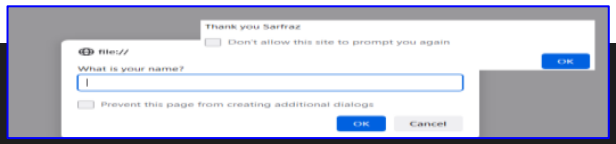
Alert Box:- It is used to display different messages. An alert will pop up on the web browser. It is window property.

Confirm Box:- It is also a window property and just like alert but it also ask question “ok” means yes cancel means no

We usually use it when you want to display something when customer leaving the web page.

```
<script>
   let a = confirm("Are you happy?");
   if(a==true){
      alert("Yes, I am sure");
   }else{
      alert("No, I am not sure");
   }
</script>
```

Prompt Box:- We use it to get answer of any questions that you want to ask from customers.It creates input box.




```
<script>
   let a = prompt("What is your name?");
   alert(`Thank you ${a}`)
</script>
```

45) How can we detect OS of the client machine using JavaScript?

```
<script>
      let a = navigator.appVersion;
      let b = navigator.userAgent;
      console.log(a);  // 5.0 (Windows)
      console.log(b);  //Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:107.0) Gecko/20100101 Firefox/107.0
   </script>
```

46) How to submit a form using JavaScript by clicking a link?

```
   <form action="" id="form1" runat="server">
      <input type="text" placeholder="enter the name">
      <a href="" id="linkID">Submit The Form</a>
      </div>
   </form>
   <script>
      let form = document.getElementById("form1");
      let link = document.getElementById("linkID");
      link.addEventListener("click", function () {
         form.submit();
      });
   </script>
```

47) Is JavaScript faster than ASP script?

Yes, because it doesn't require web server's support for execution.

48) How to change the background color of HTML document using JavaScript?

```
 <script>
      document.body.style.backgroundColor="blue";
   </script>
```

49) How to handle exceptions in JavaScript?

By the help of try/catch block, we can handle exceptions in JavaScript. JavaScript supports try, catch, finally and throw keywords for exception handling.

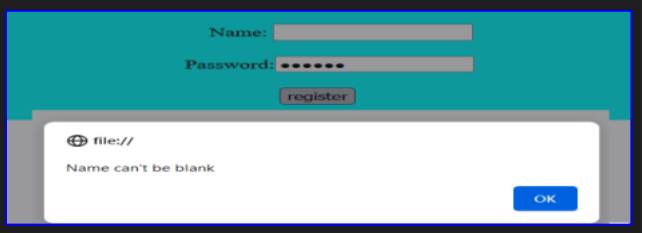
50) How to validate a form in JavaScript?

In form, we have two types of validation. First is server side and second is client side validation

Server side validation is performed at the server by server side scripting languages but client side validation is done by JavaScript. Whenever we click on submit button the input data taken by users sent to the server. If anything is wrong in the data, server will have to get back to you and then need to correct that data and resend to the server. So, before sending the data to server on button click we validate Or check the data whether customer has filled the correct data or not using JavaScript which is client Side because JavaScript run on the browser. So, first we validate through JavaScript and then send To the server and then validated at the server.

```
<style>
      form{
         text-align:center;
         background-color: aqua;
         padding: 20px;
      }
      #stl{
         margin-top: 20px;
         margin-right:17px;
      }
      #stl1 {
         margin-top: 20px;
         margin-right:35px;
      }
   </style>
</head>
<body>
```

```
   <form name="myform" method="post" action="abc.jsp" onsubmit="return validateform()">
// We have taken return  because when we click on submit button it will not submit if return false
   but it can be submitted if return true in the function as shown below. We don't want to submit the form because we will check
   Condition first or validate first and if we don't take return here and under function return false, it will directly sent to
   The server without checking the validation//
      Name: <input type="text" name="name"><br />
      Password:<input type="password" name="password" id="stl"> <br />
               <input type="submit" value="register" id="stl1">
   </form>
   <script>
      function validateform() {
         var name = document.myform.name.value;
         var password = document.myform.password.value;
         if (name == null || name == "") {
            alert("Name can't be blank");
            return false;
         } else if (password.length < 6) {
            alert("Password must be at least 6 characters long.");
            return false;
         }
      }
   </script>
</body>
```




```
 <form onsubmit="return validation()">
        <h2>Login</h2>
        <input type="text" placeholder="Enter the User Name" id="username"><br>
        <input type="password" placeholder="Enter the Password" id="password"><br>
        <input type="submit" value="Submit" id="submit">
     </form>
     <script>
              function validation(){
                 return false;    // form will not be submitted
                 return true;    // form will   be submitted
                        }
     </script>
```

```
<body>
   <div id="form-container">
      <form onsubmit="return validation()">
         <h2>Login</h2>
         <input type="text" placeholder="Enter the User Name" id="username"><br>
         <p id="userError" class="erro"></p>
         <!-- to show the error of username -->
         <input type="password" placeholder="Enter the Password" id="password"><br>
         <!-- to show the error of username -->
         <p id="passError" class="erro"></p>
         <input type="submit" value="Submit" id="submit">
      </form>
      <script>
         username = document.getElementById("username");
         password = document.getElementById("password");
         let flag = 1;  //flag=1 means true or passed the contions
         function validation() {
            if (username.value == "") {
               // console.log("UserName is empty");
               document.getElementById('userError').innerHTML = "User Name Is Empty"
               // alert("User Name Is Empty");
               flag = 0;
            } else if (username.value.length < 3) {
               alert("Enter atleast 4 charactors");
               document.getElementById('userError').innerHTML = "Enter atleast 4 charactors";
               flag = 0;
            } else {
               document.getElementById('userError').innerHTML = "";
               //username ke creteri meet hone pe error na dikhe
               flag = 1;
            }
```

```
         if (password.value == "") {
            document.getElementById('passError').innerHTML = "Password Is Empty";
            //will show no error if meet the password creteria
            flag = 0;
         } else {
            document.getElementById('passError').innerHTML = "";
            flag = 1;
         }
         // return false;  // It will not submit the form even all the conditions are met so now
         // we have to apply condition that if all the conditions are fullfilled then submit the Form
         // else not and for that we will take take a variable flag above before function declaration
         // and if(flag) means passed the condition or flag=1=true. So now we will set flag=0 for error
         // so we need to set flag=1 in else part
        if(flag) {
            return true; // will submit as contion is true
         } else {
            return false;   // will not submit as contion is false
         }
      }
   </script>
```

63. How to validate email in JavaScript?

```
<form>
   <label for="">User Name</label>
   <input type="email" onkeyup="validate()" name="" id="user" value="" placeholder="Enter Your Email">
</form>
<script>
   function validate() {
      var user = document.getElementById("user").value;
      console.log(user);// pioneer25me@gmail.com
      var userstyle= document.getElementById("user")
      console.log(userstyle); // entire form
      let userexp= /^[A-Za-z_.0-9]{6,30}@[A-Za-z]{4,12}.[A-Za-z.]{2,8}$/;
    //  pioneer25me@gmail.com// first one is to validate upper and lower {minum, maximum length}
    //https://regex101.com/ used
      if(userexp.test(user)){
         userstyle.style.outline="2px solid green";
         userstyle.style.border="none";
      }else{
         userstyle.style.outline="2px solid red";
         userstyle.style.border="none";
      }
   }
</script>
```

64. How to validate number

https://www.youtube.com/watch?v=RAKhi0pQaKU

Sign up page validation

https://www.youtube.com/watch?v=Gku9iMSMbWg

To go to website after successful login

https://www.youtube.com/watch?v=SOLHcE3bnOU

https://www.youtube.com/watch?v=XjpIPXDTWF4

64. What is the use of a Date object in JavaScript?

The JavaScript date object can be used to get a year, month and day. You can display a timer on the web page by the help of JavaScript date object.

```
<script>
            var a = new Date();
            document.write(a.toDateString() + "<br>");// To print day name month name today's data and today's year
            document.write(a.getDate() + "<br>");// Today's date
            document.write(a.getFullYear() + "<br>");//Today's year
            document.write(a.getMonth() + "<br>");//Today's month 0 means jan
            document.write(a.getDay() + "<br>");//Today's daya 0 means sunday
            document.write(a.getHours() + "<br>");//current hours of your system
            document.write(a.getMinutes() + "<br>"); //current minutes of your system
            document.write(a.getSeconds() + "<br>");//current seconds of your system
            document.write(a.getMilliseconds() + "<br>"); //current milliseconds 1000 ms=60sec=1minute of your system

             We can also check any history date, month, day, year by putting any previous date under new Date like var a=new
             Date('January 10 2010')

            var a = new Date('January 7 2010');
            document.write(a.getDate() + "<br>");// aaj ke din january 10, 2010
            document.write(a.getFullYear() + "<br>");
            document.write(a.getMonth() + "<br>");
            document.write("Day was ", a.getDay() + "<br>")
            //past mein date dalne par sirf day change hota hai jaise aaj ke din january 30 2010 ko koun sa din tha hours 0
             dikhayega
            document.write(a.getHours() + "<br>");
            document.write("<br><br><br><br><br>")
              SET METHOD
            var a = new Date();
            a.setDate(10);  // Means iss mahine ke 10 tarik ko koun sa din tha (Thu Nov 10 2022 17:13:57 GMT+0530 (India
            Standard Time)
            document.write(a + "<br>");//It gives this month's 10th june day like friday. iss mahine ke 10 tarik ko kya din tha.
            a.setMonth(0);  // Is saal ke january month me kouan sa din tha // Mon Jan 10 2022 17:16:50 GMT+0530 (India Standard
             Time)month
            document.write(a + "month" + "<br>");
            a.setFullYear(2020);
            document.write(a + "<br>");
            a.setHours(3);
            document.write(a + "<br>");
            document.write("<br><br><br><br><br>")
            document.write("<mark> How to print today's day,date,month,year and any other you want</mark>" + "<br>")
            var a = new Date();
            document.write(a.getDate() + " / " + a.getMonth() + "/ " + a.getFullYear() + "<br>");
            var a = new Date('January 10 2010');
            document.write(a.getDate() + " / " + a.getMonth() + "/ " + a.getFullYear() + "<br>");
  </script>
```

65. What is the use of a Boolean object in JavaScript?

The JavaScript Boolean is an object that represents value in two states: true or false. You can create the JavaScript Boolean object by Boolean() constructor.

```
<script>
    function display() {
       document.writeln(10 < 20);//true
       document.writeln(10 < 5);//false
    }
    display();
  </script>
```

66. What is the use of a Typed Array object in JavaScript?

Introduced in ES6 . It is good to use when working with binary data.

The JavaScript TypedArray object illustrates an array like a view of an underlying binary data buffer. There is any number of different global properties, whose values are TypedArray constructors for specific element types.

```
<script>
      function display()
      {
         var arr1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
         arr1.copyWithin(2);  //1,2,1,2,3,4,5,6,7,8
         document.write(arr1);
      }
      display();
   </script>
```

67. What are the falsy values in JavaScript, and how can we check if a value is falsy?

Those values which become false while converting to Boolean are called falsy values.

68. Define pure function and impure function

Pure functions return the same output if we use the same input parameters. In another word, pure function is a function that takes an input and return the same as an output and that output is predictable. Here, in the given function we know that when we add 10+1 we will get 11 or 1 **However, impure functions give different outcomes when we pass the same arguments multiple times**. Filter and string are pure functions because returns the same output.

Any function can be a pure function that gives/ returns same output for the same input and has no side effect(a function that does not modify the variable whenever we run that function)

Pure function is never dependent of global variables because they can be changed.

Pure function does not mutate or change the input.

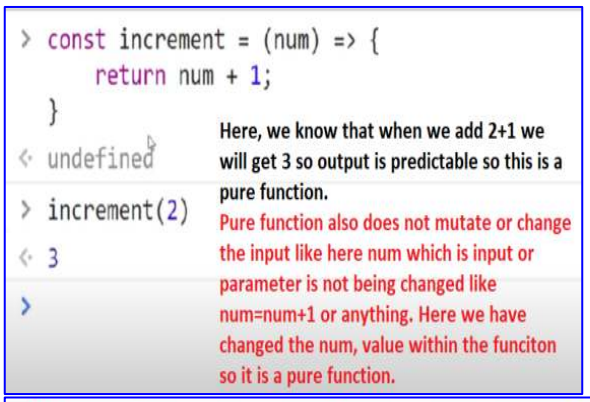
FEATURES	PURE FUNCTION	IMPURE FUNCTION
Input and output	Same output for same input	Different output for same input
Side effects	x	✓
Return value	✓	x
Behavior of the state	No change	State of the program, system will change
Testing accuracy	Easy for testing	Testing is difficult due to side effects

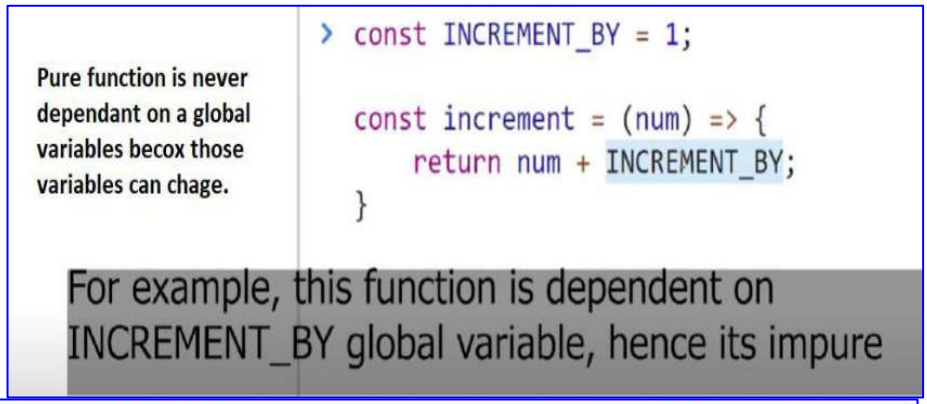
```
/*
Pure functions are functions that accept an input and returns a value without modifying any
data outside its scope(Side Effects).
predictable and without side-effects
side effect - dont modify any variable */


var x = 10;
function add(){
    console.log(x);
    x++;
}

add();   10
add();   11
add();   12
add();   13
```

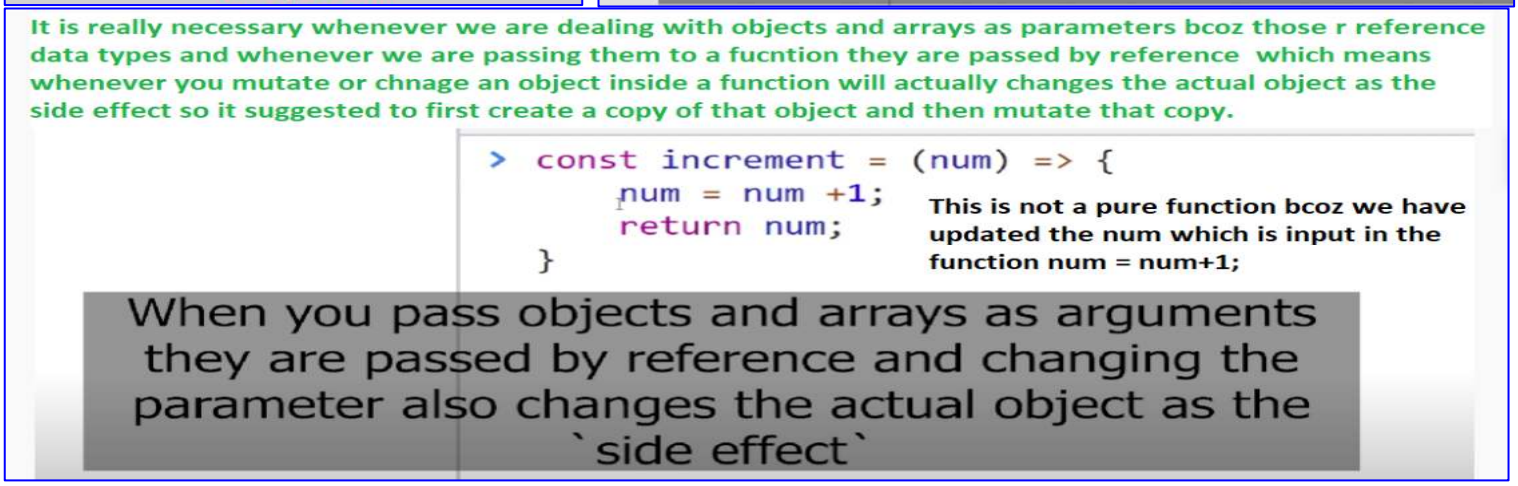
Here, variable x which is outside of the function called outside scope and it has value 10 and everytime we call it it will return different value so value is changing. Hence state is changing so it has side effect and it is impure function.

The below one is the pure function because here output is predictable.

Some more example of pure functions







. const falsyValues = ['', 0, null, undefined, NaN, false];

We can check if a value is falsy by using the Boolean function or the Double NOT operator (!!).

1.Which of the following JavaScript code snippets will provide the desired result?

true runtime error compilation error false

3.In JavaScript, which of the following is not a bug?

Make a zero division is not a bug it simply returns infinity or negative infinity.

Semicolons are missing

A grammatical mistake

Bracket is missing

5.What do you mean by block statement in JavaScript?

Block-based on condition

6.Argument class is Even in a dynamic state, objects inherit prototype characteristics.

10.The interpreter for JavaScript is Client

1) Which of the following statement is true?

If onKeyDown returns false, the key-press event is canceled.

2) Which of the following statement shows the primary difference between JavaScript and Java?

We know that Java is an OOP language while JavaScript is an OOP scripting language. The most fundamental difference between JavaScript and Java is that the functions are values, and there is no such distinction between methods and fields.

3) Which of the following syntax is valid for creating a RegExp object?

1. var txt=new RegExp(pattern,attributes);

2. var txt=/pattern/attributes;

4) Which of the following statement specifies the meaning of Augmenting classes?

"Objects inherit prototype properties even in a dynamic state" is the correct answer.

5) If para1 is the DOM object for a paragraph, which of the following would be the correct syntax to change the text within the paragraph?

"para1.value="New Text";" is the correct syntax to change the text within the paragraph.

```
<body>
   <p>Hello, I am loving India</p>
   <script>
         let b=document.querySelector("p").value="Hi";
         document.write(b);
   </script>
</body>
```

6) Which of the following is used to define the behavior of the instances present of a class inside a method?

Classes

Answer: B: "Classes" is the correct answer. The class's behavior is defined by the class and is shared by all instances.

7) Which of the following statement is true in the case of the property of the JSON() method?

A JSON() method is invoked automatically by the JSON.stringify() method

Hide Answer Workspace

8) Which of the following statement is true if class B can extend another class A? (B ja sakta hai A mein yani A baap hai)

A is the superclass, and B is the subclass.

9) Which of the following is the correct syntax of the eval() function?

```
   <script>
            eval("alert('I am loving it')"); // alert was taken as string inside eval
            function abc(a,b){
              return a+b
            }
            eval("document.write(abc(10, 20))")  // Even the function is being called inside eval fuction
            let sum = '1 + 2';
            console.log(sum);  // 1+2  as it is string
            console.log(eval(sum)); // 3 but here even string was considered as sum to add
   </script>
```

[objectName].eval(string)

The eval() function is a global fuction. It is suggested not to use much as it slow the application. It takes any JavaScript code as string and execute the result. We can call any function or use alert which are codes of javascript and they will be executed.

10) What would be the output of the following JavaScript code?

```
   <script>
      const obj1 =
      {
         a: 10,
         b: 15,
         c: 18
      };
      const obj2 = Object.assign({ c: 11, d: 12 }, obj1);
```

```
    console.log(obj2);   // Object { c: 18, d: 12, a: 10, b: 15 } // c:11 was replaced to c:18
    // Object.assign(object jise paste karna ahi, jis object me paste karna hai) is used to copy or merge the one object property and valuees to another object
    console.log(obj2.c, obj2.d); // 18  12
  </script>
```

HISTORY IN DETAILS

ES8: ES8 or ECMAScript 2017 was released in the year 2017. This version allows new methods of coding with JavaScript.

PADSTART(): This method pads a string with another string at the beginning.

```
let str='a3';
console.log(str.padStart(3,'#'));       //#a3
```

PADEND():This method pads a string with another string and makes the resulting string reach a given length. It adds spaces at the end of the string.

```
let str = 'Bat';
console.log(str.padEnd(6, '.'));        //Bat…
```

OBJECT.ENTRIES():It returns an array that contains the key-value pairs of a given object as an array.

```
const colors= { BL: 'Blue',    OR: 'Orange',    YE: 'Yellow',    PI: 'Pink'};
    Object.entries(colors);    // [['BL', 'Blue'], ['OR', 'Orange'], ['YE', 'Yellow'], ['PI','Pink']]
```

TRAILING COMMAS: A trailing comma is simply a comma that comes at the end of the last item in a list.

```
getDetails(name, age, gender,) { ... }
```

SHARED MEMORY AND ATOMICS:

The same data can be read and written on multiple threads using the SharedArrayBuffer constructor. Interruption during the process of reading or writing can be avoided by using Atomic objects. This allows the previous operation to finish prior to the next one.

OBJECT.GETOWNPROPERTYDESCRIPTORS(): An object is returned to the own property descriptors with get, set, writeable, configurable and enumerable attributes.

```
const person = { name: 'Tay' };
let propertyDescriptor = Object.getOwnPropertyDescriptor(person, 'name');
console.log(propertyDescriptor);        // {"value":"Tay","writable":true,"enumerable":true,"configurable":true}
```

OBJECT.VALUES(): It returns an array of a given object' s own enumerable property values.

```
const colors= { BL: 'Blue',    OR: 'Orange',    YE: 'Yellow',    PI: 'Pink'};
    Object.values(colors);  // ['Blue', 'Orange', 'Yellow', 'Pink']
```

ES9: ES9 or ECMAScript 2018 is the latest update and was released in the year 2018.

ASYNCHRONOUS ITERATION:

An async iterable object can be used as a loop iteration with the help of for-await-of.

```
for await (const line of readLines(filePath)) {
  console.log(line)
}
```

PROGRAMMING QUESTIONS

How to change HTML content using JavaScript?

```
   <h1>Hello</h1>
   <script>
            document.querySelector("h1").innerHTML = "Hello, Sarfraz";
   </script>
```

How to change HTML style(css)?

```
   <h1 style="color:blue">Hello</h1>
   <script>
            document.querySelector("h1").style.color = "red";
   </script>
```

How to hide and show HTML element?

```
<body>
   <button onclick="show()">Display The Content </button>
   <button onclick="hide()">Hide The Content </button>
   <h1 style="display:none ;">Hello</h1>
   <script>
            function hide(){
            document.querySelector("h1").style.display = "none";
            }
            function show(){
            document.querySelector("h1").style.display = "block";
            }
   </script>
</body>
```

How to create string?

```
<body>
       <div></div>
   <script>

         let  y="kjfadkjk"
           let x =  document.getElementsByTagName("div").innerHTML = y;
           document.write(`${x} <br>`);
   </script>
</body>
```

Whether number is prime or not,

```
<body>
   <h1>Whether Number is Prime or Not</h1>
   <script>
         let num = 10;  // jo sir apne se aur 1 se katta hai prime number hat 1, 2, 3,5, 7 , 11, 13,17 etc.
         var isPrimeNumber = true;
         for (let i = 2; i<num; i++) {
            if (num % i == 0) {
               isPrimeNumber = false;
            }
         }
         if (isPrimeNumber == true) {
            document.write(`${num} is a Prime Number`);
         } else {
            document.write(`${num} is not a Prime Number or Composite Number`)

         }
   </script>
</body>
```

Whether number is odd or even

```
   <h1>Whether Number is Odd or Even</h1>
   <script>
         let num = 103;  // jo 2 se divide ho jaye wo even number hai 2, 4, 6, 8, 10,  etc. baki odd hai

         if(num%2==0){
```

```
            document.write(`${num} is even number`);
        }else{
            document.write(`${num} is odd number`);
        }
    </script>
```

Sum of the digits.

```
    <script>
        let num = 346;  // jo 2 se divide ho jaye wo even number hai 2, 4, 6, 8, 10,  etc. baki odd hai
        let x = num%10;
        document.write( x +"<br>");  // 6
        document.write(parseInt(num/10) +"<br>");  //34
        let y = 34%10;
        document.write(y +"<br>");  // 4
        let z = parseInt(34/10);
        document.write(z +"<br>");  // 3

        document.write(x+y+z);
    </script>
```

```
<body>
    <h1>Some Of The Digits</h1>
    <script>
        let number = 346;
        var str = number.toString();
        let sum =0;
        for(var i =0; i<str.length; i++){
            sum= sum+ parseInt(str.charAt(i));  //parseInt to convert the decimal or point to integer or non point
        }
        document.write(sum);
    </script>
</body>
```

Greatest and Smallest Number

```
<script>
    let array = [12,45,36,12,20];

   let x =  array.sort(function(a,b){
       return b-a
    });
  document.write(x[0]);  // The largest number  45
  document.write(x[1]);  // The second largest number  36
  </script>
```

```
 <script>
    let array = [12,45,36,13,20];

   let x =  array.sort(function(a,b){
       return a-b
    });
  document.write(x[0]);  // The lowest number  12
  document.write(x[1]);  // The second lowest number  13
  </script>
```

72. What is the purpose of the following JavaScript code?

```
var scope = "global scope";
function check() {
var scope = "local scope";
```

```
function f()
{return scope}
return f;
}
```

Every executing function, code block, and script as a whole in JavaScript has a related object known as the Lexical Environment. The preceding code line returns the value in scope.

JavaScript Coding Interview Questions

73. Guess the outputs of the following codes:

```
// Code 1:
function func1(){
setTimeout(()=>{
console.log(x);
console.log(y);
},3000);

var x = 2;
let y = 12;
}
func1();
// Code 2:
function func2(){
for(var i = 0; i < 3; i++){
setTimeout(()=> console.log(i),2000);
}
}
func2();
// Code 3:

(function(){
setTimeout(()=> console.log(1),2000);
console.log(2);
setTimeout(()=> console.log(3),0);
console.log(4);
})();
```

Answers:

Code 1 - Outputs **2** and **12**. Since, even though **let** variables are not hoisted, due to the async nature of javascript, the complete function code runs before the setTimeout function. Therefore, it has access to both x and y.

Code 2 - Outputs **3**, three times since variable declared with **var** keyword does not have block scope. Also, inside the for loop, the variable i is incremented first and then checked.

Code 3 - Output in the following order:

```
2431 // After two seconds
```

Even though the second timeout function has a waiting time of zero seconds, the javascript engine always evaluates the setTimeout function using the Web API, and therefore, the complete function executes before the setTimeout function can execute.

75. Guess the outputs of the following code:

```
// Code 1:
let x= {}, y = {name:"Ronny"},z = {name:"John"};
x[y] = {name:"Vivek"};
x[z] = {name:"Akki"};console.log(x[y]);
// Code 2:
function runFunc(){
console.log("1" + 1);
console.log("A" - 1);
console.log(2 + "-2" + "2");
console.log("Hello" - "World" + 78);
console.log("Hello"+ "78");
}
runFunc();
// Code 3:
let a = 0;let b = false;console.log((a == b));console.log((a === b));
```

Answers:

Code 1 - Output will be **{name: "Akki"}.**

Adding objects as properties of another object should be done carefully.

Writing **x[y] = {name:"Vivek"}** , is same as writing **x['object Object'] = {name:"Vivek"}** ,

While setting a property of an object, **javascript coerces the parameter into a string.**

Therefore, since **y** is an object, it will be converted to **'object Object'.**

Both x[y] and x[z] are referencing the same property.

Code 2 - Outputs in the following order:

```
11
Nan2-22NaN
Hello78
```

Code 3 - Output in the following order due to equality coercion:

```
truefalse
```

76. Guess the output of the following code:

```
var x = 23;

(function(){
var x = 43;
(function random(){
x++;
console.log(x);
```

```
var x = 21;
})();
})();
```

Answer:

Output is **NaN**.

random() function has functional scope since x is declared and hoisted in the functional scope.

Rewriting the random function will give a better idea about the output:

```
function random(){
var x; // x is hoisted
x++; // x is not a number since it is not initialized yet
console.log(x); // Outputs NaN
x = 21; // Initialization of x
}
```

77. Guess the outputs of the following code:

```
// Code 1

let hero = {
powerLevel: 99,
getPower(){
return this.powerLevel;
}
}

let getPower = hero.getPower;

let hero2 = {powerLevel:42};
console.log(getPower());
console.log(getPower.apply(hero2));



// Code 2

const a = function(){
console.log(this);

const b = {
func1: function(){
console.log(this);
}
}
```

```
const c = {
func2: ()=>{
console.log(this);
}
}

b.func1();
c.func2();
}

a();



// Code 3

const b = {
name:"Vivek",
f: function(){
var self = this;
console.log(this.name);
(function(){
console.log(this.name);
console.log(self.name);
})();
}
}
b.f();
```

Answers:

Code 1 - Output in the following order:

```
undefined
42
```

Reason - The first output is **undefined** since when the function is invoked, it is invoked referencing the global object:

```
window.getPower() = getPower();
```

Code 2 - Outputs in the following order:

```
global/window object
object "b"
global/window object
```

Since we are using the arrow function inside **func2, this** keyword refers to the global object.

Code 3 - Outputs in the following order:

```
"Vivek"
undefined
"Vivek"
```

Only in the IIFE inside the function **f, this** keyword refers to the global/window object.

78. Guess the outputs of the following code:

**Note - Code 2 and Code 3 require you to modify the code, instead of guessing the output.

```
// Code 1


(function(a){
return (function(){
console.log(a);
a = 23;
})()
})(45);
// Code 2
// Each time bigFunc is called, an array of size 700 is being created,// Modify the code so that we don't create the same array again and again
function bigFunc(element){
let newArray = new Array(700).fill('♥');
return newArray[element];
}
console.log(bigFunc(599)); // Array is createdconsole.log(bigFunc(670)); // Array is created again
// Code 3
// The following code outputs 2 and 2 after waiting for one second// Modify the code to output 0 and 1 after one second.
function randomFunc(){
for(var i = 0; i < 2; i++){
setTimeout(()=> console.log(i),1000);
}
}
randomFunc();
```

Answers -

Code 1 - Outputs **45**.

Even though a is defined in the outer function, due to closure the inner functions have access to it.

Code 2 - This code can be modified by using closures,

```
function bigFunc(){
let newArray = new Array(700).fill('♥');
```

```
return (element) => newArray[element];
}
let getElement = bigFunc(); // Array is created only once
getElement(599);
getElement(670);
```

Code 3 - Can be modified in two ways:

Using **let** keyword:

```
function randomFunc(){
for(let i = 0; i < 2; i++){
setTimeout(()=> console.log(i),1000);
}
}
randomFunc();
```

Using closure:

```
function randomFunc(){
for(var i = 0; i < 2; i++){
(function(i){
setTimeout(()=>console.log(i),1000);
})(i);
}
}
randomFunc();
```

79. Write a function that performs binary search on a sorted array.

```
function binarySearch(arr,value,startPos,endPos){
if(startPos > endPos) return -1;

let middleIndex = Math.floor(startPos+endPos)/2;

if(arr[middleIndex] === value) return middleIndex;

elsif(arr[middleIndex > value]){
return binarySearch(arr,value,startPos,middleIndex-1);
}
else{
return binarySearch(arr,value,middleIndex+1,endPos);
}
}
```

80. Implement a function that returns an updated array with r right rotations on an array of integers a .

Example:

Given the following array: **[2,3,4,5,7]**
Perform **3** right rotations:
First rotation : [7,2,3,4,5] , Second rotation : [5,7,2,3,4] and, Third rotation: [4,5,7,2,3]

return **[4,5,7,2,3]**

Answer:

```
function rotateRight(arr,rotations){
if(rotations == 0) return arr;
for(let i = 0; i < rotations;i++){
let element = arr.pop();
arr.unshift(element);
}
return arr;
}
rotateRight([2, 3, 4, 5, 7], 3); // Return [4,5,7,2,3]
rotateRight([44, 1, 22, 111], 5); // Returns [111,44,1,22]
```

81. Write the code for dynamically inserting new components.

```
<html>
<head>
<title>inserting new components dynamically</title>
<script type="text/javascript">
function addNode () { var newP = document. createElement("p");
var textNode = document.createTextNode(" This is other node");
newP.appendChild(textNode); document.getElementById("parent1").appendChild(newP); }
</script>
</head>
<body> <p id="parent1">firstP<p> </body>
</html>
```

82. Write the code given If two strings are anagrams of one another, then return true.

```
var firstWord = "Deepak";var secondWord = "Aman";


isAnagram(wordOne, wordTwo); // true
function isAnagram(one, two) {
//Change both words to lowercase for case insensitivity..
var a = one.toLowerCase();
var b = two.toLowerCase();

// Sort the strings, then combine the array to a string. Examine the outcomes.
a = a.split("").sort().join("");
b = b.split("").sort().join("");

return a === b;
}
```

83. Write the code to find the vowels

```
const findVowels = str => {
let count = 0
const vowels = ['a', 'e', 'i', 'o', 'u']
for(let char of str.toLowerCase()) {
if(vowels.includes(char)) {
count++
}
}
return count
}
```

1. What is jQuery?

101577889097
Gn/ggn/10640/622784

jQuery is a lightweight, browser independent(No need to to check the code on different browsers whether working or not) client-side, cross-platform JavaScript library. It is used to make more interactive, attractive and dynamic web pages. It helps to speed up the development process because it shortens the length of the code. It is based on write less and do more. It is very easy to manipulate the DOM, traverse the document, applying CSS, event handling, Ajax and animation.

1.1 Does jQuery work for both HTML and XML documents?

Ans:- No. jQuery works only for HTML documents.

1.2 Is jQuery a JavaScript or JSON library file?post

Ans:- jQuery is said to be a library of single JavaScript files which consists of DOM/CSS manipulations, event effects or animations, AJAX functions and various commonly used plugins.

1.1 What is CDN?

CDN stands for Content Delivery Network or Content Distribution Network. We know that we can use jQuery by downloading and the jQuery file and then use that in our project HTML file under script tag with 'src' attribute. But, when we use this process, our server will have to load 30kb file every time we run the code. So, instead of loading this 30kb on every run, we will use CDN path under before closing tag of body or head tag.

Suppose a visitor requested for something like a visitor tries to open the your website, then server will open that website and return HTML, CSS, IMAGES AND JQUERY IF USED as a website is a combination of these things. If we have multiple visitors who will try to access our website at the same time will increase load on the server because server is giving HTML, CSS , images and 30kb jQuery file too to all the visitors. So, to reduce the load of the server we provide 30kb jQuery file from different server called third party server from where our jQuery file is being received by the visitors. They are receiving HTML, CSS and images from one server which is our server and jQuery from another server called CDN server. It is called Content Delivery Network because your website content is coming from different networks. We can keep the images and CSS in different network too.

To use CDN path of jQuery, go to jQuery.com and copy CDN path and then use in your project.

1.2 What is the goal of CDN and what are the advantages of using CDN?

The primary goal of the CDN is to provide content to the end-users with high availability and high performance.

Advantages of using CDN:

- It reduces the load from the server.
- It saves bandwidth. jQuery framework is loaded faster from these CDN.

If a user regularly visits a site which is using jQuery framework from any of these CDN, it will be cached.

1.3 What are the two types of CDNs?

There are two types of CDN:

- **Microsoft:** It loads jQuery from AJAX CDN.
- **Google:** It loads jQuery from Google libraries API.
- Yahoo:-

2. Why do we use jQuery / What is jQuery used for/What are the advantages of jQuery??

- It is very easy to learn and use.
- It keeps the code simple, readable, clear and reusable.
- It also makes it easy to add animations and transitions to your pages. With the help of jQuery, you can perform many basic activities that would otherwise require numerous lines of JavaScript code by calling methods instead.
- It provides easy selectors just $("#sar) instead of document.getElementByid("")
- It is easy to handle complex animation because in Ajax we use long & length codes but here the same can be done in one line.
- CSS styling and DOM manipulation are very easy.
- It is browser independent.
- It improves the performance of an application.
- It is very fast and extensible and reduces the size of the file

3. What is the difference between JavaScript and jQuery?

JavaScript	jQuery
JavaScript is an interpreted language written in C and is a combination of ECMAScript and DOM	j Query is a JavaScript library developed to run things faster and make things simplified for JavaScript. jQuery doesn't have the ECMAScript.
JavaScript doesn't have cross-browser compatible functionality which is why a developer has to write code manually to implement the functionality.	Whereas the cross-browser code compatibility is inbuilt j Query and j Query is much smaller than other libraries.
JavaScript requires long lines of code to code	It writes less and does more and it does not have any dependencies on other libraries or frameworks. (Like babel).
JavaScript has lots of libraries and frameworks	j Query is not as heavy as other frameworks, It only uses the predefined JavaScript to make web apps interactive.

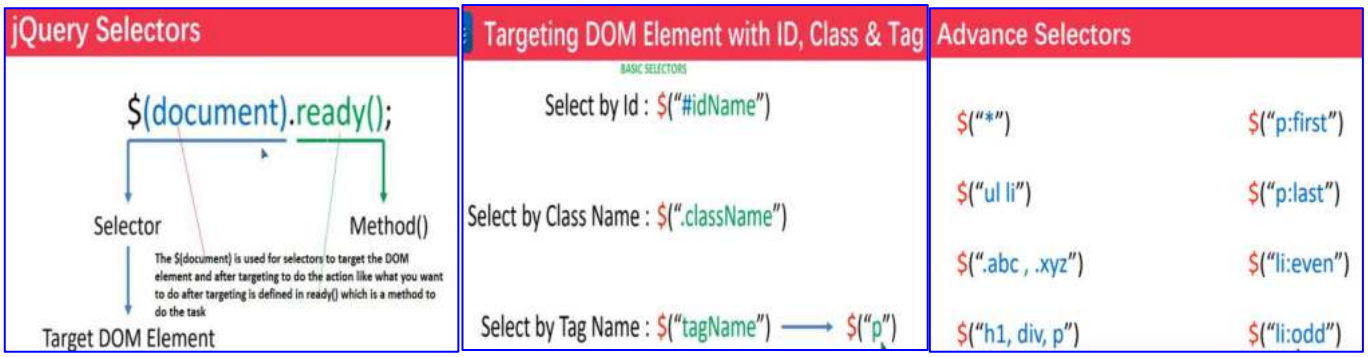
4. List some Features of jQuery.

DOM Manipulation:- It is very easy to select the DOM elements, traverse them and modify the content. It is very easy to target and use id, class, and tag.
Event Handling:- It helps in event detection and handling.
Ajax Support:- It is very easy to create a dynamic web-page using Ajax technology.
Animation:- It provides a plenty of built-I animation effects to use in the web-page.
Lightweight and Cross Browser:- supported by almost all the browsers.
Easy JSON parsing
It provides utilities such as feature detection and plug-ins for all kind of needs.

5. What are the selectors in jQuery? How many types of selectors in jQuery?

jQuery selectors are used to select the HTML elements and allows you to manipulate the HTML elements in a way we want. It selects the HTML elements on a variable parameter such as their name, classes, id, types, attributes, attribute values, etc. All selectors in jQuery are selected using a special sign i.e. dollar sign and parentheses:

- **Name or Element Selector:** It is used to select all elements which match with the given element Name. Example $("h1")
- **#ID:** It is used to select a single element which matches with the given ID
- **.Class:** It is used to select all elements which match with the given Class.
- **Universal (*):** It is used to select all elements available in a DOM.
- **Multiple Elements E, F, G:** It is used to selects the combined results of all the specified selectors E, F or G.
- **Attribute Selector:** It is used to select elements based on its attribute value.



$(".abc, .xyz") are two classes that we have targeted, $("h1, div, p") are three tags, $("p;first") used to target the first child of p tag, even will target the even one.

State some different types of jQuery Methods.

jQuery provides a variety of methods for doing various tasks, such as manipulating the DOM, events, and ajax. The table below covers many technique categories.

Essential Methods	Characterization	Classification
Before(), after(), append(), prepend(), appendTo(), prependTo(), empty(), remove(), clone() attr(),replaceWith(), replaceAll(),wrap(), wrapAll(), wrapInner() etc.	These methods alter DOM elements in some way, such as modifying attributes, style attributes, adding and deleting elements, and so on.	DOM Manipulation
addClass(), CSS(), hasClass(), removeClass(), toggleClass(), etc.	These methods retrieve and set CSS attributes of elements.	CSS
attr(), html(), HTMLremoveAttr(), prop(), val(), etc.	These methods retrieve and set element DOM properties.	Attributes
bind(), blur(), change(), click(), focus(), keyup(), keydown(), etc.	These methods deal with DOM or JavaScript events.	Events

Essential Methods	Characterization	Classification
		
animate(), fadeIn(), fadeOut(), fadeToggle(), hide(), show(),toggle(), stop(), slide down, slide up and slide toggle etc.	These techniques are used to provide elements of animation.	Effects
get(), getJson(), post(), load(), etc.	These techniques provide Ajax functionality with jQuery.	AJAX
children(), closest(), each(), first(), next(), filter(), **parent(), siblings(), etc.**	These methods assist in traversing from one DOM element to another in a parent-child hierarchy, such as locating ancestors, descendants, or sister elements of a specific element.	Traversing
blur(), change(), val(), submit(), etc.	Forms and their many components are handled by these methods and event handlers.	Forms
inArray(), isArray(), isFunction(), isNumeric(), isWindow(), isXmlDoc(), etc.	Utility methods are useful for obtaining information about various objects, such as a browser, function, array, or window.	Utilities
Width(), Height Methods, Position(), offset(), scrollTop(), scrollLeft()		

A. DOM Manipulation Method

Methods: append() to add content at last and prepend() at top inside the div , appendTo() & prependTo() are just like append and prepend method, they have just syntax difference as shown below before() method will be just above the div and after() method will be just below the div. Empty() method will remove the child element or content and remove() method will remove the entire div (parent and child inside that), clone() method is used to copy the object, element from any div or something and paste to some other div or anything

```
  <style>
    .sar{
      width: 200px;
      height: 200px;
      border: 1px solid red;}
  </style>
  <script src="jquery.js"></script>
</head>
<body>
  <script>
      $(document).ready(function(){
        $(".sar").append("<h2>Hello</h2>");
      });
  </script>
    <div class="sar">
      <p>All details</p>
    </div>
</body>
```

All details

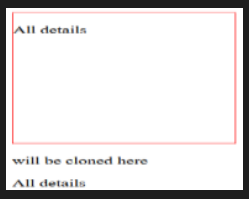
Hello

```
// On button click
```

```
 <script>
    $(document).ready(function () {
      $('button').click(function () {
        $(".sar").append("<h2>Hello</h2>");
      });
    });
  </script>
  <div class="sar">
    <p>All details</p>
  </div>
  <button>click</button>
</body>
```

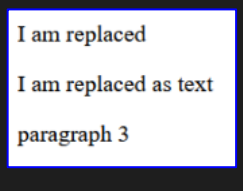
```
$(document).ready(function () {
      $('button').click(function () {
        $("<h2>Hello</h2>").appendTo(".sar");
      });
    });
```


```
                        // clone
<body>
  <script>
    $(document).ready(function(){
      $(".sar p").clone().appendTo(".sar1");  // sar class ke all details ko sar1 class ke nich copy kar diya
    });
  </script>
  <div class="sar">
    <p>All details</p>
  </div>
  <div class="sar1">
    <p>will be cloned here</p>
  </div>
  <button>click</button>
</body>
```

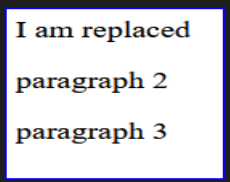


replaceWith() is used to replace any tag and text node to another tag and text node but replaceAll() will replace only one tag to another tag not text and the syntax

```
<body>
  <script>
    $(document).ready(function(){
      $(".sar1 p:first-child").replaceWith("<p>I am replaced</p>");  //
      $(".sar1 p:nth-child(2)").replaceWith("I am replaced as text");
    });
  </script>
  <div class="sar1">
    <p>paragraph 1</p>
    <p>paragraph 2</p>
    <p>paragraph 3</p>
  </div>
</body>
```




```
<body>
  <script>
    $(document).ready(function () {
      $("<p>I am replaced</p>").replaceAll(".sar1 p:first-child"); // fist content comes and then selector
      // $("I am replaced as text").replaceAll(".sar1 p:first-child"); // It will not print as taken as text
    });
  </script>
  <div class="sar1">
    <p>paragraph 1</p>
    <p>paragraph 2</p>
    <p>paragraph 3</p>
  </div>
</body>
```



wrap() method is used to create a parent element of any element or put or wrap any element inside any another element and unwrap method is used to remove the parent element.

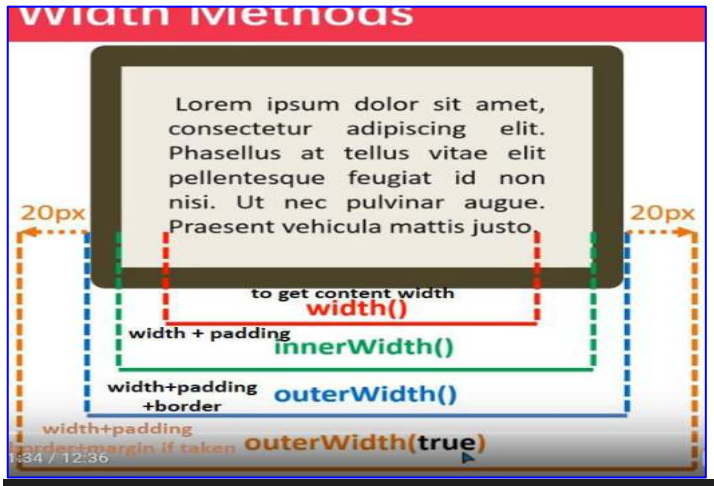
```
  <script>
```

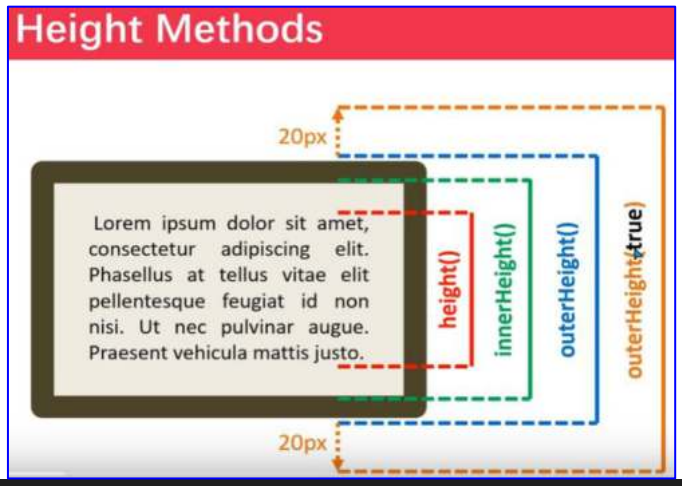
```
    $(document).ready(function(){
      $(".sar").wrap("<h1></h1>"); // hello was wrapped under h1 tag/h1 is parent of hello now.
    });

  </script>
  <div class="sar" style="width:100px; height:100px; border:1px solid red" ;>Hello</div>
</body>
```

WrapAll ()- to wrap more than one elements inside any element and wrapInner(() will wrap <I>Hello</I>

```
  <script>
    $(document).ready(function(){
      $(".sar, .sar1").wrapAll("<h1></h1>"); // both divs were wrapped under h1 tag
    });

  </script>
  <div class="sar" style="width:100px; height:100px; border:1px solid red" ;>Hello</div>
  <div class="sar1" style="width:100px; height:100px; border:1px solid red" ;>Hello1</div>
</body>
<script>
    $(document).ready(function () {
      $(".sar").wrapInner("<i></i>");
    });
  </script>
  <div class="sar" style="width:100px; height:100px; border:1px solid red" ;><span>Hello</span></div>
</body>
```

B. Width,Height Methods, Position(), offset(), scrollTop(), scrollLeft()






```
<body>
  <script>
    $(document).ready(function () {
      $('button').click(function () {
        console.log($(".sar").width());  // 100
        console.log($(".sar").innerWidth());  //120
        console.log($(".sar").outerWidth());  //121
        console.log($(".sar").outerWidth(true));  //141
      })
    });
  </script>
  <div class="sar" style="width:100px; height:100px; border:1px solid red; padding:10px; margin:10px;">
    <span>Hello</span>
  </div>
  <button>click</button>
</body>
```

position():- It will give top and left content position and it is relative to the parent element,

offset():- will give top and left position content position with padding and it is relative to the document.

```
<body>
  <script>
    $(document).ready(function () {
```

```
    $('.btn1').click(function () {
      console.log($(".sar h2").position());  //// Object { top: 81, left: 8 } h2 position h2 position
    });
    $('.btn').click(function () {
      console.log($(".sar h2").offset());  //Object { top: 101, left: 8 }  h2 offset
      let x = ($(".sar h2").offset());
      console.log("offset top position is: " + x.top) // offset top position is: 101
      ($(".sar h2").offset({top:0, left:0})); // It will take the h2 tag top as offset works relative to the screen



    })
  });
</script>
<h1>jquery offset and position method</h1>
<div class="sar" style="width:200px; height:200px; background-color: aqua; border:1px solid salmon;">
  <h2>Test Box</h2>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum cum voluptatum facilis
    repudiandae, incidunt exercitationem facere eos expedita ipsam laudantium.
  </p>
</div>
<button class="btn1" style="margin-top:10px ;">position</button>
<button class="btn">offSet</button>
</body>
```

scrollTop() & scrollLeft()

```
<script>
  $(document).ready(function () {
    $('.btn1').click(function () {
      console.log($(window).scrollTop());  // to get scroll top value
      console.log($(window).scrollLeft(200));  // to set 200px to left on button click
    });
  });
</script>
<div class="sar">
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum cum voluptatum facilis
    repudiandae, incidunt exercitationem facere eos expedita ipsam laudantium.
  </p>
</div>
<button class="btn1" style="margin-top:10px ;">position</button>
</body>
```

5.1 What are the basic selectors in jQuery ?

- Element ID
- CSS Name
- Tag Name
- DOM hierarchy

5.2 What are the types of selectors in jQuery?

There are three types of selectors in jQuery:

- CSS Selector
- XPath Selector
- Custom Selector

5.3 What is the slowest selector in jQuery ?

Class selectors are the slowest selectors in jQuery.

5.4 Which is the fastest selector in jQuery ?

ID and Element selector are the fastest selectors in jQuery.

6. What is the exact difference between the methods window.onload() and document.ready()?

The document.ready() event happens once all HTML documents are loaded and it is jQuery method, however window.onload() happens once all content (including images) has been loaded or when DOM and images are loaded on the page and it is JavaScript method. So, the document.ready() event fires first.

The $(document).ready() function is a jQuery extension and used to initialize a new page or to load scripts or styles into the document. However, it can also be used to load a script or style into the document at any time. It's not useful for loading scripts or styles into the document when the document is already loaded.

```
<!-- <script src="jquery.js"></script> -->
  <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
  <script>
    $(window).load(function () {
      alert("window loaded");    // will load third
    });
    $(document).ready(function () {
      alert("document loaded");    // will load first
    });
  </script>
</head>
<body>
  <img id="img1" src="./My Passport-size image.png" alt="my image" />
  <!-- will load second -->
</body>
```

```
<script src="jquery.js"></script>
  <script>

    $(document).ready(function () {
      alert("document loaded");    // will load second
    });
    $(window).on("load", function () {
      alert("window loaded");
      // The $(window).load() is NOT available in jQuery 3.0
      //To get around it, you can use it as an "Event Handler Attachment" It will print first
    });
  </script>
</head>
<body>
  <img id="img1" src="./My Passport-size image.png" alt="my image" />
  <!-- will load third -->
</body>
```

6.1 Is it possible to pause or postpone the execution of the document.ready for a period of time?

Yes, that is possible. With the release of jQuery 1.6, a new method called "jQuery.holdReady(hold)" was added. This function allows you to postpone the execution of the document.ready() event.
The **document.ready()** event is triggered as soon as your DOM is ready, however, there may be occasions when you wish to load extra JavaScript or plugins that you have referenced.

6.2 Can you use multiple document.ready() function on the same page?

Yes. You can use any number of document.ready() function on the same page. For example:

6.3 What is the use of CSS() method in jQuery?

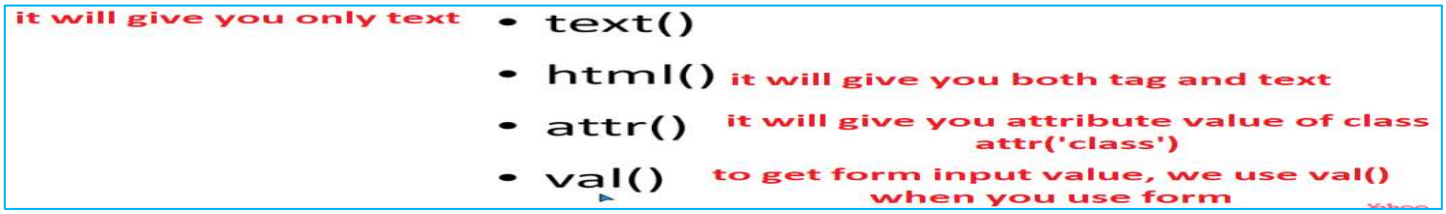
The CSS() method is used to change the style property of the selected element as shown above.

```
<script src="jquery.js"></script>
```

```
</head>
<body>
   <h1> I am bing used in jQuery</h1>
  <script>
    $(document).ready(function () {
      $("h1").css("background-color", "blue");
    });

    $(document).ready(function () {
      $("h1").css("color", "red");
    });
  </script>
```

I am bing used in jQuery

6.3.1 Get and set method




```
<body>
    <script>
      $(document).ready(function(){
        $('button').click(function(){
        let x =  $('form').html();
        let y=  $('form').text();
        let z=  $('form').attr('class');
          console.log(x);  // <p>Hello</p>  <input type="text">
          console.log(y);  //   Hello
          console.log(z);  //   sar
        })
      })
    </script>
  <form class="sar" id="fraz">
    <p>Hello</p>
    <input type="text">
  </form>
  <button>click</button>
</body>
```

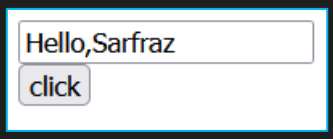
Set method example

```
<body>
    <script>
      $(document).ready(function(){
        $('button').click(function(){
        let x =  $('form').html("<b>I am changed</b>");
        let y=  $('form').text("Hello Sarfraz");
        let z=  $('form').attr('class', 'red'); // first parameter will have class or id and 2nd will have value. We have sed
        under style to clas red as  .red{color:red}

          console.log(x);  //form will be replaced with bold I am changed
          console.log(y);  //   Hello
          console.log(z);  //   sar
        })
      })
    </script>
  <form class="sar" id="fraz">
    <p>Hello</p>
    <input type="text" id="inputarea">
  </form>
  <button>click</button>
</body>
```

Val() method as set method

```
<body>
  <script>
```

```
    $(document).ready(function () {
      $('button').click(function () {
        let v = $('#inputarea').val('Hello,Sarfraz'); //well set on button click
      })
    })
  </script>
  <form class="sar" id="fraz">
    <input type="text" id="inputarea">
  </form>
  <button>click</button>
</body>
```



6.4 Explain how CSS classes can be manipulated in HTML using jQuery.

Query provides several methods to manipulate the CSS classes assigned to HTML elements. The most important methods are addClass(), removeClass() and toggleClass().

```
  <style>
    .class1 {
      color: red;
    }
    .class2 {
      font-size:50px;
    }
  </style>
  <script src="jquery.js"></script>
</head>
<body>
  <script>
    $(document).ready(function () {
      $('button').click(function(){
        // $('div').addClass('class1 class2');    // to add the class
        // $('div').removeClass('class2');       // to remove the class
        $('div').toggleClass('class2');    // to toggle the class
      });
    });
  </script>
  <div>class will be added, removed and toggled</div>
  <button>Click</button>
</body>
```

6.5 hasClass() method

```
<body>
  <script>
    $(document).ready(function () {
      $('.btn1').click(function () {
        console.log($('.sar').hasClass('apple')); // It will return true as we have apple class
      });
    });
  </script>
  <div class="sar apple">
  </div>
  <button class="btn1" style="margin-top:10px ;">position</button>
</body>
```

7. What is the $() function in the jQuery library?

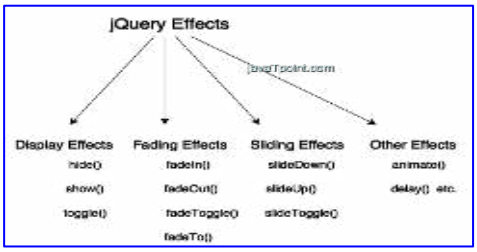
The $() function is used to access the properties of elements in the DOM (Document Object Model). $() is similar to avaScript's selector functions, but it is more powerful and has more options.

$() can be used to access attributes, classes, id, data-* attributes, and more.

Example - Suppose you want to change the colour of all the heading1 (h1) to green, then you can do his with the help of jQuery as -

```
$(document).ready(function () {
      $("h1").css("background-color", "green");
    });
```

8. What are the effects methods used in jQuery?



These are some effects methods used in jQuery:

- show() - It displays or shows the selected elements.
- hide() - It hides the matched or selected elements.

8.1 Which jQuery method is used to hide selected elements?

Ans:- The hide() function in jQuery is used to try and hide the chosen element.

- toggle() - It shows or hides the matched elements. In other words, it toggles between the hide() and shows() methods.
- fadeIn() - It shows the matched elements by fading it to opaque. In other words, it fades in the selected elements.
- fadeOut() - It shows the matched elements by fading it to transparent. In other words, it fades out the selected.

hide() & show() Method

```
<body>
  <script>
    $(document).ready(function () {
      $('.btn1').click(function () {
        $('.sar').hide();

        $('.btn2').click(function () {
          $('.sar').show()});
      });
      });
  </script>
  <div class="sar">
    <P>Hello</P>
  </div>
  <button class="btn1" style="margin-top:10px ;">hide</button>
  <button class="btn2" style="margin-top:10px ;">show</button>
</body>
```

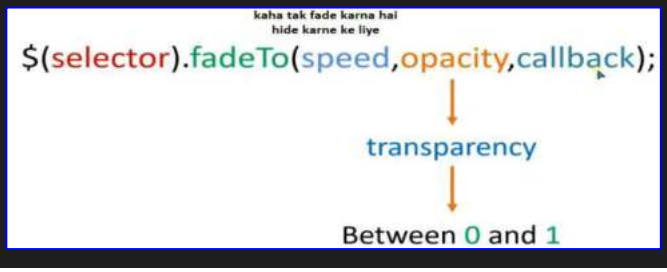
fadeIn(), fadeOut()

```
<body>
  <script>
    $(document).ready(function () {
      $('.btn1').click(function () {
        $('.sar').fadeIn("slow");   // To show // It has two parameters for speed slow, fast, 1000(means 1 second)

        $('.btn2').click(function () {
          $('.sar').fadeOut(3000)  // To hide by fading in 3 second
        });
      });
    });
  </script>
  <div class="sar">
    <P>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Beatae nulla fugiat provident molestias assumenda
      reiciendis, eos, ex harum rem magnam tempore enim dolorum iste reprehenderit ipsam nostrum quis eaque voluptates?
    </P>
  </div>
  <button class="btn1" style="margin-top:10px ;">show/fadeIn</button>
  <button class="btn2" style="margin-top:10px ;">hide/fadeOut</button>
</body>
-----------------------------------------------------------------------------------------------------------------------
        $('.btn2').click(function () {
```

```
        $('.sar').fadeOut(3000, function(){
          console.log("Hello")  //// To hide by fading in 3 second with message
        })
```

fadeToggle()

```
<body>
  <script>
    $(document).ready(function () {
      $('.btn1').click(function () {
        $('.sar').toggle();
        $('.sar').fadeToggle(); // it will hide and show by fading in and out

      });
      });
  </script>
  <div class="sar">
    <P>Hello</P>
  </div>
  <button class="btn1" style="margin-top:10px ;">toggle</button>
</body>
```



fadeTo()

```
    <script>
        $(document).ready(function () {
            $('.btn2').click(function () {
                $('.sar').fadeTo(3000, 0.5, function () {
                    console.log("Hello")  //Kaha tab fade karna hai 0.5 meas half trasparent
                })
            });
        });
    </script>
```

slide down, slide up and slide toggle(It will hide and show by sliding and gives animation effect)

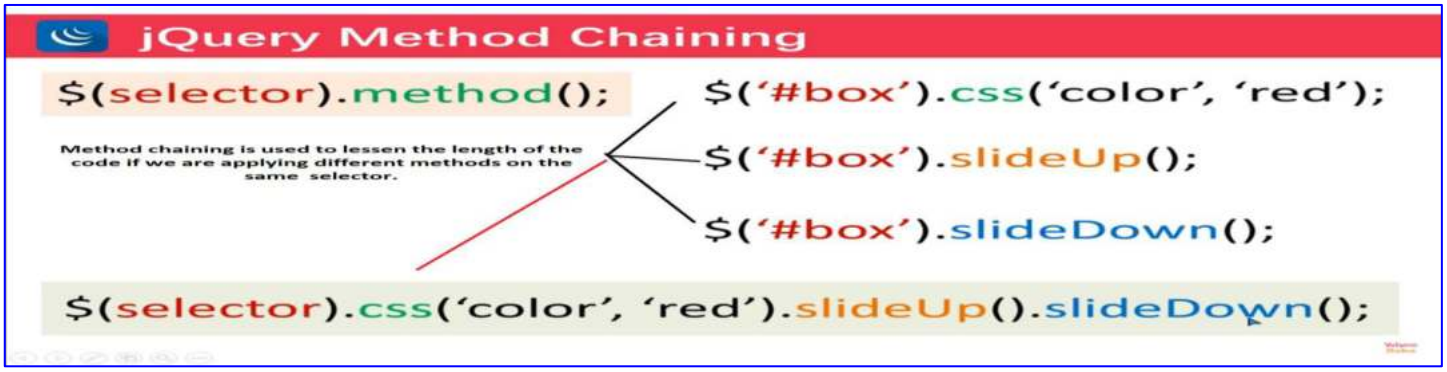
```
<body>

  <script>
    $(document).ready(function () {
      $('.btn1').click(function () {
        $('.sar').slideUp("slow");   // To show // It has two parameters for speed slow, fast, 1000(means 1 second)

        $('.btn2').click(function () {
          $('.sar').slideDown(3000, function () {
            console.log("Hello")  //// To hide by fading in 3 second with message
          })
          $('.btn3').click(function () {
            $('.sar').slideToggle(3000, function () {
              console.log("Hello")  //// To hide by fading in 3 second with message
            })
          });
        });
      });
    });
   </script>
    <div class="sar" style="width:200px; height:200px; background-color:aqua">
        <P>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Beatae nulla fugiat provident molestias assumenda
            reiciendis, eos, ex harum rem magnam tempore enim dolorum iste reprehenderit ipsam nostrum quis eaque
      voluptates?
          </P>
      </div>
    <button class="btn1" style="margin-top:10px ;">slideUp</button>
    <button class="btn2" style="margin-top:10px ;">slideDown</button>
    <button class="btn3" style="margin-top:10px ;">slideToggle</button>
</body>
```

Animate Method(to animate any element)


```
<body>
  <script>

  </script>
    <div class="sar" style="width:200px; height:200px; background-color:aqua; position: relative;">
        <P>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Beatae nulla fugiat provident molestias assumenda
            reiciendis, eos, ex harum rem magnam tempore enim dolorum iste reprehenderit ipsam nostrum quis eaque
      voluptates?
          </P>
      </div>
    <button class="btn1" style="margin-top:10px ;">Animate</button>
</body>
```

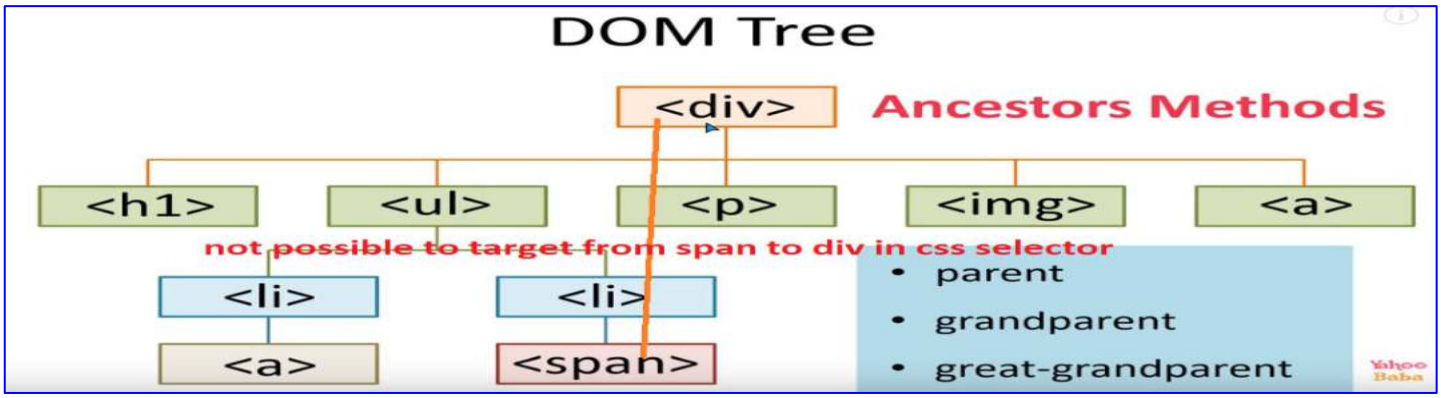
Stop method(it is used to stop the animation in between) Stop(true)- will stop all the animation Stop(true, true)- send parameter as a true will take the animation at the end quickly

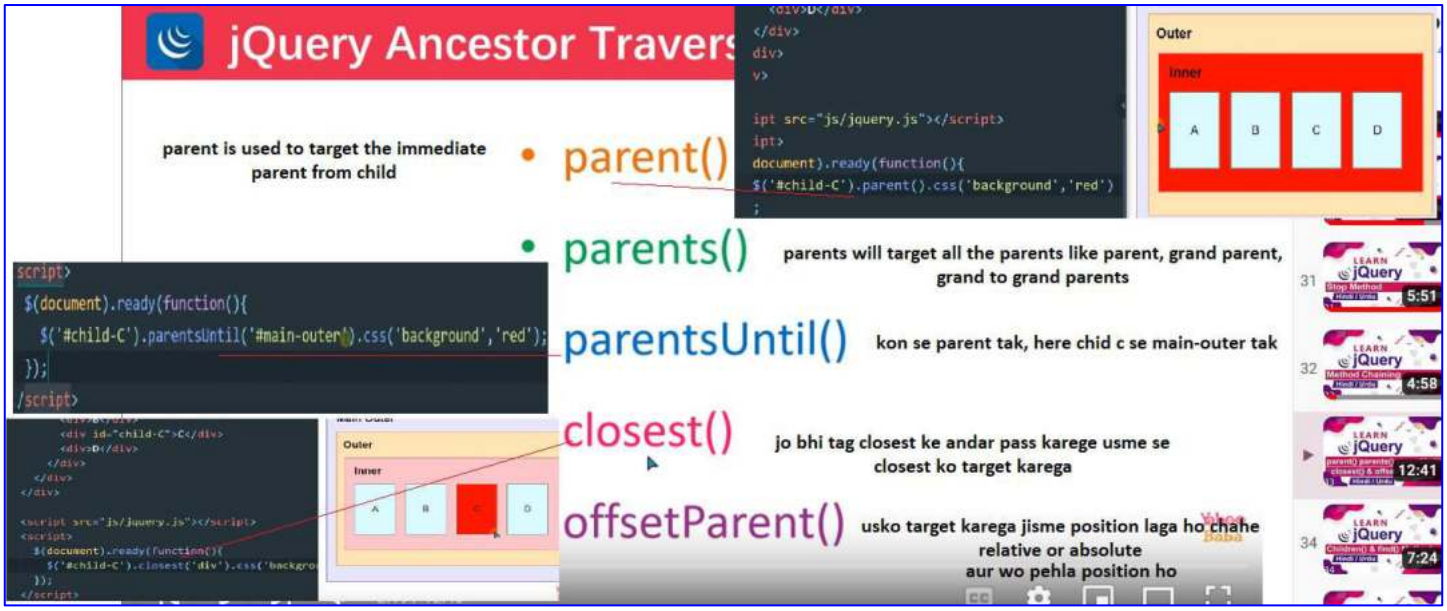
```
<script>
    $(document).ready(function () {
      $('.btn1').click(function () {
        $('.sar').animate({ bottom : '150px'}, 3000);  // First step of  animation
        $('.sar').animate({ left : '150px'}, 3000);  // Second step of animation
        $('.sar').animate({ borderRadius : '100px'}, 3000);
      });
      $('.btn2').click(function(){
        $('.sar').stop(true)  // We pass true to stop all the animation and if we pass stop(true, true )
       // $('.sar').stop(true,true)  // We pass true to stop all the animation and if we pass stop(true, true )
        // the second true will take the animation at the end point quickly
        // If we pass true as a parameter, second step of animation left 150 and borderRadius will not
        //execute but without true second animation will execute if clicked on stop during first animation.
      })
    });
  </script>
    <div class="sar" style="width:200px; height:200px; background-color:aqua; position: relative;">
        <P>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Beatae nulla fugiat provident molestias assumenda
            reiciendis, eos, ex harum rem magnam tempore enim dolorum iste reprehenderit ipsam nostrum quis eaque
      voluptates?
          </P>
      </div>
    <button class="btn1" style="margin-top:10px ;">Animate</button>
    <button class="btn2" style="margin-top:10px ;">Stop</button>
</body>
```



8.2 DOM MANIPULATION(DOM TRAVERSERING METHODS)

We know that we can not do all works with CSS selectors such as id, class, name , universal etc. jQuery provides some methods to target DOM which is not possible with CSS selectors and they are **Ancestors, Descendants(children(), find()), Siblings, Filtering methods**. If we have to target parent element from grand child, it is not possible with CSS selectors.





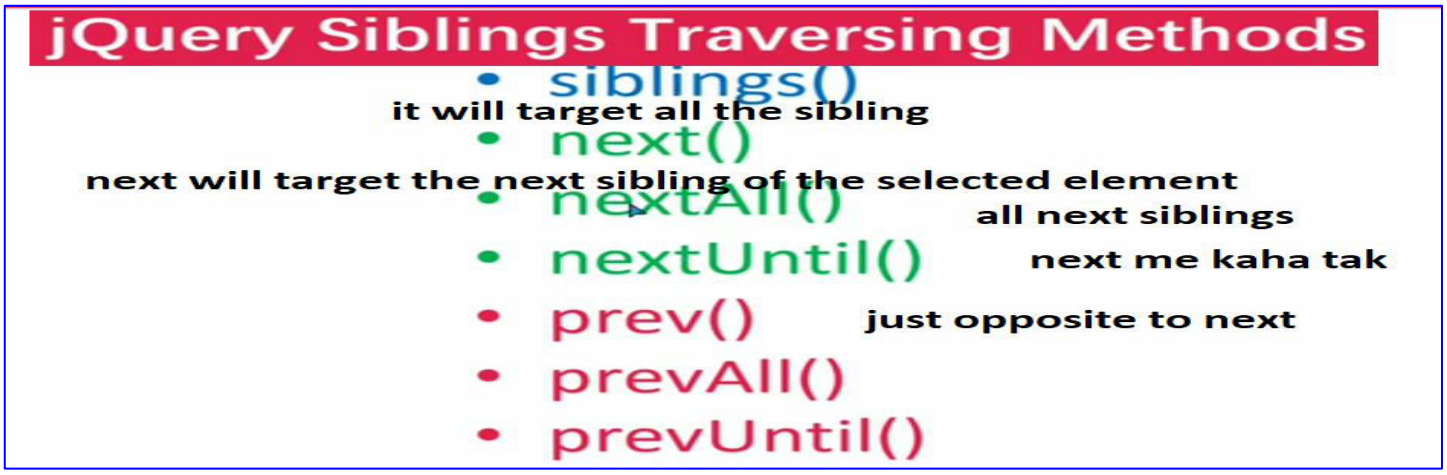
Descendants(children(), find())

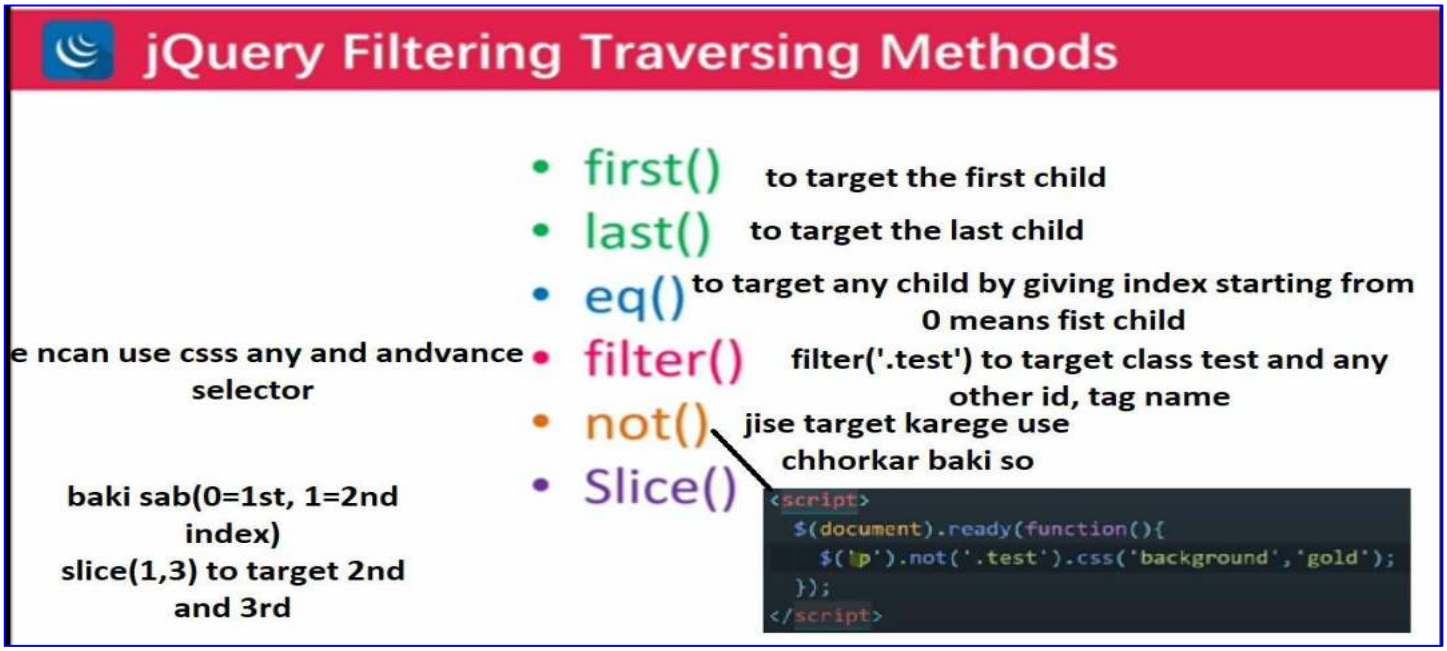
Children is used to target all the children and we can also select a particular child by out of many children by its tag name, id or class selector. Children will find the selected element only in the children not grand children

Find() method is used to fine any child in the given children. Find() method will find the selected element whether it is in children, grand-children

```
<body>
  <script>
      $(document).ready(function(){
        $('.btn').click(function(){
          $('.main').children().css('color', 'red'); // all the children will be targeted
          $('.main').children('.child').css('color', 'blue'); // present as child
          $('.main').find('ul li').css('background-color', 'pink');  // It will target li present in grand child
        })
      });
  </script>
  <div class="main">
    <div class="child">
      <p>sibling1</p>
      <p>sibling2</p>
      <p>sibling3</p>
      <p>sibling4</p>
      <ul>
        <li>list tag</li>
      </ul>
    </div>
```

```
    <h1>Hello, I am Sarfraz</h1>
  </div>
  <button class="btn">click</button>
</body>
```






```
<body>
  <script>
      $(document).ready(function(){
        $('.btn').click(function(){
          // $('.child p').slice(1, 3).css('color', 'red'); // sibling2 and sibling 3 ko target karega
          // $('.child p').not('.cl').css('color', 'blue'); // cl class ke chhokar sare p tag ko target karega
          $('p').filter('.cl').css('color', 'yellow'); // it will check paragraph cl class
        //  $('.child p').css('color', 'yellow'); // The same will be targeted without filter method
        })
      });
  </script>
  <div class="main">
    <div class="child">
      <p>sibling1</p>
      <p>sibling2</p>
      <p class="cl">sibling3</p>
      <p>sibling4</p>
      <ul>
        <li>list tag</li>
      </ul>
    </div>
    <h1>Hello, I am Sarfraz</h1>
  </div>
  <button class="btn">click</button>
</body>
```

Has method is used to target the child from parent or to check if a parent or any element has particular child and is method is used in any condition like if div tag has p tag as a child do this.

has method()

```
<body>
  <script>
      $(document).ready(function(){
        $('.btn').click(function(){
          $('h1').has('i').css('color', 'yellow'); //to target the child of h1, p, and class sar
          $('p').has('span').css('color', 'red');
          $('.sar').has('p').css('fontSize', '25px');
        })
      });
  </script>
    <h1> <i>Hello, I am Sarfraz</i></h1>
    <p> <span>Hello, I am Sarfraz</span></p>
    <div class="sar"><p>Sarfraz</p></div>
  <button class="btn">click</button>
</body>
```

is() method

```
<body>
  <script>
      $(document).ready(function(){
        $('.btn').click(function(){
          if($('span').parent().is('p')){
              $('span').css('fontSize', '50px');
          }else{
            $(this).css('color', 'red');
          }
        })
      });
  </script>
    <p> <span>Hello, I am Sarfraz</span></p>
  <button class="btn">click</button>
</body>
```

eachMethod () method/function

It works like loop method and runs through each element. So, we can target any element and do whatever we want.

```
<body>
  <script>
    $(document).ready(function () {
      $('ul').each(function(){
      $('ul li').text('sar'); // It will loop for each element and replace with 'sar'
      });
    });
  </script>
  <ul>
    <li>one</li>
    <li>two</li>
    <li>three</li>
    <li>four</li>
  </ul>
</body>
```

- sar
- sar
- sar
- sar

9. Can you tell something about jQuery each() method?

The each() method in jQuery allows us to loop through different datasets such as arrays or objects (even DOM objects).

It can be used to loop through a number of DOM objects from the same selectors.

We can also use each() to loop through the arrays and object of data and get the index and the value of the position of data inside the array.

```
  <script>
    var list = ["InterviewBit", "jQuery", "Questions"];
    $.each(list, function (index, value) {
      console.log(index + " " + value);
    })
    var obj = { "name": "InterviewBit", "type": "jQuery" };
    $.each(obj, function (key, value) {
      console.log(key + " - " + value);
    })
  </script>
```

```
0 InterviewBit
1 jQuery
2 Questions
name - InterviewBit
type - jQuery
```

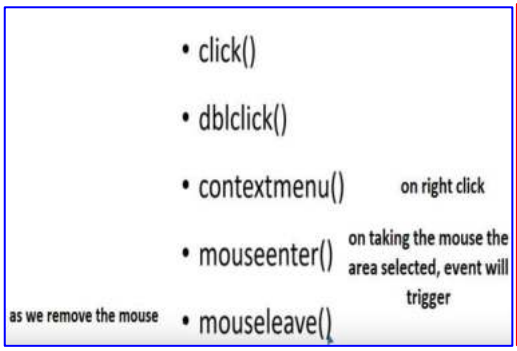
9.1 How to iterate/loop through all p elements in jQuery.

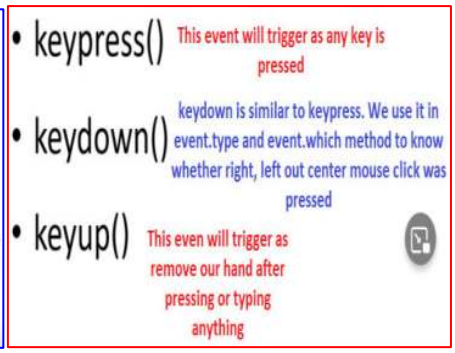
```
<body>
    <p id="h">Hello</p>
    <p id="w">Hello1</p>
  <script>
    $("p").each(function() {
   console.log( $(this).attr('id'));  // it will return h and w
   console.log( $(this).attr('id', 'sar')); // it will set sar in both of them as id value
});
  </script>
```

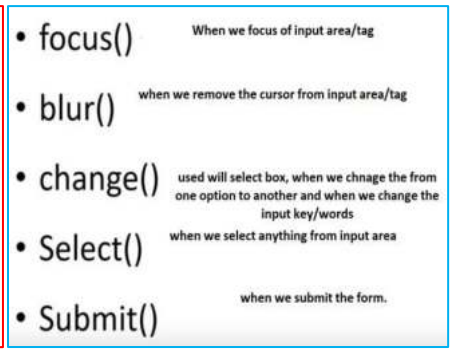
get() and set() method in jQuery

10. Define event.

User actions on a web page are called events and handling responses to those is called event handling. jQuery provides simple methods for attaching event handlers to selected elements. When an event occurs, the provided function is executed.







Mousedown:- means right, left and center click

Mouse events

```
  <script>
      $(document).ready(function(){
        $('body').click(function(){
          $(this).css('background', 'red'); // pure body mein he style lag jaye
        });
      });

  </script>
  <body>I am Sarfraz</body>
</body>
<body>
  <script>
      $(document).ready(function(){
        $('body').keypress(function(){
          $(this).css('background', 'red'); // pure body mein he style lag jaye
        });
      });
  </script>
```

```
  <body></body>
</body>
```

Form events

```
<body>
  <script>
    $(document).ready(function () {
      $('#inputarea').focus(function () {
        $(this).css('background', 'red');
      });
      $('#inputarea').blur(function () {
        $(this).css('background', 'yellow');
      });

      $('select').change(function () {
        let x = $(this).css('background', 'blue');// To set background when changed the option.
        $('div').html(x.val());// the value will store inside the div
      });
      $('#inputarea').change(function () {
        $('div').html($(this).val());// the value will store inside the div
      });
      $('#inputarea').select(function () {
        $(this).css('fontSize', '25px')// the font Size will increase if we select the input area portion
      });
      $('form').submit(function () {
        window.alert('The form has been submitted');
      });
    });
  </script>
  <form action="">
    <label for="">Your Message</label> <input type="text" id="inputarea"><br>
    <label for="">Your Country</label> <select><label>Country</label>
      <option value="India">India</option>
      <option value="Japan">Japan</option>
      <option value="Pakistan">Pakistan</option>
    </select><br>
    <input type="submit" value="Submit">
  </form>
  <div></div>
</body>
```

X and Y Co-ordinates (pageX, pageY)

10. Describe jQuery Connect in brief. Also, list its uses

```
<body>
  <script>
    $(document).ready(function () {
      $(document).mousemove(function(event){
        $('div').text("X-co-ordinate is : " + event.pageX); //pageX is used to get x axis co-ordinate
        $('#div2').text("Y-co-ordinate is : "+ event.pageY); ////pageY is used to get y axis co-ordinate
      });
    });
  </script>
  <h1>PageX & PageY(x & y co-ordinates)</h1>
  <div>X Co-ordinate will be shown here</div>
  <div id="div2">Y Co-ordinate will be shown here</div>
</body>
```

11. Describe Event all type

event.type method(to know about event type applied)

```
<body>
  <script>
    $(document).ready(function () {
```

```
      $('h1').on('click dblclick mouseover', function (event) {
        $('h1').html(event.type)
      });
    });
  </script>
  <h1>To know event type</h1>
</body>
```

Multiple events and on() method or method chaining

```
<script>
   $(document).ready(function(){
    $('h1').click(function(){
      $(this).css('background', 'red');
    })
    $('h1').dblclick(function(){

      $(this).css('background', 'blue');

    })
    $('h1').mouseenter(function(){
      $(this).css('background', 'yellow');
    })
    $('h1').mouseout(function(){
      $(this).css('background', 'pink');
    })
   })
  </script>
  <h1 >To know event type</h1>
```

On method to apply multiple events

[on method has two parameters, first is event with single or double quotation and second is function to do what you want]

```
<body>
  <script>
    $(document).ready(function () {
      $('h1').on('click dblclick mouseover' , function () {
        $('h1').css('color', 'blue'); // click, doubleclick and mouse over pe color blue ho jayega
      })
    });
  </script>
  <h1>To know event type</h1>
</body>
```

To add and remove event using on and off method

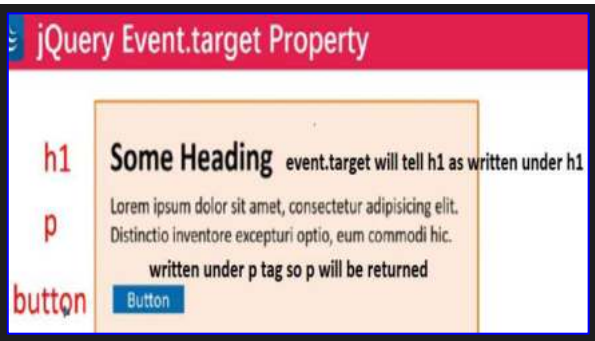
```
<body>
  <script>
    $(document).ready(function () {
      $('h1').on({
        'click': function () {
          $(this).css('color', 'red')

        }, 'dblclick': function () {
          $(this).css('color', 'blue')
        }
      })
      $('button').click(function(){
        $('h1').off("dblclick");  //It will remove the double click even on button click
      });
    });
  </script>
  <h1>To know event type</h1>
  <button>click</button>
</body>
```

```
<body>
  <script>
```

```
    $(document).ready(function () {
      $('h1').on('mousedown mouseover mousedown mouseout click dblclick', function (event) {
        $('h2').html(event.type +':' +event.which) // It will return which event was
        // applied and event.which will be tell if right, left and center clicked that happened with mousedown
      });
    });
  </script>
  <h1 style="width:200px ; height: 200px; background-color: aqua;">To know event type</h1>
  <h2>To know event type</h2>
</body>
```

Event.target() method(It will tell which html tag I have clicked)

```
<body>
  <script>
    $(document).ready(function () {
    $('div').on('click', function(event){
      $('p').html(event.target.nodeName)
   $('p').html(event.target.innerHTML) to get innerHTML ON YOU CLICK
// We need to use JavaScript property nodeName to get exact tag name
    })
    });
  </script>
  <div style="width:200px ; height: 200px; background-color: aqua;">
    <h2>To know event type</h2>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Suscipit, fugiat.</p>
    <button>Click</button>
  </div>
  <p></p>
</body>
```



event.preventDefault() will stope the default property. Here, anchor tag will not work

```
<body>
  <script>
    $(document).ready(function () {
    $('a').on('click', function(event){
      event.preventDefault();
    })
    });
  </script>
    <a href="https://www.youtube.com/watch?v=YMtCERAmvM4&list=PL0b6OzIxLPbzSyiC0PFaqeabe1aGhfrbW&index=45">Link</a>
</body>
```

event.stopPropagation();

```
  <div id= 'hello' style="width:200px ; height:200px; border:1px solid red;">
    <h1>To know event type</h1>
    <button>click</button>
  </div>
  <script>
    $(document).ready(function () {

      $('#hello').click(function () {
        alert('div clicked')
      });
      $('h1').click(function (event) {
        event.stopPropagation();
        alert(' h1 clicked')
      });
      $('button').click(function (event) {
        event.stopPropagation();
        alert('button clicked')
      })
    });
  </script>
```

Event.data() is a property to read the index of data you click on

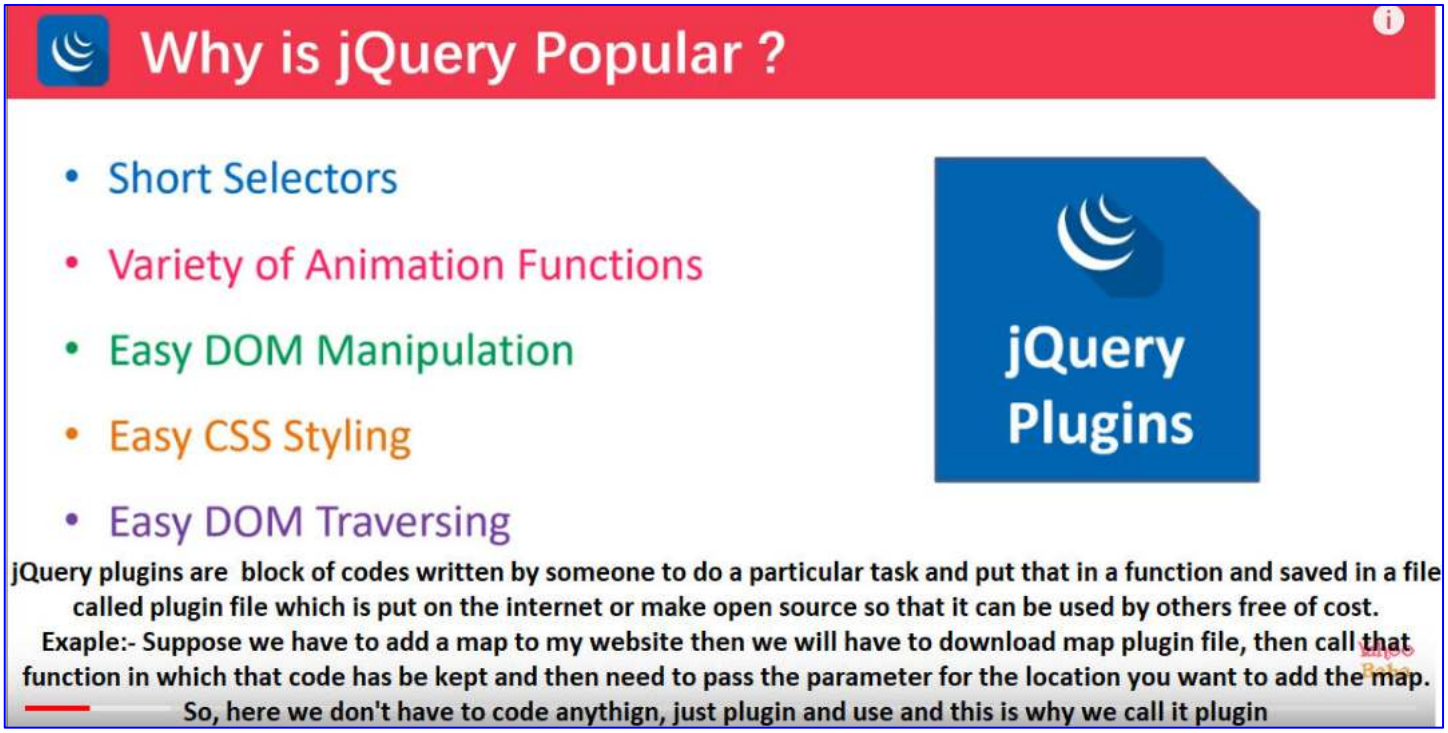
```
<body>
  <div id= 'hello' style="width:200px ; height:200px; border:1px solid red;">
```

```
    <p>para1</p>
    <p>para2</p>
    <p>para3</p>
    <p>para4</p>
  </div>
  <script>
    $(document).ready(function () {
      $('p').each(function(i){
        $(this).on('click', {value:i}, function(event){
          alert(event.data.value); // It will return the index of paragraph on which we click using event.data

        })
      })
    });
  </script>
</body>
```

Passing data to event handler using event.data property

```
<body>
    <button>Click</button>
  <script>
    $(document).ready(function () {
      $('button').on('click', { firsName: 'Sarfraz', lastName: 'Alam' }, function (event) {
        alert(`My fist name is ${event.data.firsName} and my last name is  ${event.data.lastName}`);
      })
    })
  </script>
</body>
```

Plugins(Features of jQuery)



How to add plugins

The given website has plugins for both jQuery and JavaScript, premium one is paid plugins, on the right hand side we have all the categories such as menu, accordions, animation, nav etc.

1. Visit https://www.bestjquery.com/
2. Follow the link https://www.youtube.com/watch?v=wFKVd8wnhyc

12. Describe jQuery Connect in brief. Also, list its uses.

A **jQuery connect** is a plugin that is used to connect or bind a function with another function. Connect is used to execute the function from the other function or plugin is executed.

Connect also lets you connect a function to a DOM object. With connect, you bind more than one function.
Example

```
$.connect(obj1, 'fun1', obj2, fun2);
```

Here we are binding **fun2** function of **object 2** to the fun1 function of **object 1**. So, when fun2 is executed, fun1 will also be executed.

How to use connect

Download jQuery connect file from jQuery.com

Include that file in the HTML file.

Use $.connect function to connect a function to another function.

13. What is jQuery Mobile or mobile-first app?

Query Mobile is a JavaScript library that enables developers to create mobile-first applications. It is a lightweight framework that allows developers to create rich, touch-first interfaces that are optimized for touch devices.

jQuery Mobile is a mobile-first JavaScript library that aims to provide a modern and easy-to-use framework for developing mobile apps. It was originally developed by Facebook to improve the performance of its iOS and Android apps. Since then, it has been adopted by many other companies, including Google, Yahoo, and Mozilla.

When using jQuery Mobile, you can use the library to create a mobile-first app that is faster and easier to maintain than traditional web apps. It also makes it easier to add new features to your app without having to rewrite your code.

JQuery Mobile is built on top of the jQuery JavaScript library. It works by using the same techniques that are used when you are developing a traditional web app. However, instead of using JavaScript, jQuery Mobile uses HTML5 and CSS3 to create a modern and easy-to-use framework for developing mobile app

14. What is the significance of jQuery.length?

jQuery.length property is used to **count the number of the elements** of the jQuery object.

```
</head>
<p>sarf</p>
<p>sarf</p>
<p>sarf</p>
<body>
  <script>
    let x = $('p').length;
    console.log(x);  // 3
  </script>
</body>
```

15. What is the purpose of J Query's delay() method? Can you use this for different types of browsers like (Internet Explorer)?

The delay() method is used to set the delay between two events. It has two parameters first is duration which is give in milliseconds 1000 ms= 1 second, second parameter is que name which is optional.

The **delay() method** is an inbuilt method in jQuery that is used to set a timer to delay the execution of the next item in the queue. **Syntax:**

```
$(selector).delay(para1, para2);
```

Parameter: It accepts two parameters which are specified below:

- **para1:** It specifies the speed of the delay.
- **para2:** It is optional and specifies the name of the queue.

This can be useful when you want to wait for an event to occur before doing something else.

But the latest Microsoft browser (Microsoft Edge) that replaces the internet explorer has support for the jQuery delay() method.

```
<body>
  <script>
    $(document).ready(function () {
      $('h1').click(function () {
       $('h1').slideUp(5000)  // hide in 5 seconds and then font size will increase and then delay for 5 seconds
              .css('fontSize', '50px')
              .delay(5000)
              .fadeIn(5000);  // show in 5 seconds
      })
    })
  </script>
  <h1>To know event type</h1>
</body>
```

2nd Example:-

```
<body>
  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("#d1").delay("slow").fadeIn();
        $("#d2").delay("fast").fadeIn();
        $("#d3").delay(1000).fadeIn();
        $("#d4").delay(4000).fadeIn();
      });
    });
  </script>
  <button>Click</button>
</body>
```

16. Explain jQuery no-conflict.

jQuery no-conflict is a jQuery option that allows you to avoid conflicts between various JavaScript frameworks or libraries and jQuery. Suppose, we are creating a website in which we have used jQuery and along with that we also want to use JavaScript other frameworks such as Angular.js, backbone, Ember, Knockout and libraries. We also use $() dollar sign When we use jQuery shortcut as a selector, but we also use $ dollar sign for other frameworks as and if we use the same sign for two frameworks or libraries, it will create conflict or give error. So, to avoid this issue, we use jQuery no-conflict. So, we will replace $ sign to jQuery and it will work same as $() sign without any conflict.

```
  <style>
    .class2 {
      font-size:50px;
    }
  </style>
  <script src="jquery.js"></script>
</head>
<body>
```

```
  <script>
    $.noConflict();              // no conflict has been enabled and jQuery used instead of $
    jQuery(document).ready(function () {
      jQuery('button').click(function(){
        jQuery('div').toggleClass('class2');    // to toggle the class
      });
    });
  </script>
  <div>class will be added, removed and toggled</div>
  <button>Click</button>
</body>
```

17. What exactly is a jQuery Data Table plug-in? Also, explain the uses with examples.

A data table plug-in is a jQuery plugin that can be used to create custom tables. It allows you to create a table with a custom layout, without having to write any code. It can be used to create tables with different layouts, or even to create tables with custom columns. The data table plug-in is a great way to add extra functionality to your websites, especially if you want to add a table to your website that is not included in the default layout.

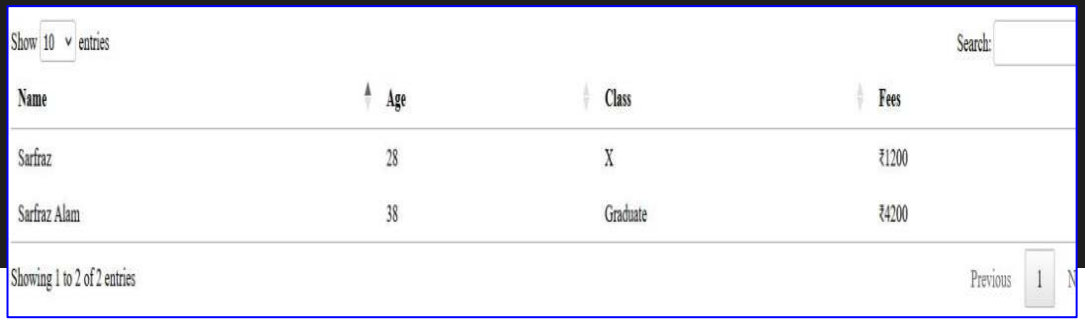
Some of the uses for data tables are listed below -

Data tables are great for displaying a lot of data at once. **For example**, you could display a list of products on your website. You could also display the same data in a table in your blog.

Data tables are also great for displaying information that you want your visitors to see. **For example**, you could display a list of products that you're selling on your website. You could also display the same information in a table in your blog.

Data tables are great for showing a lot of information in one place. **For example**, you could display a list. of products on your website and then add additional information about each product in the table.

```
<title>Document</title>
  <script src="jquery.js"></script>  // jquery downloaded file added
  <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.13.1/css/jquery.dataTables.css"> // table cdn
        CSS path added
  <script type="text/javascript" charset="utf8" src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.js"></script>
// CDN javascript path added
</head>
<body>
  <script>
    $(document).ready(function () {
      $('#tbl').DataTable();             // function called
    });
  </script>
  <table id="tbl">
    <thead>
      <tr>
        <th>Name</th>
        <th>Age</th>
        <th>Class</th>
        <th>Fees</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Sarfraz</td>
        <td>28</td>
        <td>X</td>
        <td>&#8377;1200</td>
      </tr>
      <tr>
        <td>Sarfraz Alam</td>
        <td>38</td>
        <td>Graduate</td>
        <td>&#8377;4200</td>
```



```
      </tr>
    </tbody>
  </table>
</body>
```

18. In jQuery, distinguish between the bind(), live(), and delegate() functions.

bind () function is used to bind or attach an event to the handler for the HTML elements.
Syntax:- $(selector).bind(event, function). Bind actually takes any event and allows to perform action what you want to do with the event.

```
<body>
  <script>
        // $(document).ready(function(){
        //   $('button').click(function(){
        //     $('p').css("background", "blue")
        //   })
        // })
        // The same work can be done using bind
        $(document).ready(function(){
          $('button').bind('click', function(){
            $('p').css("background", "blue")
          })
        })
  </script>
      <p>I will change</p>
      <button>Click</button>
</body>
```

```
  <script>
        $(document).ready(function(){
          $('button').live('click', function(){
            $('p').css("background", "blue")
          }) // The live() method was deprecated in jQuery version 1.7, and removed in version 1.9. It will not work as using
version 3.0
        })
  </script>
      <p>I will change</p>
      <button>Click</button>
```

The **bind()** function does not connect events to items added after loading the DOM. In contrast, the **live()** and **delegate()** methods also attach events to future items.

```
<script>
    $(document).ready(function () {
      $('p').click OR bind(function () {
// It will append when v use live or delegate()
        $('p').parent().append('<p>I am new </p>');
      })
    })
  </script>
  <p>I will change</p>
```

I will change

I am new

When we click on I will change, it will append "I am new" but if we click on "I am new" again, it will not append the child or "I am new" becuase used click and the same with bind() if we use 1.7 version instead of click. But it will append "I am new" if we use delegate and live() method becuase bind function does not connect event to the DOM elements or items after loading the DOM.

The distinction between **live()** and **delegate()** methods is that **live()** does not support chaining. It will only function on a selection or an element. However, the **delegate()** function supports chaining.

The **bind()** method works only within a DOM element. If you want to bind events to items in a different element, you can use the **delegate()** function.

The **bind()** method is not recommended for use in JavaScript and in a production environment because it does not offer the same performance benefits the **delegate()** method.

The **bind()** method is only used to control the order in which items are displayed. If you need to add items to a different order, you can use the **delegate()** function.

It is suggested to use on function if using version 1.7 and above and before 1.7 delegate.

With on() and delegate() functions the event gets bubbled up to the specified parent element, whereas with live function the event gets bubbled up to the document object. Live was removed in 1.9

Both on() and delegate() functions allow us to perform event delegation and have the same operation. Before 1.7 version delegate() used.

```
  <script>
    $(document).ready(function () {
          $('ul').on('click', 'li',function(){
            $(this).fadeOut(500)  // $(this) will display none on which li we click , ul will ul and li will li
          })

          $('button').on('click', function(){
            $('ul').append('<li>New child added</li>')
          })
    })
  </script>
```

The same thing can be done with delegate method, here selector comes first then event

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.js"></script>
</head>
<body>
  <script>
    $(document).ready(function () {
          $('ul').delegate('li', 'click',function(){
            $(this).fadeOut(500)  // It will display none the ul
          })
          $('button').on('click', function(){
            $('ul').append('<li>New child added</li>')
          })
    })
  </script>
```

We will use delegate if using less than 1.7 jQuery version else on as both do the same

```
  <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.js"></script>
</head>

<body>
  <script>
    $(document).ready(function () {
      $('ul').delegate('li', 'click', function () {
        $(this).fadeOut(500)  // It will display none the ul
      })
      $('#sar').on('click', function () {
        $('ul').append('<li>New child added</li>')
      });
      $('#stop').click(function(){
        // $('ul').off('click', 'li');
        $('ul').undelegate('li', 'click'); // It will stop the li to disappear
      })
    })
  </script>
```

Live() function

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.js"></script>
</head>
<body>
  <script>
    $(document).ready(function () {
```

```
        $('ul').live('click', 'li',function(){
          $(this).fadeOut(500)  // It will display none the ul
        })
        $('button').on('click', function(){
          $('ul').append('<li>New child added</li>')
        })
  })
</script>
```

To get old link

https://learn.microsoft.com/en-us/aspnet/ajax/cdn/overview#jQuery_Releases_on_the_CDN_0

19. Distinguish between jQuery.min.js and jQuery.js.

The functionality of jquery.min.js and jquery.js is the same. However, If you are serving a page that contains a lot of JavaScript, it is a good idea to minify it. Minifying JavaScript **(jquery.min.js)** reduces the size of your file and makes it faster to load on all platforms. Minifying your code also reduces the chances that erroneous code will be rendered as a malicious attack because it makes it more difficult to test. With minified JavaScript, you will get better page performance, faster loading, and shorter wait times for your visitors.

There are also some of the major advantages of the **jquery.min.js.** -

When JS files are minified in a production environment, they load faster and provide faster and better page performance.

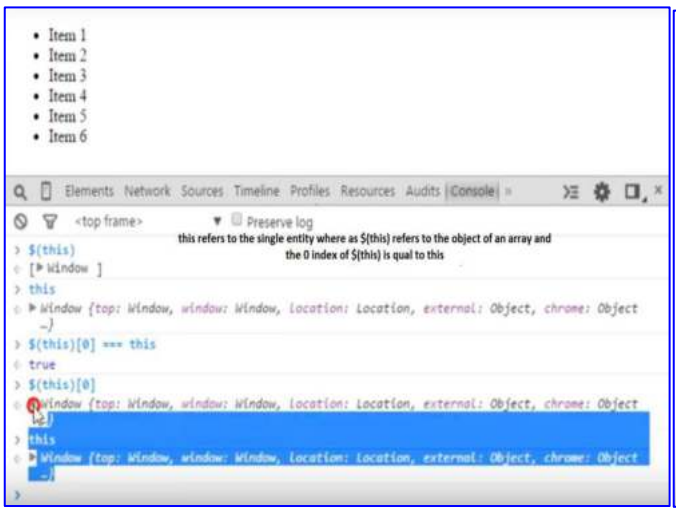
Minifying a script also reduces the size of the file in the server's temporary directory and browser cache, which results in a faster download and a smaller server response.

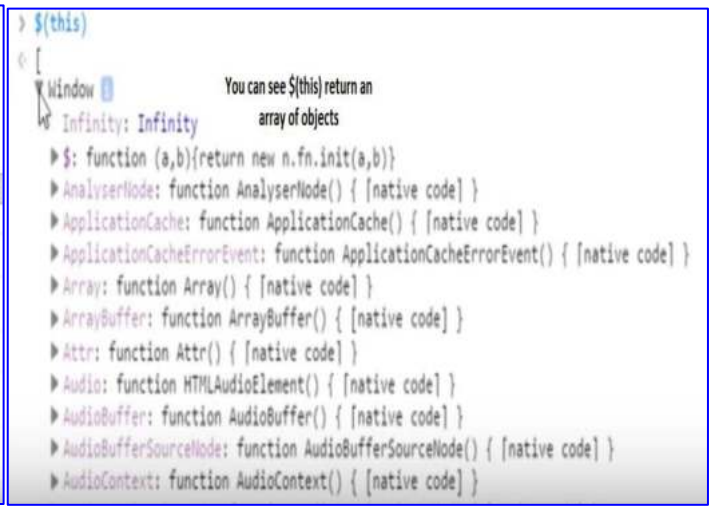
Another advantage of using minified JS is that it removes the need to include any external JS files in the HTML document. This makes it easier to structure the code and easier to edit the code.

20. In jQuery, what is the difference between $(this) and this.

$(this) is a jQuery object, whereas **this** is a JavaScript global object reference. We may refer to DOM elements in HTML documents using **this**. $(this) refers to the parent such as ul only if we have <ul> h

$(this) refers to the parent object, whereas **this** is a DOM element that, in the case of an array, represents an object with the **.each()** method, which shows the current iteration.





https://www.youtube.com/watch?v=TBVpCKNuPtw

21. Can you explain about ajaxStart(), ajaxStop() etc. functions?

AJAX is a technique not language to display the page without load. We use AJAX in JavaScript, jQuery and other programming language to communicate with the database. We can insert, update, edit , read and delete the database data. Facebook chat, comment, share and like are the example of AJAX which will not reload the entire page.

There are three ways to send AJAX request in jQuery.
Post, get and ajax
The given methods are available in jQuery.

The **ajaxStart()** event is a global event or method that occurs when an Ajax request begins or Ajax starts.

ajaxStop/ajaxComplete is also a global method that occurs/invoke when all the Ajax completed.It will occur whether Ajax is successful or unsuccessful.

ajaxSend() is also a global method that occurs when any request it sent to the server.

ajaxSuccess is also a global method that occurs when any Ajax request/call is success.

ajaxError When any error occurs. It is also a global method.

Step 1

```
function getData(){
    $.ajax()
}
```

```
Step 2
function getData(){
    $.ajax({})
}
```

```
<div id="demo"></div>
<!-- <button onclick="loadData()">Click to fetch the data</button> -->
<script>
    // function loadData() {
    //     var ajax = new XMLHttpRequest();
    //     ajax.onreadystatechange = function () {
    //         if (this.readyState == 4 && this.status == 200) {
    //             console.log(this.responseText);
    //             //alert(this.responseText);
    //             document.getElementById("demo").innerHTML = this.responseText;
    //         }
    //     };
    //     ajax.open('GET', "data.txt", true);
    //     ajax.send();
    // };

    $(document).ready(function () {
        loadData();
    })
    function loadData() {
        $.ajax({
            // url: 'data.html',
            // url: 'data.txt',
            url: 'jsonData.json',
            // url: 'https://jsonplaceholder.typicode.com/users/1/posts',
            type: 'GET',
            // dataType: 'text',
            dataType: 'json',
            success: successFn,
            error: errorFn,
```

```
            complete: function (xhr, status) {
                console.log("completed")
            }
        })
    }
    /*  function successFn(result) {
          console.log('successful');
          $('#demo').append(result);
      }
  */
    function successFn(result) {
        console.log('successful');
        $.each(result, function (index, item) {
            $.each(item, function (key, value) {
                $('#demo').append(`${key} : ${value} <br>`);
            })
        })
        $('#demo').append(result);
    }
    function errorFn(xhr, status, string) {
        console.log(xhr); //xhr is a combination of xmlHTTPRequest & xmlHTTPResponse, it will return everythig like status code,
status message
        console.log(status); // It will return error as could not be successful
        console.log(string); //statusText: not found will be returned
        console.log('Unsuccessful')
    }
    // https://www.youtube.com/watch?v=A1tZmY8jVdY
</script>
```

Posting the data in the server by taking from users through input

https://www.youtube.com/watch?v=5nL7X1UMWsc

Points to remember, when we add bootstrap & jQuery in the same file

```
<body>
    <h1 id="target"></h1>

    <script src="jquery.js"></script>
    <!-- First we need to add jQuery then bootsrap js file and then javascript -->
    <!-- <script src="bootstrap/js/bootstrap.min.js"></script> -->
    <script src="ajax.js"></script>
</body>
</html>
```

The **ajaxComplete()** is called regardless of whether the request is successful or fails, and a complete callback is returned, even for synchronous queries.

The ajaxComplete() method specifies a function to be run when an AJAX request completes.

$(document).ajaxComplete(*function(event,xhr,options)*)

- *event* - contains the event object
- *xhr* - contains the XMLHttpRequest object
- *options* - contains the options used in the AJAX request

This is very useful for error handling. If a request fails but the result is not needed immediately, it is transferred to a cache, and then called later. This is especially useful if you want to keep track of failed requests and retry those that were not successful. A typical scenario is the case of an AJAX request but no data is returned, or the data is returned but not as intended. In this case, you can call ajaxComplete() to get a new set of data and continue with your business logic. A successful AJAX request is not necessarily a

reason to call **ajaxComplete()**. you can still do other stuff while the data is being transferred from the server to the browser. **For example**, you can scroll, pause the video, or show a loading indicator.

21. Describe the benefits of jQuery Ajax techniques.

With the aid of DOM and JavaScript, There is a great advantage of AJAX. Ajax can request and receive data from the server without requiring a page reload. jQuery Ajax methods are a powerful way to make your web applications more responsive. They enable you to take advantage of the power of Ajax by using JavaScript to make your web pages load faster and more efficiently.

By using jQuery Ajax methods, you can take advantage of the power of Ajax by using JavaScript to make your web pages load faster and more efficiently. The following advantages of the AJAX Methods are -

It allows us to eliminate the complete page reload and instead load only a portion of the page.

API that is simple.

Cross-browser compatibility.

POST and GET are supported.

Upload a JSON, XML, HTML, or script file.

27. What is the purpose of JQuery's serialize() method?

https://www.youtube.com/watch?v=wUbt_LCXx2o

It collects form data through name .

The serialize() method is a utility method of the jQuery library that allows you to serialize data from a DOM element and return it in a format that can be used by other libraries.

This is useful when you want to pass data between different libraries or frameworks, or if you want to create a custom API that can be used by other developers. **For example** - if you want to pass JSON data from your application to a third-party API, you can use the serialize() method to serialize the data and then pass it to the API. This way, the API knows how to handle the data, and you don't have to worry about the format of the data.

The serialize() method is also useful when you need to send data between different browsers. **For example** - if you want to send JSON data from one browser to another, you can use the serialize() method to serialize the data and then send it using a custom HTTP request.

If you're not sure what data you need to serialize, there's an easy way to find out: Just type "serialize" into the console. You'll see a list of all the methods that are available on your element.

```
<body>
    <script>
        $(document).ready(function(){
            $('#btnClick').on('click', function(){
                //let x= $('#formData').serialize();
                // It will return name=Sarfraz&age=25&countryDetails=india
                //console.log(x);
                $('.getData').html($('#formData').serialize());
            })
        });
    </script>
    <form id="formData">
```

```
        <input type="text" placeholder="Enter your name" name="name"> <br><br>
        <input type="number" placeholder="Enter your age" name="age"> <br><br>
        <select name="countryDetails">
            <option hidden>Select Your Country</option>
            <option value="india">India</option>
            <option value="japan">Japan</option>
            <option value="dubai">Dubai</option>
        </select> <br><br>
        <input type="button" value="Insert" id="btnClick"> <br>
        <!-- <button id="btnClick">ClickMe</button> -->
        <!-- // It will display the output and disappear but not in input so took input not button -->
    </form>
        <div class="getData"></div>
</body>
```

jQuery AJAX Methods

AJAX is the art of exchanging data with a server, and update parts of a web page - without reloading the whole page.

The following table lists all the jQuery AJAX methods:

Method	Description
$.ajax()	Performs an async AJAX request
$.ajaxPrefilter()	Handle custom Ajax options or modify existing options before each request is sent and before they are processed by $.ajax()
$.ajaxSetup()	Sets the default values for future AJAX requests
$.ajaxTransport()	Creates an object that handles the actual transmission of Ajax data
$.get()	Loads data from a server using an AJAX HTTP GET request
$.getJSON()	Loads JSON-encoded data from a server using a HTTP GET request
$.parseJSON()	Deprecated in version 3.0, use JSON.parse() instead. Takes a well-formed JSON string and returns the resulting JavaScript value
$.getScript()	Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request
$.param()	Creates a serialized representation of an array or object (can be used as URL query string for AJAX requests)
$.post()	Loads data from a server using an AJAX HTTP POST request
ajaxComplete()	Specifies a function to run when the AJAX request completes
ajaxError()	Specifies a function to run when the AJAX request completes with an error
ajaxSend()	Specifies a function to run before the AJAX request is sent
ajaxStart()	Specifies a function to run when the first AJAX request begins
ajaxStop()	Specifies a function to run when all AJAX requests have completed
ajaxSuccess()	Specifies a function to run when an AJAX request completes successfully
load()	Loads data from a server and puts the returned data into the selected element
serialize()	Encodes a set of form elements as a string for submission
serializeArray()	Encodes a set of form elements as an array of names and values

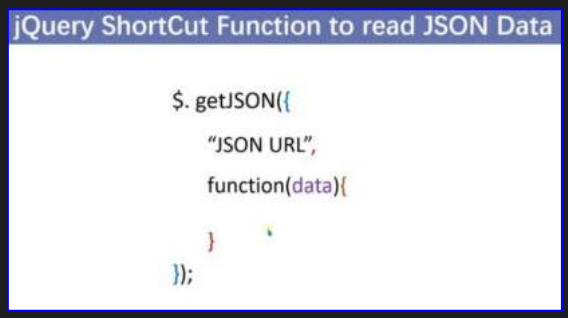
$.ajax method

```
    <script>
       $.ajax({
            type: 'GET',
            url: 'https://jsonplaceholder.typicode.com/users/1',
```

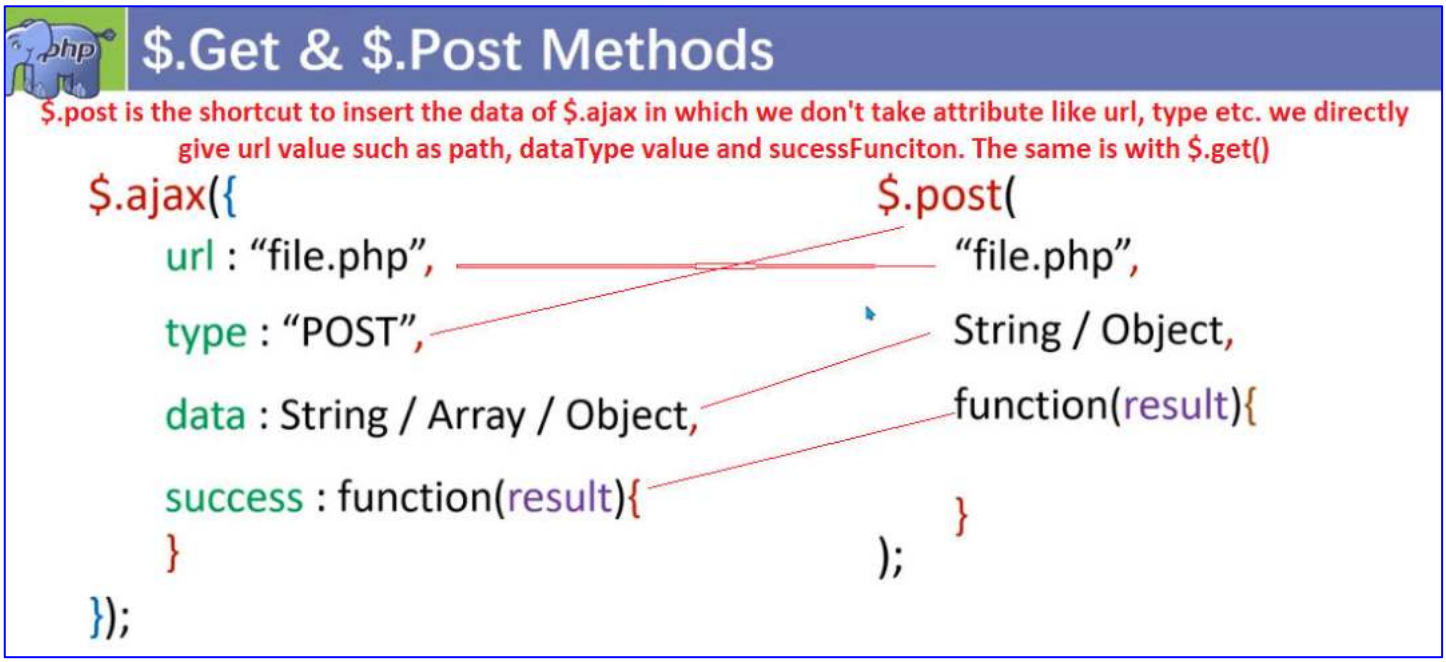
```
            dataType: 'JSON',
            success: function (result) {
                console.log(result);
                // $('#getData').append(result);
                $.each(result, function (key, value) {
                    console.log(key); // to know what we got in key, data will come to result and then value
                    $('#getData').append(key + ':' +value + "<br>");
                });
            },
            error: function () {
                document.write("Unsuccessful");
            }
        });
    </script>
    <div id="getData"></div>
```

$.getJSON()

```
<script>
        $.getJSON(
            'https://jsonplaceholder.typicode.com/users/1',
            function (result){
                $.each(result, function (key, value) {
                    $('#getData').append(key + ':' + value + "<br>");
                });
            },
        );
    </script>
```



$.get and post()



To get api for post

https://www.youtube.com/watch?v=Z9PSICfMbqQ

jQuery Tutorial in Hindi #9 AJAX | How to call POST API from jQuery | Ajax Request from jQuery

jQuery Coding Questions

43. How to perform jQuery AJAX requests?

jQuery provides the *ajax()* method to perform an AJAX (asynchronous HTTP) request.

Syntax: *$.ajax({name:value, name:value, ... })*. The parameters specify one or more values of name-value pairs.

url: this name specifies the URL to send the request to. Default is the current page.
success(result,status,xhr): success callback function which runs when the request succeeds
error(xhr,status,error): A function to run if the request fails.
async: Boolean value that indicates whether the request should be handled asynchronous or not. The default value is true.
complete(xhr,status): A function to run when the request is completed (after success and error functions are handled)
xhr: A function used for creating the *XMLHttpRequest* object

Example:

```
$.ajax({
  url: "resourceURL",
  async: false,
  success: function(result){
    $("div").html(result);
  },
  error: function(xhr){
    alert("An error occured: " + xhr.status + " " + xhr.statusText);
  }
});
```

44. What does the following code do?

$("div#firstDiv, div.firstDiv, ol#items > [name$='firstDiv']")

The given code is an example of getting elements that satisfy multiple selectors at once. The function returns a jQuery object having the results of the query.

The given code does a query to retrieve those * * element with the id value *firstDiv* along with all * * elements that has the class value *firstDiv* and all elements that are children of the *<ol id="items">* element and whose name attribute ends with the string *firstDiv*.

45. Write a jQuery code to create and delete cookies from the browser.

There is no direct way to access the cookies using jquery. We can easily do this with the help of pure javascript. To work with cookies in jQuery, you must install the Dough cookie plugin or any other cookies plugins. The dough is simple to use and offers a number of useful capabilities.

Creating a cookies - `$.dough("cookieName", "cookieValue");`

Reading a cookies - `$.dough("cookieName");`

Deleting a cookies - `$.dough("cookieName", "remove");`

46. Consider the following code that exists in following HTML with the CSS:

```
<div id="expand"></div>

<style>
  div#expand{
    width: 50px;
    height: 50px;
    background-color: gray;
  }
</style>
```

Write jQuery code to animate the #expand div, expanding it from 50 * 50 pixels to 300 * 300 pixels within five seconds.

We can do this by using the animate() function. We first need to have access to the div element which has id value of expand and then apply animate function on the element as follows:

47. What does the following code do?

```
$( "div" ).css( "width", "500px" )
          .add( "p" )
          .css( "background-color", "yellow" );
```

The given code first selects all the `<div>` elements and applies width of 500px to them and adds all the `<p>` elements to the elements selection after which the code can finally change the background color to yellow for all the `<div>` and `<p>` elements

The given code is an example of method chaining in jQuery which is used to accomplish a couple of things in one single instruction.

```
<body>
  <script>
    $(document).ready(function () {
      $(document).ajaxStart(function () {
        $("#wait").css("display", "block");

      });
      $(document).ajaxComplete(function () {
        $("#wait").css("display", "none");
      });
      $("button").click(function () {
        $("#txt").load("demo_ajax_load.asp");
      });
    });
  </script>
  <body>
    <div id="txt"> </div>
```

```
    <div id="wait">loading....</div> <br>
    <button>Change Content</button>
  </body>
```

48. Can you explain the difference between jQuery.get() and jQuery.ajax()?

jQuery.ajax():- It is used to appy AJAX technique in jQuery.

jQuery.get():- Loads data from a server using an AJAX HTTP GET request. It is used under $.ajax() method.

```
      $.get(URL, callbackFu);
//The URL of the file in which data is available and you want to read or get that on the webpage, callback is a function that will to perform any action after getting the file data.
```

EXAMPLE

```
  <script>
    $(document).ready(function(){
      $.get('data.txt', function(data, status){
          $('#sar').append(data);
      });
    });
  </script>
  <body>
    <div id="sar"> </div>
  </body>
```

There are various other pre-built AJAX requests given by jQuery such as:

jQuery.post(): Loads data from a server using an AJAX HTTP POST request. It is used under $.ajax()to insert the data in the server.

```
  <script>
    $(document).ready(function(){
      $.post('data.txt',{name:'sarfraz', age:25}, function(data, status){
          $('#sar').append(data);
      });
    });
  </script>
  <body>
    <div id="sar"> </div>
  </body>
</html>
```

jQuery.getScript(): Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request

jQuery.getJSON() for loading JSON-encoded data from the server using a GET HTTP request.

49. Write a code snippet for preventing the default behavior of the submit button for performing another event.

```
<script>
  $(document).ready(function(){
    $('#submitButton').click(function(event){
      event.preventDefault();

    });
  });
</script>
<form action="" id="onSubmit">
  <label for="">Name </label> : <input type="text">
  <input type="submit" value="Submit"><br>
```

```
  <button id="submitButton">Submit</button>
  <!-- It will stop the default behavior -->
</form>
</body>
```

50. Which of the two lines of code below is more efficient and why?

Would you still favor jQuery in a case when things can be done simply using JavaScript?

51. No. If we can accomplish our goals using simple JavaScript, then we should avoid using jQuery. It is because the jQuery library is always xx kilobytes in size, thus there is no use in wasting bandwidth. And because jQuery is built on top of JavaScript, it has numerous functions that are more complicated than the simple job you can accomplish with JavaScript. jQuery will also load all dependencies. It will have some speed issues as compared to the JavaScript code.

document.getElementById("interviewBit"); **OR** *$("#interviewBit");*

The code document.getElementById("interviewBit"); is more efficient because its the pure JavaScript version.
The reason being jQuery is built on top of JavaScript and internally uses its methods to make DOM manipulation easier. This introduces some performance overhead. We need to remember that jQuery is not always better than pure JavaScript and we need to use it only if it adds advantage to our project.

51. How do you disable elements in Jquery Using "attr"?

```
$('class_or_id_of_element_to_disable').attr('disabled', true);
$('class_or_id_of_element_to_disable').attr('disabled', false);
```

52. Can you write a jQuery code selector that needs to be used for querying all elements whose ID ends with string “IB”?

We can use the following selector:

```
$("[id$='IB']")
```

53. Explain the .promise() method in jQuery.

The *.promise()* method returns a dynamically generated promise that is resolved when **all actions** of a certain type bound to the collection, queued or not, **have ended.**
The method takes two optional arguments:

type - The default type is “fx” which indicates that the returned promise is resolved only when all animations of the selected elements have been completed.
target - If a target object is specified, *.promise()* will attach to promise to that specified object and then return it rather than creating a new one.

54. Write A Jquery Code Snippet To Sort A String Array?

```
<script>
  $(document).ready(function () {
    var arr = ["jQuery", "Interview", "Questions", "By", "InterviewBit"];
    console.log(arr);
    sortedArr = arr.sort();
    console.log(sortedArr);
   $("#elementId").html(sortedArr.join(" , "));  //By , Interview , InterviewBit , Questions , jQuery
  });
</script>
<div id="elementId"></div>
```

The Output will be

`["By","InterviewBit","Interview","jQuery","Questions"]`

55. Consider the below code snippet and assume that there are 5 elements on the page. What is the difference between the start and end times displayed?

```
function getMinsSeconds() {
  var dt = new Date();
  return dt.getMinutes()+":"+dt.getSeconds();
}

$( "input" ).on( "click", function() {
  $( "p" ).append( "Start: " + getMinsSeconds() + "<br />" );
  $( "div" ).each(function( i ) {
        $( this ).fadeOut( 1000 * ( i * 2 ) );
  });
  $( "div" ).promise().done(function() {
        $( "p" ).append( "End: " + getMinsSeconds() + "<br />" );
  });
});
```

For the above code, the difference between the start and end times will be **10 seconds**.

This is because `.promise()` will wait for all `<div>` animations (here, all `fadeOut()` calls) to complete, the last one will complete 10 seconds (i.e. 5 * 2 = 10 seconds) after the `fadeOut()` starts.

56. Can you provide an example of chaining using a code snippet?

Old Code:

```
$(document).ready(function(){
    $('#id').addClass('ib');
    $('#id').css('color', 'blue');
    $('#id').fadeIn('slow');
});
```

New Code after Chaining:

```
$(document).ready(function(){
    $('#id').addClass('ib')
    .css('color', 'blue')
    .fadeIn('slow');
});
```

57. Can you tell the difference between prop() and attr()s?

Both `prop()` and `attr()` can be used to get or set the value of the specified property of an element attribute.

The `attr()` gives the **default value** of a property whereas `prop()` returns its **current value**.

jQuery MCQ

1.The one below is a factory function. That is -
jQuery
$()
onclick()
Queryanalysis
2.Which of the following functions has a high level of overload?
script()
$()
jQuery()
Both jQuery and $()
3.In the jQuery library, which of the following is a single global function defined?
$()
jQuery()
global()
Queryanalysis()
4.Which of the following jQuery method is used to check whether or not the selected elements have the specified class name?
addClass() method
hasClass() method
toggleClass() method
find() method
5.Which of the above jQuery techniques is used to condense the collection of matched components into just one element?
isEqual() method
delegate() method
eq() method
val() method
6.Which of the following is a single global function defined in the jQuery library?
jQuery()
$()
Queryanalysis()
global()
7.Which is code asks for the set of all div elements in a document?

A) *var divs = $(div);*

B) *var divs = jQuery("div");*

C) *var divs = $("div");*

D) *var divs = #("div");*

A
A and B
B and C
A and C
8.Which method operates on the return value of $()?
show()
css()
click()

each()

9.Which method is used for parsing JSON text?

jQuery.each()

jQuery.parseJSON()

jQuery.noconflict()

jQuery.done()

10.Select the correct jQuery code to set the background color of all div elements to red?

$(“div”).css(“background-color”,“red”);

$(“div”).style(“background-color”,“red”);

$(“div”).layout(“background-color”,“red”);

$(“div”).manipulate(“background-color”,“red”);

11.What does the following selector select? $("span.first").

All span elements with class=“first”

The first span element with class=“first”

The first span element with id=“first”

All span elements with id=“first”

12.Which method is used to perform an asynchronous HTTP request?

jQuery.asyncAjax()

jQuery.ajaxSetup()

jQuery.ajaxAsync()

jQuery.ajax()

13.Select the correct jQuery code for making all div elements 100 px high?

$(" ").height(100)

$("div").yPos(100)

$(" ").height=“100”

$("div").height(100)

14.Which method is used to switch between adding/removing one or more classes (for CSS) from selected elements?

toggleClass()

switch()

switchClass()

altClass()

15.What does the following selector select $("div p")?

The first p element inside a div element

All div elements with a p element

All p elements inside a div element

None of the above

16.What does the following selector select $(":disabled")?

All hidden elements

All elements that does not contain the text “disabled”

All disabled input elements

All elements containing the text “disabled”

17.Can the animate() method be used to animate ANY CSS property?

Yes

No

Only properties containing numeric values

All properties except the shorthand properties

18.$.foobar() is equivalent to which of the following below??

javascript.foobar()

document.foobar()

jQuery.foobar()

None of the above

19.What does the following lines of code do?

```
$(document).ready(function() {
        // Some code.
        });
```

Make sure no code is executed till the entire page is fully loaded
Make sure no code is executed till the DOM is fully loaded
Both A and B
Neither A nor B

20.Query.noConflict(true) is used to for?

Freeing up the $ symbol for use by other libraries
Improving compatibility
Removing all jQuery variables from the global scope
All of the Above

21.What is the phenomenon used in the below code?

```
$('#mainList').find('li')
        .width(1000).addClass('selectedElement');
```

Toggling
Event bubbling
Chaining
Animating

22.Select the correct options relating to the :odd and :even filters?

They allow you to determine if a number is odd or even.
They allow you to determine if a specific element is in an odd or even position.
They allow to remove the odd or even elements appropriately.
None of these

23.Identify the difference between .width() and .outerWidth()?

The methods are basically the same. The only difference is that .width() returns a number, whereas outerWidth() a string.
No difference. width() is a shorthand alias for outerWidth()
width() returns the computed width of the element, while outerWidth() returns the width plus all the margins and paddings.
There is no such method called outerWidth().

24.Which of the following code fetches the first span on the page, which has the class 'grey'?

$('span, .grey, :first')
$('first .grey span')
$('span.grey:first')
None of the above

25. What function is used to stop your jQuery for a few milliseconds?

stop()
delay()
slowdown()
pause()

Short Questions and Answers

1) Is jQuery a programming language?

jQuery is not a programming language but a well-written JavaScript code. It is used to traverse documents, event

handling, Ajax interaction, and Animation.

2) Is jQuery replacement of JavaScript?

No, jQuery is not the replacement of JavaScript. jQuery is written on the top of JavaScript, and it is a different library. jQuery is a lightweight JavaScript library which is used to interact with JavaScript and HTML.

3) Is it possible that jQuery HTML work for both HTML and XML document?

No, jQuery HTML only works for HTML document. It doesn't work for XML documents.

4) What is the use of html() method in JQuery?

The jQuery html() method is used to change the entire content of the selected elements. It replaces the selected element content with new contents.

Syntax:

```
$(document).ready(function(){
        $("button").click(function(){
            $("p").html("Hello <b>Javatpoint.com</b>");
        });
    });
```

For complete example: Click here

5) What is the use of css() method in JQuery?

The jQuery CSS() method is used to get (return)or set style properties or values for selected elements. It facilitates you to get one or more style properties. The jQuery CSS() provides two ways:

Return a CSS property

It is used to get the value of a specified CSS property.

```
    $(document).ready(function(){
        $("button").click(function(){
            alert("Background color = " + $("p").css("background-color"));
        });
    });
```

Set a CSS property

This property is used to set a specific value for all matched element.

```
    $(document).ready(function(){
        $("button").click(function(){
            $("p").css("background-color", "violet");
```

```
    });
});
```

6) Is jQuery library used for server scripting or client scripting?

It is a library for client-side Scripting.

7) Is jQuery a W3C standard?

No, jQuery is not a W3C standard.

8) What is the starting point of code execution in jQuery?

$(document).ready() function is the starting point of jQuery code. It is executed when DOM is loaded.

9) What is the basic requirement to start with the jQuery?

You need refer to its library to start with jQuery. You can download the latest version of jQuery from jQuery.com.

10) Can you use any other name in place of $ (dollar sign) in jQuery?

Yes, instead of $ (dollar sign) we can use jQuery as a function name. For example:

```
jQuery(document).ready(function() {
jQuery("p").css("background-color", "pink");
});
```

11) What is the difference between find and children methods?

Find method is used to find all levels down the DOM tree while children method is used to find single level down the DOM tree.

12) How can you use a jQuery library in your project?

You can use a jQuery library in the ASP.Net project from downloading the latest jQuery library from jQuery.com and include the references to the jQuery library file in your HTML/PHP/JSP/Aspx page.

```
<script src="_scripts/jQuery-1.2.6.js" type="text/javascript"></script>
<script language="javascript">
$(document).ready(function() {
alert('test');
});
</script>
```

13) What is a use of jQuery filter?

: jQuery filter is used to filter the specific values from the object. It filters the result of your original query into specific elements.

14) What is the difference between the ID selector and class selector in jQuery?

ID selector and class selector are the same as they are in CSS. ID selector uses ID while class selector uses a class to select elements.

You use an ID selector to select just one element. If you want to select a group of elements, having the same CSS class, use class selector.

15) What is the use of serialize() method in JQuery?

The jQuery serialize() method is used to create a text string in standard URL-encoded notation. It serializes the form values so that its serialized values can be used in the URL query string while making an AJAX request.

Syntax:

```
$(document).ready(function(){
    $("button").click(function(){
        $("div").text($("form").serialize());
    });
});
```

16) What is the use of val() method in JQuery?

The jQuery val() method is used:

- To get the current value of the first element in the set of matched elements.
- To set the value of every matched element.

Syntax:

```
$(selector).val()
```

For complete example: Click here

17) How to add and remove CSS classes to an element using jQuery?

You can use addclass() jQuery method to add CSS class to an element and removeclass() jQuery method to remove CSS class from an element.

18) Can you write a jQuery code to select all links inside the paragraph?

Yes. You can use <a> tag nested inside paragraph <p> tag to select all links. For example:

```
<!DOCTYPE html>
<html>
```

```
<head>
 <title>jQuery Example</title>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
 </script>
 <script type="text/javascript" language="javascript">
$(document).ready(function()
{
   $("p a").attr("href", "https://www.javatpoint.com");
});
 </script>
 </head>
<body>
   <p><a>Learn JavaScript</a></p>
   <p><a>Learn jQuery</a></p>
</body>
</html>
```

19) What is the difference between prop and attr?

attr(): It gets the value of an attribute for the first element in the set of matched element.

prop(): it gets the value of a property for the first element in the set of matched elements. It is introduced in jQuery 1.6.

20) What is the use of the animate() method in jQuery?

The animate function is used to apply the custom animation effect to elements. Syntax:

```
$(selector).animate({params}, [duration], [easing], [callback])
```

- "param" defines the CSS properties on which you want to apply the animation.
- "duration" specify how long the animation run. It can be one of the following values: "slow," "fast," "normal" or milliseconds
- "easing" is the string which specifies the function for the transition.
- "callback" is the function which we want to run once the animation effect is complete.

21) How can you disable jQuery animation?

By using jQuery property "jQuery.fx.off" and setting it to true, you can disable jQuery animation.

22) Does jQuery HTML work for both HTML and XML documents ?

No, JQuery HTML doesn't work with XML document. It only works for HTML documents.

23) Difference between .empty(), .remove() and, .detach() in Jquery ?

- jQuery empty() Method: The empty() method in jQuery is used to remove all child nodes and its content for the selected elements.
- jQuery remove() Method: The remove() method in JQuery is used to remove all the selected elements including all the text. This method also remove data and all the events of the selected elements.
- jQuery detach() Method: The detach() method in jQuery is used to remove the selected elements from the DOM tree including its all text and child nodes but it keeps the data and the events. Document Object Model (DOM) is a World Wide Web Consortium standard. This defines for accessing elements in the DOM tree.
- **Note:** The **remove()** method is faster than **empty()** or **detach()** method.

24) What are the various ajax functions available in Jquery ?

Ajax allows the user to exchange data with a server and update parts of a page without reloading the entire page. Some of the functions of ajax are as follows:

- jQuery ajaxSetup() Method: The ajaxSetup() method is used to set the default values for future AJAX requests.
- jQuery ajax() Method: The ajax() method is used to perform an AJAX request or asynchronous HTTP request.
- jQuery getScript() Method: The getScript() method is used to run a JavaScript using AJAX HTTP GET request.
- jQuery getJSON() Method: The getJSON() method fetches JSON-encoded data from the server using a GET HTTP request.

25) **Mention the compatible operating systems with jQuery.**

- Mac
- Windows
- Linux

26) **How to include the jQuery library in the ASP.Net project ?**

- Download the jQuery library from jQuery.com
- include that reference in the asp.net page.

27) **Explain bind() vs live() vs delegate() methods in jQuery.**

The bind() method does not attach the events to those elements that are added once DOM is loaded. whereas live() and delegate() methods attach events to the future elements also.

The difference between live() and delegate() methods is that the live() function does not work in chaining. It will work only on a selector or an element while delegate() method will work in chaining.

28) **Write the command that gives the version of jQuery ?**

The command $.ui.version returns jQuery UI version.

29) What is the use of param() method in JQuery ?

The **param() method** in jQuery is used to create a serialized representation of an object.

30) **What are the browser related issues for jQuery ?**

Compatibility with the Browsers of jQuery plugin is an issue.

31) **Difference between jQuery.size() and jQuery.length ?**

jQuery.size() method is used to find the number of elements matched by the given selector and jQuery.length property is used to count number of the elements of the jQuery object. jQuery.length property is preferred because it does not have the overhead of a function call.

32) **Can we call C# code behind using jQuery ?**

Yes, we can call C# code from jQuery because it supports .net application.

33) **Is jQuery required for bootstrap ?**

Bootstrap uses jQuery for JavaScript plugins (like models, tooltips, etc). If just CSS part of Bootstrap is used, you don't need jQuery.

34) **Can a jQuery library be used for server scripting ?**

jQuery is designed with the functionality for client-side scripting. jQuery is not compatible with server-side scripting.

35) **Why jQuery is better than JavaScript ?**

jQuery is a library used for developing Ajax application and it helps to write the code clean and concise. It also handles events, animation, and Ajax support applications.

36) **What is QUnit ?**

QUnit is a powerful, easy-to-use JavaScript unit testing framework. It is used by the jQuery, jQuery UI, and jQuery Mobile projects and is capable of testing any generic JavaScript code.

37) **What is the method used to define the specific character in place of $ sign ?**

'NoConflict' method is used to reference a jQuery and save it in a variable. That variable can be used instead of Sign.

38) **Where jQuery code is getting executed ?**

Client side (Browsers) are used to execute the jQuery codes.

39) **Explain and contrast the usage of event.preventDefault() and event.stopPropagation() method.**

The preventDefault() Method in jQuery is used to stop the default action of selected element to occur. It is used to check whether preventDefault() method is called for the selected element or not. The event.stopPropagation() method is an inbuilt method in jQuery which is used to stop the windows propagation. In the DOM tree, when setting an event with the child element and the parent element as well then if you hit on the child element event it will call both child and the parent element as well.

40) **What is the proper way in jQuery to remove an element from the DOM before its Promise is resolved ?**

A returned Promise in jQuery is connected to a delayed object stored on the data() for an element. Since the remove() method removes the element's data still because of the element itself. It will prevent any of the element's unresolved promises from resolving.

Therefore, it is necessary to remove an element from the DOM before its Promise is resolved. Use detach() method instead and follow with removeData() method when resolution.

41) **How method can be called inside code behind using jQuery ?**

$.ajax can be called and by declaring Web Method inside code behind using jQuery.

42) **How to work with parent(), children() and siblings() methods in jQuery ?**

The parent() method returns the parent of the chosen element by calling the jQuery parent() method. The siblings() method returns all the siblings of given HTML elements.

43) **How to find browser and browser version in jQuery ?**

Using $.browser property of jQuery returns the browser info.

Using $.browser isn't suggested by jQuery itself, thus this feature has been moved to the jQuery.migrate plugin that is accessible for downloading if the user desires.

1.

- It is a vulnerable practice to use the same. Use it only if needed.
- It is continuously higher to not use browser-specific codes.

44) **What are all the ways to include jQuery in a page ?**

Following are the ways to incorporate jQuery during a page:

- Local copy inside script tag
- Remote copy of jQuery.com
- Remote copy of Ajax API
- Local copy of script manager control
- Embedded script using client script object

45) **How can we debug jQuery ?**

There are two ways to debug jQuery code:

Debugger keyword

- Add the debugger to the line from wherever we have to start out debugging and run Visual Studio in debug mode with F5 function key.
- Insert a breakpoint after attaching the process.

46) **How to check the data type of any variable in jQuery ?**

Use **$.type(Object)** method to get the data-type of object.

47) **How to check if an element is hidden in jQuery?**

To check if an element is **hidden** or **not**, jQuery **:hidden** selector can be used. **.toggle()** function is used to toggle the visibility of an element.

```
$(element).is(":hidden");
```

The End

1. What is React?

React is a front-end and open-source JavaScript library which is used to make user interfaces as fast as possible. In React, it renders the only page or content we make changes with the help of virtual DOM and feels like the entire website is on a single page . It is helpful in building **complex and reusable** user interface(UI) components of mobile and web applications as it follows the component-based approach.

Form Details

We need onChange event to get form input value and use wherever you want. First e.target.value (as soon as we type inside the input area e will return complete DOM , target on which we are targeting like ab so b will be targeted and value to get the entire value that written inside input area.So, the given code will return the image value

```
    function get(e){
        console.log(e.target.value);
    }
             <input type="text" placeholder='Enter Your Name' onChange={get}/>
```

Now, we will use useState to store or print the data we got from input area.

Print? Means if print value is true

Print? <h1> {data} </h1> agar print value true hai to show h1 data

: mean else or otherwise

Print? <h1> {data} </h1> : null means sahi hai to h1 warna kuch nahi

```
import React, { useState } from 'react';
function State() {
    const[data, setData]=useState("");
    const[print, setPrint]=useState(false);
    function stop(e){
        e.preventDefault();}
    function get(e){
        console.log(e.target.value);
        setData(e.target.value);}
    let style= {
        color:"blue",
        fontSize: 30,
        textAlign:"center"}
    return (
        <>
            <h1 style={style}>How to create a form in React & Validate it</h1>
            <h3 style={{textAlign:"center"}}>Data Received From Input Area </h3>
            {/* <h5 style={{textAlign:"center"}}>{data}</h5> */}
            {/* onwriting in the input, it will show */}
            <h3 style={{textAlign:"center"}}>{print?<h3>{data}</h3>:null}</h3>
            {/* on button click pe dikhega */}
            <form onSubmit={stop} style={style}>
             <input type="text" placeholder='Enter Your Name' onChange={get}/> <br/>
             <button onClick={()=>{setPrint(true)}}>Get The Data On Click</button>
             {/* jo dikha raha nahi dikhane ke liye */}
            </form>
        </>
    )
}
```

Hide and Show

```jsx
import logo from './logo.svg';
import './App.css';
import React from 'react'
function App() {
  const [status,setStatus]=React.useState(true)
  return (
    <div className="App">
      {
        status? <h1>Hello World !</h1>:null
      }
      <button onClick={()=>setStatus(false)} >Hide</button>
      <button  onClick={()=>setStatus(false)}>Show</button>

    </div>
  );
}

export default App;
```

What are React Hooks?

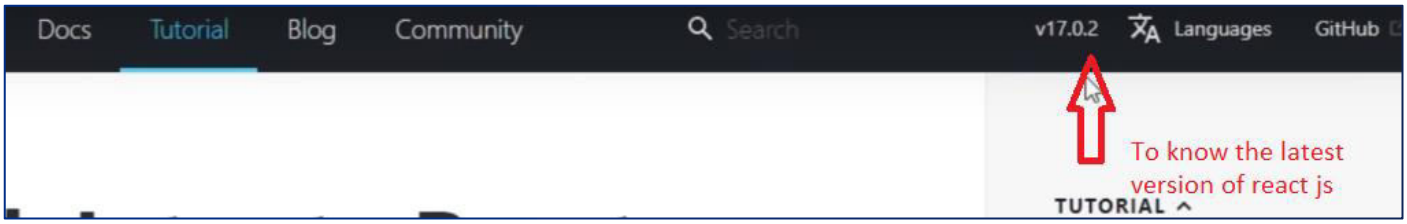
Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class.

We know that when we use class components we have various inbuilt features such as state, life-cycle-methods, pure component, ref etc. But, these features are not available in functional components by default and to use them we use Hooks such as useState to use state in functional component, useEffect to use life-cycle-methods in functional component, use memo for pure component and useRef for ref etc.

Q. UI(User Interface) and Single Page Application(SPA).

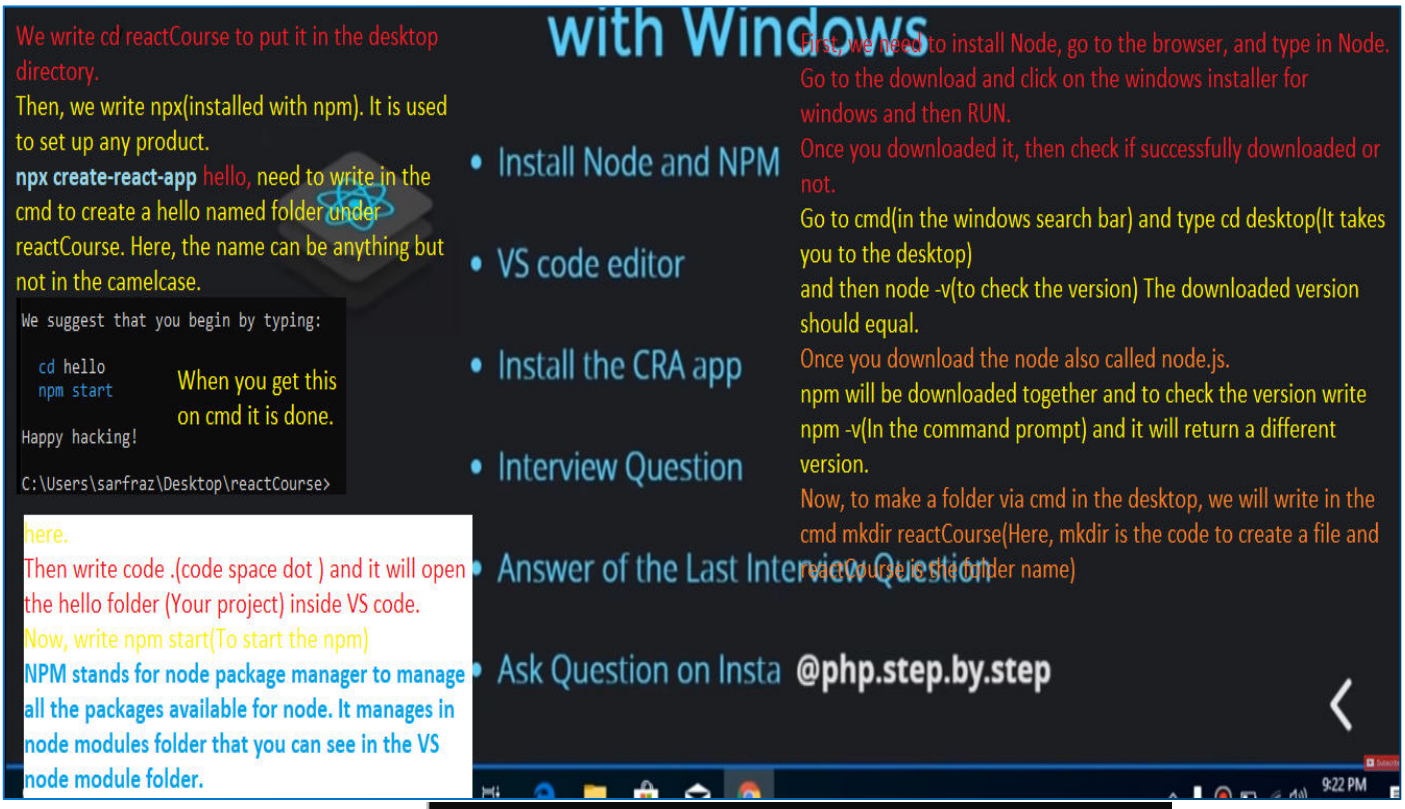
UI means the appearance of the webpage. This entire page is a UI. So, react builds a fast UI(User Interface or the things that a user looks at on the web page.
Single Page Application means https://reactjs.org/ . A website will not reload the entire page if you click on any content on the contrary it will only change the content you click on and feels like it is a single-page application. In https://reactjs.org/ []reload control page(Ctrl + R)] located on the left side of the URL bar, will not move as it is a single-page application.

Q. History and Version of React.js



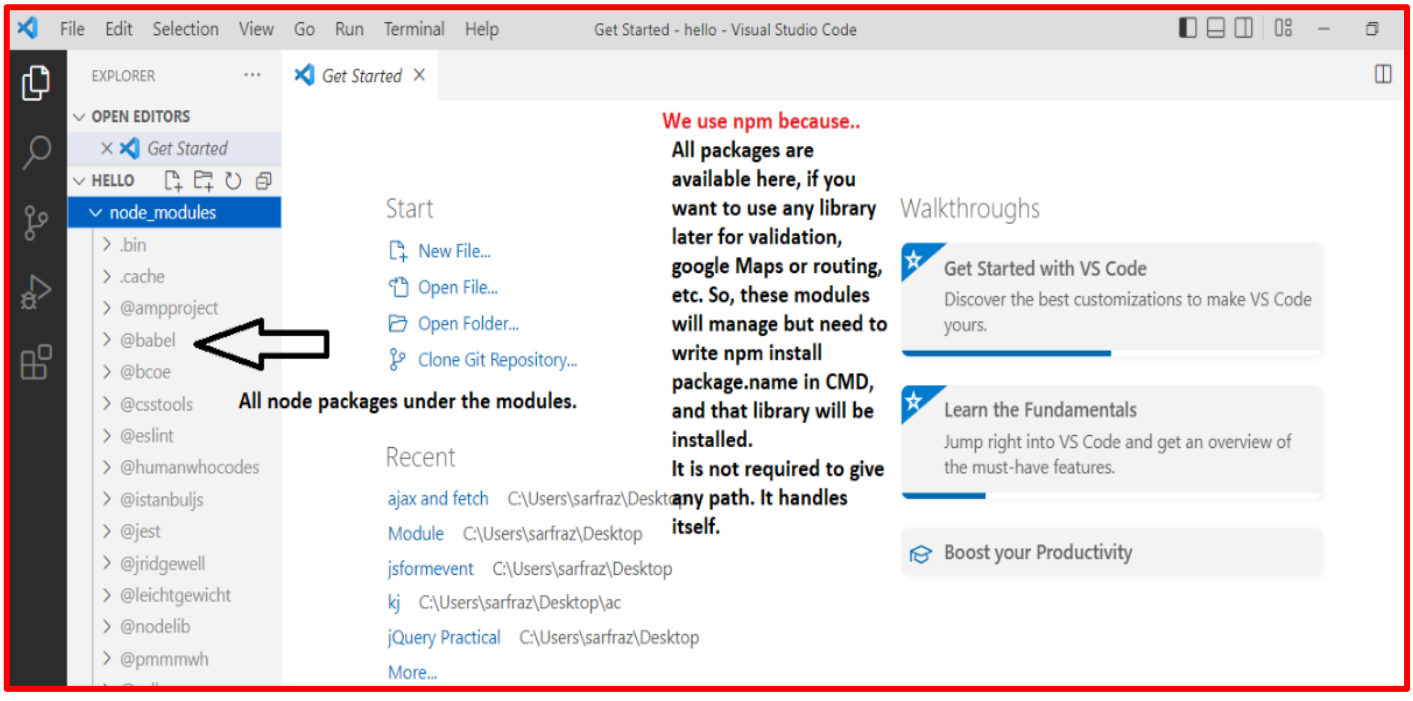
Q. Define NPM of React.js and How to make first app on React.js or How to install the NPM?

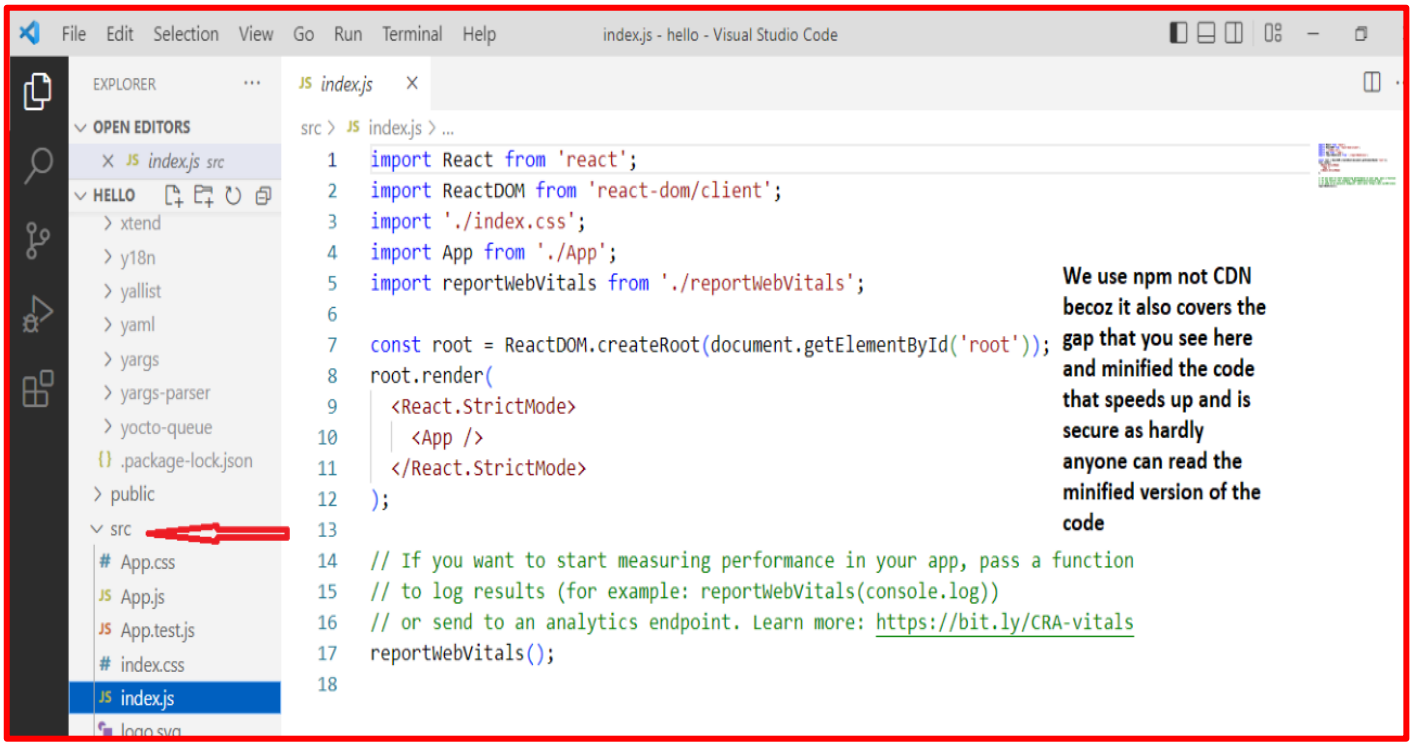
NPX - node package execute that runs the NPM package that you want to use from NPM registry.

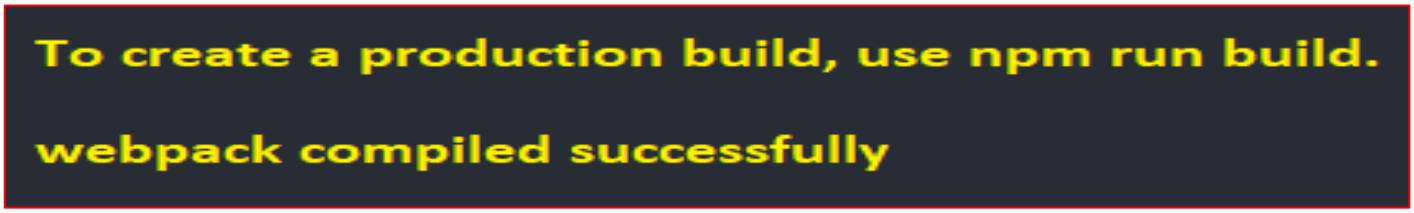



```
C:\Users\pione>cd OneDrive

C:\Users\pione\OneDrive>cd Desktop

C:\Users\pione\OneDrive\Desktop>mkdir reactProject
```

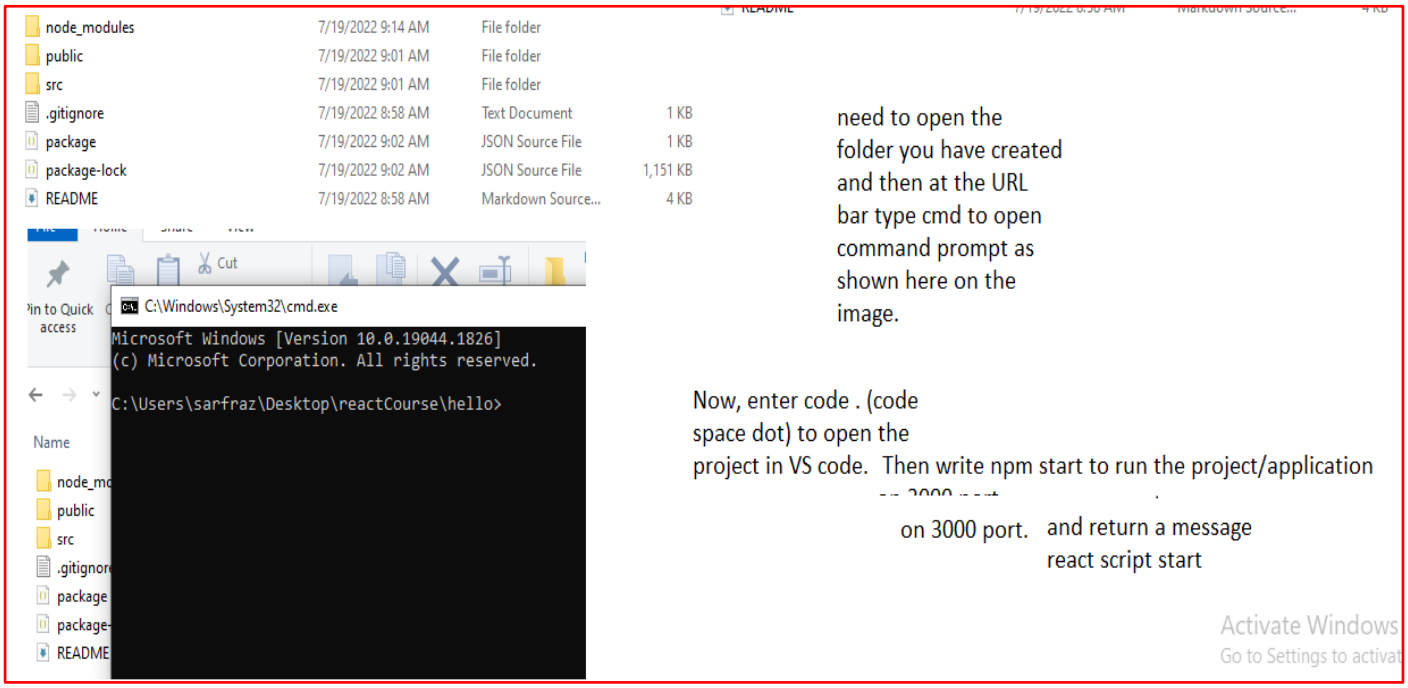
Q. Where all the packages available and why do we use NPM Or code to install any library?







Q. How to open the project in VS code?



2. Why use React instead of other frameworks, like Angular?

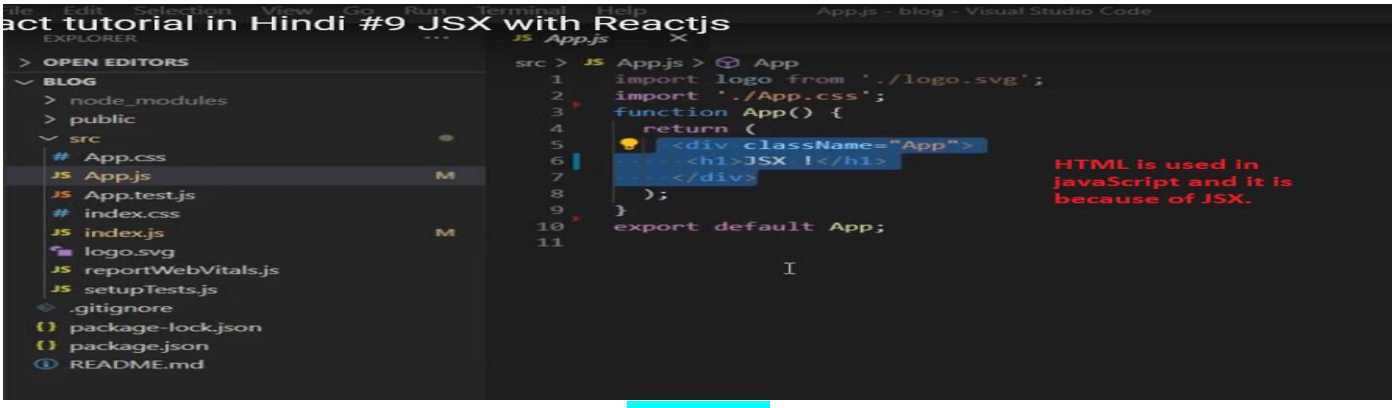
1. We can make UI very fast and it is on high demand due to its fast speed. It is easy to learn and it is famous for its simplicity and high scalability.
2. It is managed by Facebook. So, it will last long in the market.
3. We can create Mobile Apps as well by learning React.native.
4. We use components to create web page in react which can be reused.
5. Since components are reused, it enhances the performance and easy to manage and debug.
6. React has dedicated tools for debugging released by Facebook called chrome extension which makes the process of debugging very fast.

7. In React, data flows in one direction called unidirectional data flow or one way data binding, it becomes very easy to find where the error is. So, easy to debug.
8. We can easily create dynamic web pages.
.

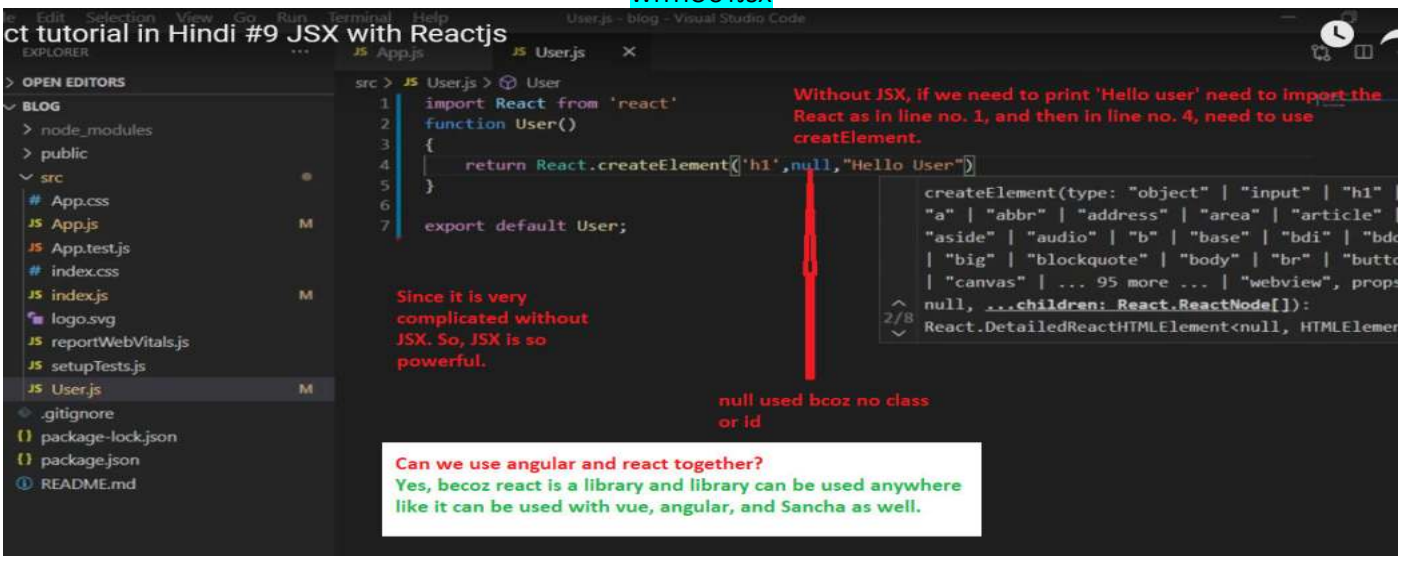
3. What are the features/advantages of React?

1. JSX- It stands for JavaScript XML which is a syntax extension to JavaScript and it is used to write JavaScript and HTML together without script tag. We can use React without JSX but the code will be very lengthy.

```
1. class App extends React.Component {
2.  render() {
3.   return(
4.     <div>
5.       <h1>Hello JavaTpoint</h1>
6.      </div>
7.    )
8.  }
9. }
```

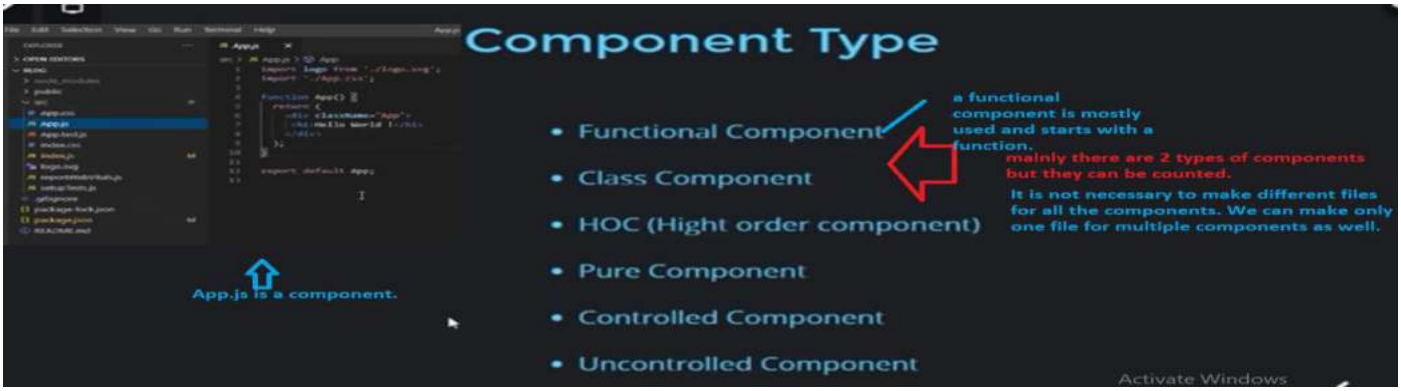


WITHOUTJSX




```
import React from "react";
function State(){
   return React.createElement("h5", null, "hello user")

}
export default State;
```

2. COMPONENTS- It is piece of code that we can reuse such as a function but it is more powerful than function like header, footer are two components.



3. VIRTUAL DOM Virtual DOM is a light-weight JavaScript object and is a copy of actual DOM. We know that In react, everything is a component based. Whenever we make any web page in React, we use different components like header, body and footer. Here, React renders these components to the actual DOM which display on the screen by reflecting and also create a virtual DOM and when we make any changes it again create a virtual DOM and compare from the previous virtual DOM and finalize where and what changes have been made and once it is done the real DOM is updated with only the things that actually changed. React.js is so fast due to Virtual DOM.
So, this is how React works. (**Ho`w React Works)**

Virtual DOM is a part of a React whereas Actual DOM is a part of JavaScript.
React keeps a copy or representation of the DOM tree or copy of actual DOM in the memory which keeps on changing with the codes we write and it finally renders when we have to transfer the codes(changes) to actual DOM. The process of transferring the data or the changes we have made from virtual DOM to actual DOM is called Re-conciliation. Here, minimum changes are done because it takes place at time of render.
There are some rules of re-conciliation
It compares previous virtually DOM tree to current virtually DOM tree and if it finds any changes in current DOM tree then previous DOM tree is removed and updated with the new DOM tree or the changes we have made.
If previous and current DOM tree of any element are same then it will check the attribute or child elements and whenever changes are made will be updated with the new changes.

1. Div span -- span will be updated and div will be discarded
2. DIV>span DIV>b Here child b will be updated

React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects.

4. ONE WAY DATA BINDING-

5. IMPROVE SEO PERFORMANCE- As it has less problems in comparison to the JavaScript codes

6. HIGH PERFORMANCE-

4. **Can we use angular and react together?**

Yes, because React is a library and library can be used anywhere like with Vue, Angular, and Sancha.

5. **How is React different from Angular?**
MVC is a design pattern that separate an application into three logical components and those components are Model, View and Controller.

1. React uses a view of MVC but Angular is made complete of MVC
2. React uses virtual DOM but Angular uses actual DOM
3. React has one-way Data binding but Angular has two-way data binding.
4. React uses JSX but Angular uses Type Script which is the super set of JavaScript.
5. React has server side rendering but Angular has client side rendering.

6. What is the difference between the ES6 and ES5 standards?

These are the few instances where ES6 syntax has changed from ES5 syntax:

Kindly refer the link to get the answer.

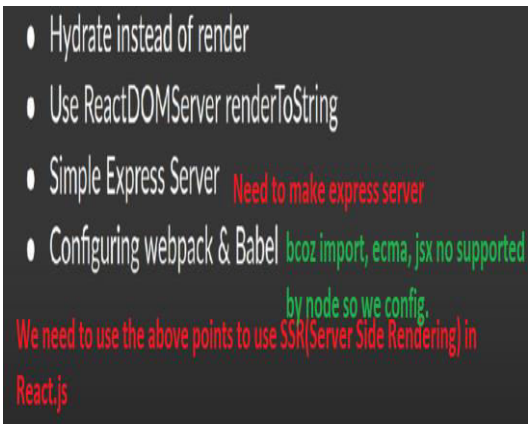
https://www.javatpoint.com/react-interview-questions

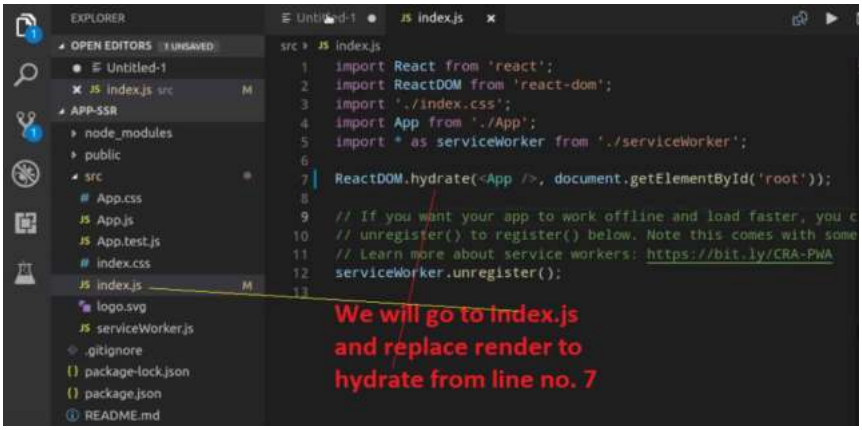
6. Server Side Rendering and Client Side Rendering.(difference)

Writing HTML in server side pages script like PHP, ASP.NET, JAVA etc is the example of server side rendering because HTML renders through the server side but writing HTML in client side script like React.js, Angular.js and Vue.js etc is the example of client side rendering because HTML generates from front end like JavaScript.

Server Side Rendering- It is a technique by which we send client side HTML code to the server side and a file of the code is created on server side and then returned to the client side which helps to improve the SEO because Search Engine can understand server side code not the client side code and when the code will come from the server side, it will load quickly on the browser because it will not show the blank page for the time being as you can see on the client side scripting language. Server side script will quickly load or render the page like HTML content first and then images and user will not have to wait for 3 to 4 seconds.

Ways to use SSR in React.js





npm install express and then npm install babel

https://www.youtube.com/watch?v=RWHQpPly28I

6. Why can’ t browsers read JSX?

Browsers can only read JavaScript objects but JSX in not a regular JavaScript object. Thus to enable a browser to read JSX, first, we need to transform JSX file into a JavaScript object using JSX transformers like Babel and then pass it to the browser.

7. What are the limitations of React?

The few limitations of React are as given below:

React is just a library. It is not a complete framework.

It has a huge library which takes time to understand.

It may be difficult for the new programmers to understand code.

React uses JSX where in both HTML and JavaScript are written together so it becomes a bit confusing to understand the coding for the beginner and JSX is not understood by the browsers so we need to convert this into JavaScript object using JSX transformer Babel. We also need to learn about Web-pack used to build the application and NPM to run the React application. So, we need to learn so many things to resolve the errors if occur.

React does not have things inbuilt like Angular, we need to include other external files to perform routing but not in Angular.

8. Difference between Real DOM and Virtual DOM.

Real DOM	Virtual DOM
1. It updates slow.	1. It updates faster.
2. Can directly update HTML.	2. Can't directly update HTML.
3. Creates a new DOM if element updates.	3. Updates the JSX if element updates.
4. DOM manipulation is very expensive.	4. Virtual DOM manipulation is not expensive.
5. Too much of memory wastage.	05. No memory wastage.

9. How do you create a React app?

Ans:- Get the answer from notes

10. What is state & useState() in React?

state is just like an object that store data like variable in JavaScript. We use state in React to store the value or data. We use state not variable in React like in functional component because if we change the value of variable in functional component it will not update and this is why we use state instead of variable in React components.

useState() is a hook of React used to manage or store the data available in the React. useState is assigned on the top of any component under the curly braces
and written as useState() in camel-case and then parenthesis which has initial data and it returns an array of two elements first is state variable you can give any name like data and second element will be the updated function written as setData or updatedData. Here, if we write 0 in the parenthesis then 0 will be the current data of data.

In order to use useState, we need to define the use state 2 ways.

Syntax of useState

Var/let/const[data, setData] = useState(0);

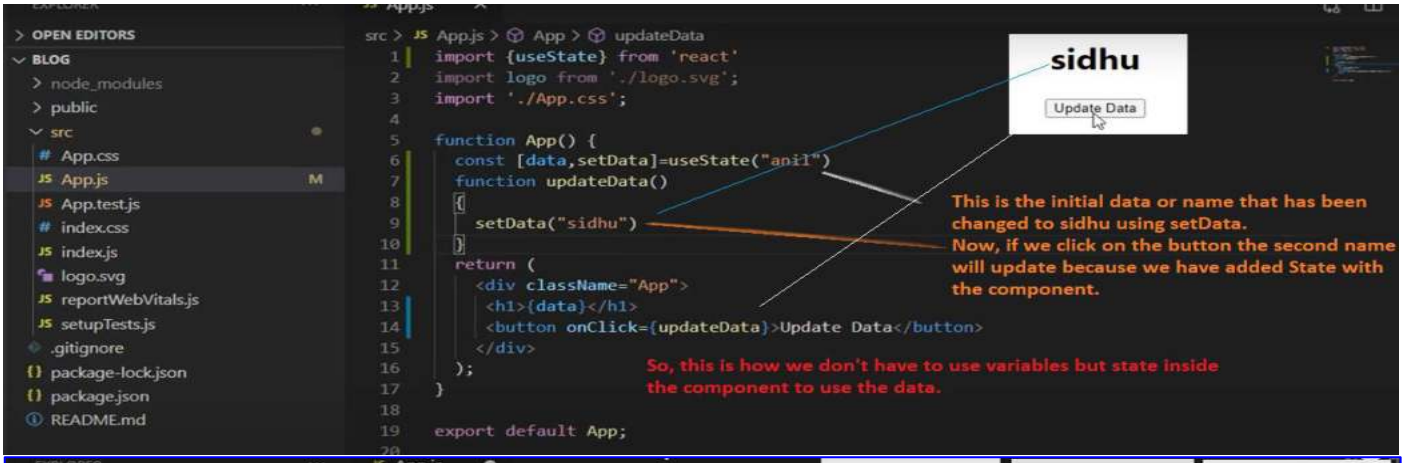
First way to define it .

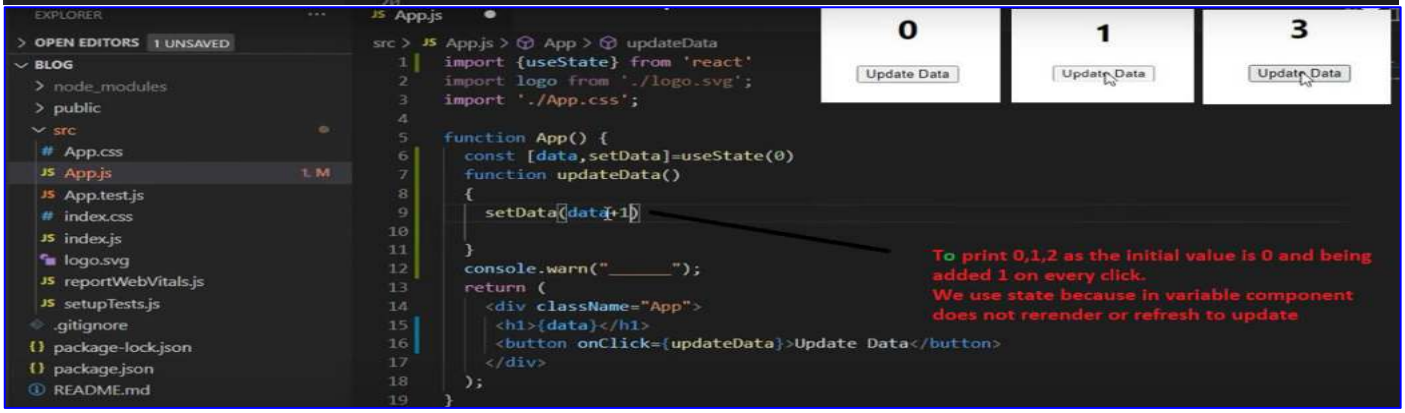
Var/let/const[data, setData] = React.useState(0);

Second way to define useState

Import React, {useState} from 'react'

Var/let/const[data, setData] = useState(0);





11. What is an event in React and how to use event in React(refer note)?

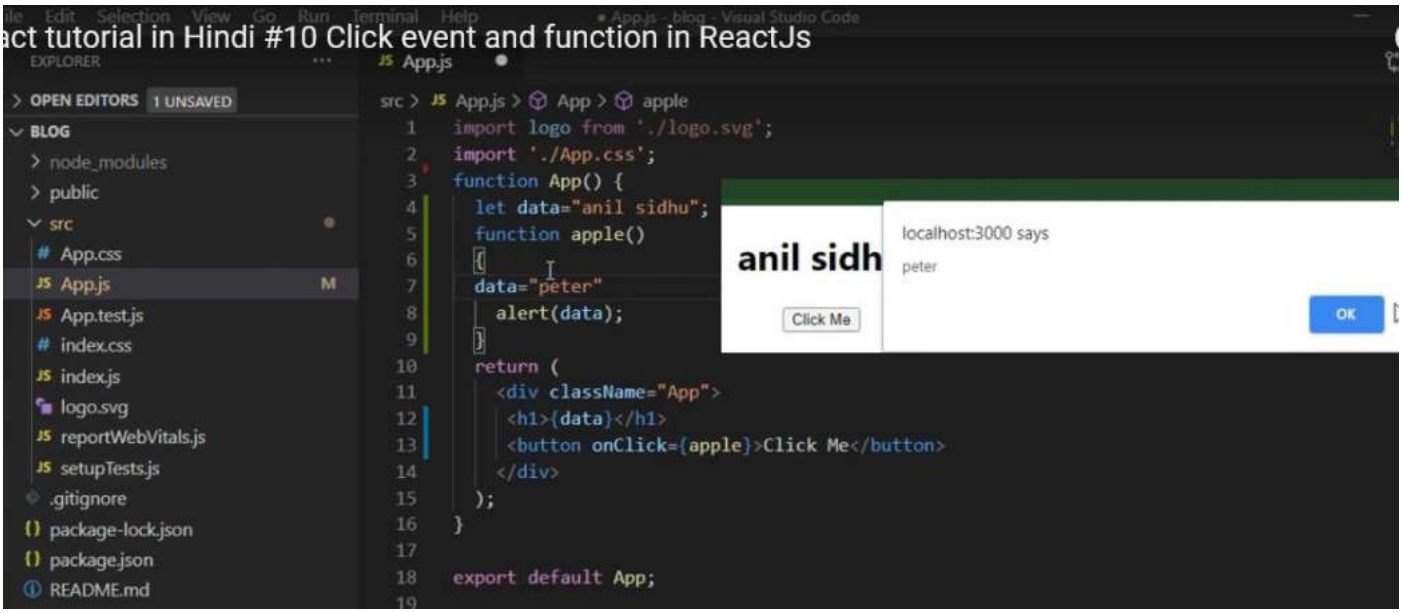
An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

We can update the state on clicking on the button using React event.

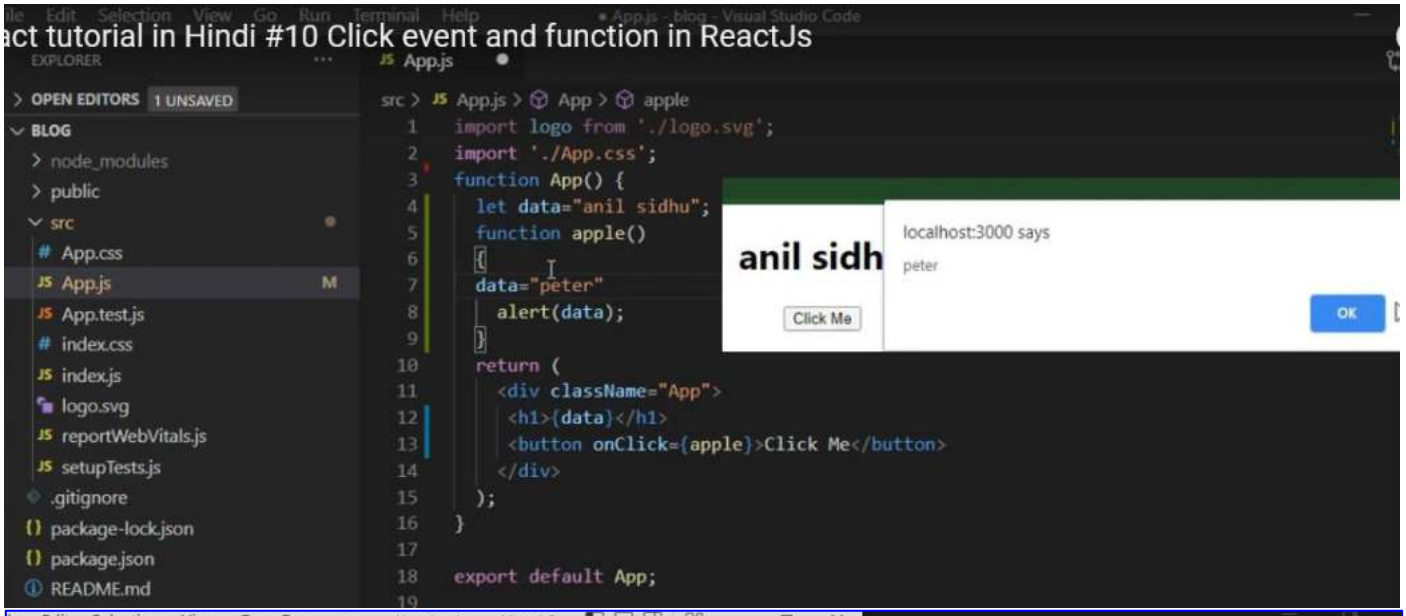
Here, event name will be written in camelCase and function name will be also be written in camelcase under curly braces.

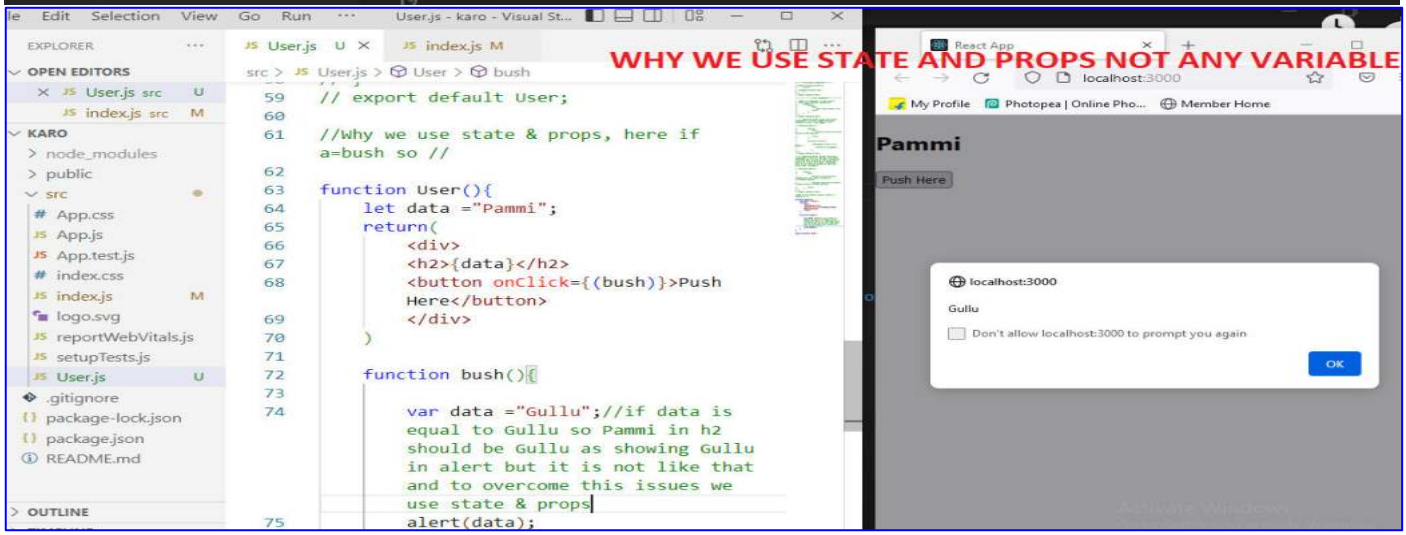
We need to take a button like <button onClick ={updateData}> Click on me </button>

EXAMPLE



WHY PROPES OR STATES NOT VARIABLE





11. What is the purpose of render() in React.

The render is a method which is used to display the specified HTML codes inside the specified HTML element. We display the specified HTML code inside the specified HTML element using a function called ReactDOM.render(). It takes three arguments but the third one is not necessary. First is element, container[, callback]

- It is rendered at container available at Public>index.html>body section>   (This is where our React application will be rendered) and the render method is available at index.js inside the React project.
- Here is the link of practical.
- https://www.w3schools.com/REACT/react_render.asp

Points to Note:

- Each render() function contains a return statement.
- The return statement can have only one parent HTML tag. Like all elements

The Root Node

The root node is the HTML element where you want to display the result and it is like container.
The result is displayed in the element:

```
<body>
  <div id="root"> Root Node(yahi pe display hoga) </div>
</body>
```

12. How can you embed two or more components into one?

You can embed two or more components into the following way:

```
1.  import React from 'react'
2.
3.  class App extends React.Component {
4.     render (){
5.        return (
6.           <h1>Hello World</h1>
7.        )
8.     }
9.  }
10.
11. class Example extends React.Component {
12.    render (){
13.       return (
14.          <h1>Hello JavaTpoint</h1>
15.       )
16.    }
17. }
18. export default App
```

13. What is Props and state in React and how is it used?

Props is the shorthand for Properties in React. It is used to transfer the data from parent to child component. In order to pass the data from parent to child, we have to pass the data in parent component that you want to send like
<User name ={" sarfraz" }/>in parent component and to receive it in child component we will go to child component and write props under the parenthesis of User component like function User(props) and then write console.log(props.name)(here we can take any name instead of props) or under div like

```
<div>
      <h2> {props.name} </h2>
</div>
```

States are the heard of React components. States are the source of data and must be kept as simple as possible. Basically, states are the objects which determine components rendering and behavior. They are accessed via **this.state()**.

14. Difference between props and state.

1. State can be updated/changed/mutable but props are not.
2. State does not allow to pass the data from parent to child or it holds the data in the component itself but props does allow to transfer the data from parent to child as arguments.
3. State can not have stateless component(The component does not have state) because state require state component but props can have stateless component because we do not require state in props.
4. Props make component reusable but state does not.
5. Props are external and controlled by whatever renders the component whereas The state is internal and controlled by the React component itself.

15. Differentiate between states and props.

Conditions	State	Props
1. Receive initial value from parent component	Yes	Yes
2. Parent component can change value	No	Yes
3. Set default values inside component	Yes	Yes
4. Changes inside component	Yes	No
5. Set initial value for child components	Yes	Yes
6. Changes inside child components	No	Yes

15. How can you update the State of a functional and class component?

We can update the State of a component using this.setState() method

Refer the notes.

16. Differentiate between stateless and stateful components.

STATEFUL	STATELESS
Stateful components are those component in which we use or define state whether it is a class or functional component.	Do not use state in stateless component.
Stateful component can be reused. Stateful is also known as class component.	Can not be reused because no state Stateless is also know as functional component.
Stateful component can work with all life-cycle methods.	Can not work with all life cycle methods.
Stateful component is a bit confusing.	Not that confusing.

17. What is an arrow function and how is it used in React?

https://www.youtube.com/watch?v=tmt7lf2nCBY&list=PL8p2I9GklV47BCAjiCtuV_liN9IwAl8pM&index=11

An arrow function is used to write an expression in React. It allows users to manually bind components easily.

Consider the following example:

//Here MyInput onChange equal to, we are connecting to this.handleChange.bind which is longer version of how you can connect it to an event handler

```
render() {
return(
<MyInput onChange={this.handleChange.bind(this) } />
```

```
);
}
//Making use of the arrow function and this is the shorter version to connect it to the event handler.
render() {
return(
<MyInput onChange={ (e) => this.handleOnChange(e) } />
);
}
```

An arrow function is a short way of writing a function to React. In another word, It makes the syntax of function shorter.

The arrow function is the best way to bind the events and also pass the parameters to the event handlers easily.
In arrow function, it is really not required to bind "this" keyword as it is already bind with arrow function.

In React, we need to bind(use) events so that "this" keyword would not return an undefined but when we use an arrow function, it is really not required to bind "this" keyword as it will update the state itself if we use arrow function but in other methods such as constructor and render methods in class component we need to bind the this keyword.
When we need to update any state on button click or with event handler in class and functional component, we need to use arrow function so that we don't get undefined error else need to bind with other methods. We can bind(or use) this keyword in constructor method and render method to change the state or value from set to updateSet on button click or event hand
ler. It is best to use arrow function to update the state on click in functional component.
It is unnecessary to bind 'this' inside the constructor when using an arrow function. This prevents bugs caused by the use of 'this' in React callbacks.

```
 <script>
              var name = "demo";
              function abc(){
                 return this.name    // This will take parent object reference
              }
              const aa = ()=>{ return this.name;}

              obj1 = {
                 name: 'Sarfraz',
                 abc,
                 aa
              }
              console.log(obj1.abc());
              console.log(obj1.aa());
              // It will not print sarfraz as arrow function does not have this. if we remove name like return this it will
return window because window is global variable and if we add any variable like var name = "demo" it will be
added in this context  and return demo if we add return this.name So, we will get demo from arrow function
              //and this why in react we use bind function for normal function and nothing for arrow function
   </script>
```

17. What is binding event handlers in React.

Binding event handlers in React tells that whenever any event is triggered the function bind(associated) with that event will be called.

18. What are the different phases of React component' s lifecycle?

There are three different phases of React component's lifecycle:

1. Mounting- It means creating a new method or showing something on the click of a button in DOM
2. Updating- It means passing the props or updating the state in a component.
3. Unmounting It means deleting any method or hiding something on button click from DOM.

19. Explain the lifecycle methods of React components in detail.

Some of the most important lifecycle methods are:

Mounting:-

Constructor () - It is first method of life-cycle that calls itself even before the render method in class component.The constructor method is used in class component and as soon as we call the class, this constructor method calls itself. So, constructor method is the first method that calls itself when we call the class component and this is why it is very important and since it calls first we define state in it.

We write constructor method first in class component because it calls before the render method in which we write HTML code.

Why do we need constructor?

We need constructor in a` class component to use state and inside the constructor we will declare state like **this.state= {message: "God is great"}**and before that super() method as well.

getDerivedStateFromProps() It will call just before the render. It will keep props initial state

Render() Method- It is used in class component to display the HTML code in the HTML element. Class component will not work if we don't use render method. Render method will call itself when HTML is ready, state is updated, props are passed.

As soon as we create class component, our class component will be ready and then render method will call.

ComponentDidMount() We use componentDidMount when we have to call API(call any data from any file/folder). It will call itself after render when all HTML, CSS and other things are ready/loaded/written.

Updating:- When we are updating state or passing props.

getDerivedStateFromProps() - It will keep props initial state or initial data of the state

shouldComponentUpdate()- When we setState() or change the state value. It is a unique method of life cycle method that ask question whether I should update the component or not. So, we set any condition and it will render only if the condition is fulfilled. So, it has the power to stop the render as well. In order to update the value, we need to return true in this method as you can see in the notes.

render()

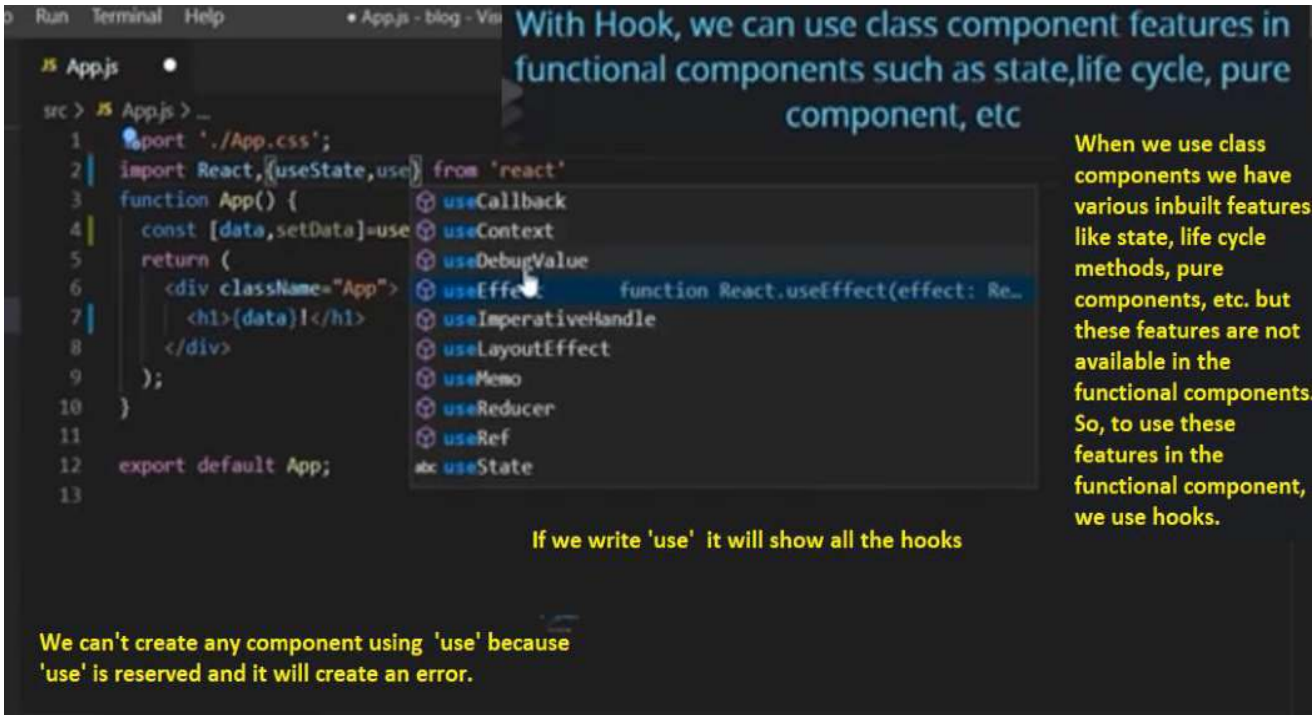
getSnapshotBeforeUpdate()- We will get the value of snapshot if we already defined the getSnapshotBeforeUpdate

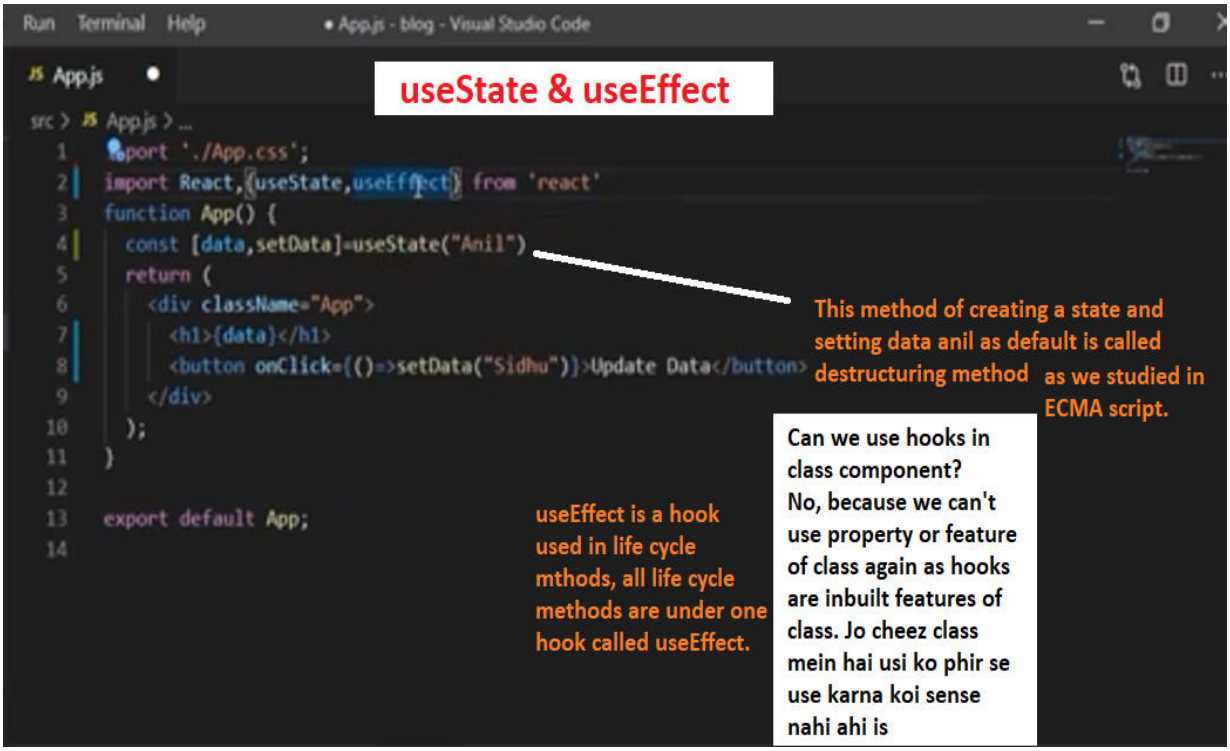
componentDidUpdate()- It has three parameters like componentDidUpdate(preProp, preState, snapshot)- [preProp and preState will call the previouse props and states . The benfit of preState is that we can check if the prevouse state is same with the current value. We use to check if we receive the same data twice from the user through API.

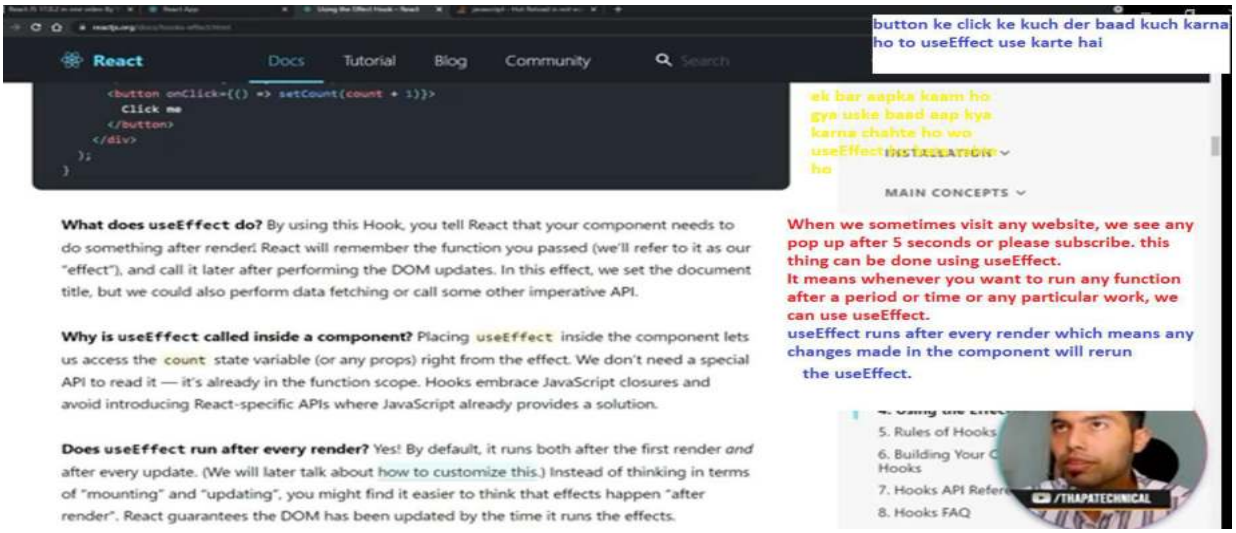
Unmounting:-

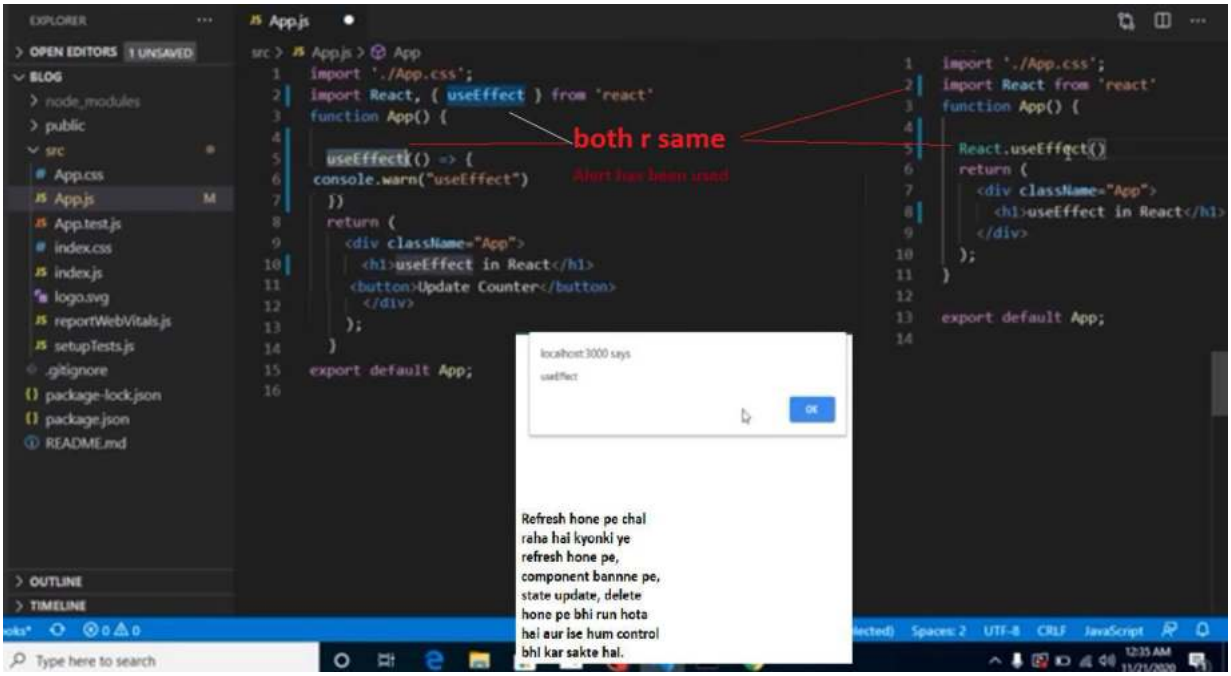
componentWillUnmount()- This method will be called as soon as any method/component is removed from DOM. Whenever we hide any component it removes that component from DOM entirely. It will call just before removing any component.

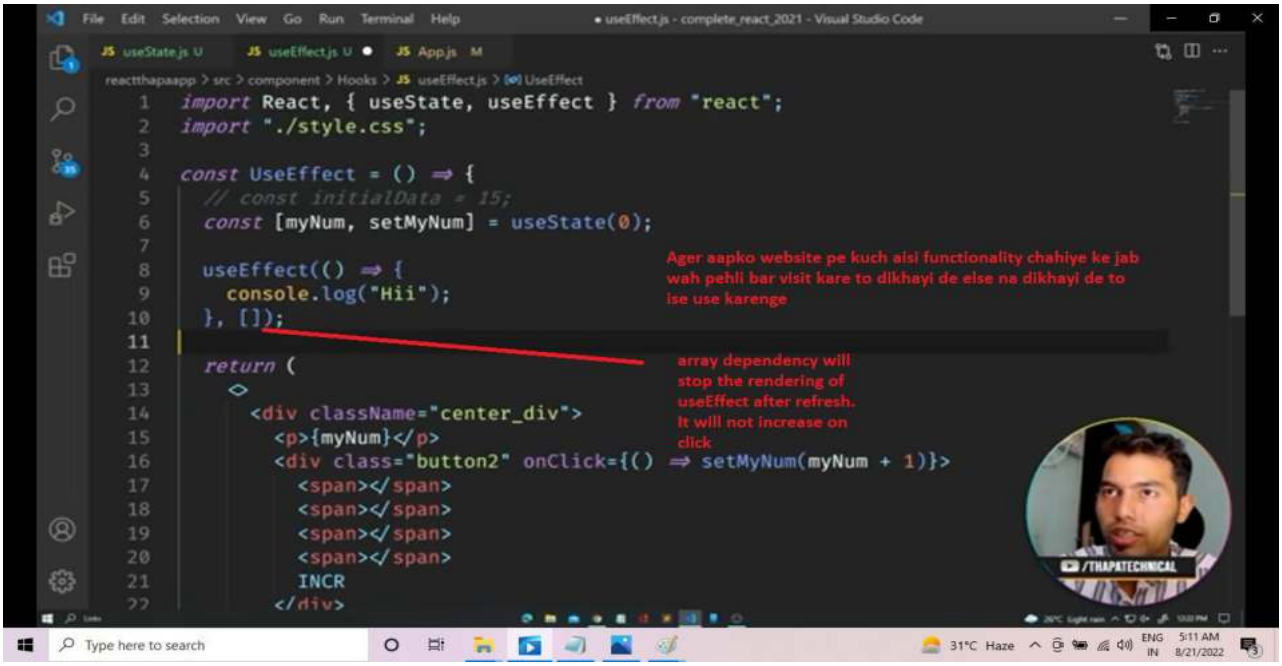
HOOKS


With Hook, we can use class component features in functional components such as state,life cycle, pure component, etc
When we use class components we have various inbuilt features like state, life cycle methods, pure components, etc. but these features are not available in the functional components. So, to use these features in the functional component, we use hooks.
If we write 'use' it will show all the hooks
We can't create any component using 'use' because 'use' is reserved and it will create an error.

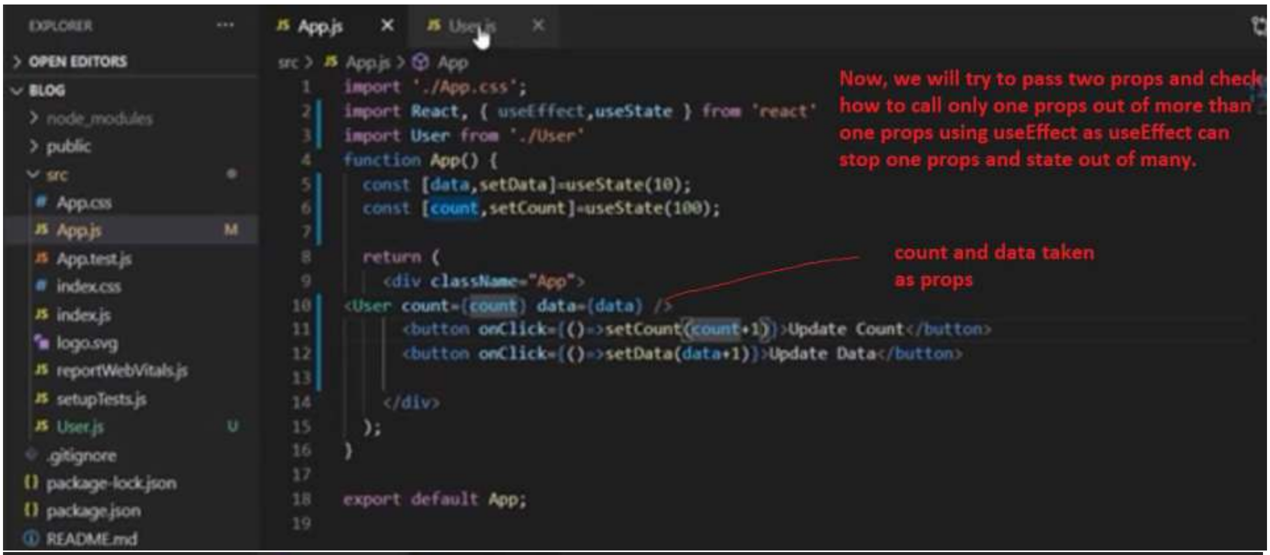

useState & useEffect
This method of creating a state and setting data anil as default is called destructuring method as we studied in ECMA script.
useEffect is a hook used in life cycle mthods, all life cycle methods are under one hook called useEffect.
Can we use hooks in class component? No, because we can't use property or feature of class again as hooks are inbuilt features of class. Jo cheez class mein hai usi ko phir se use karna koi sense nahi ahi is

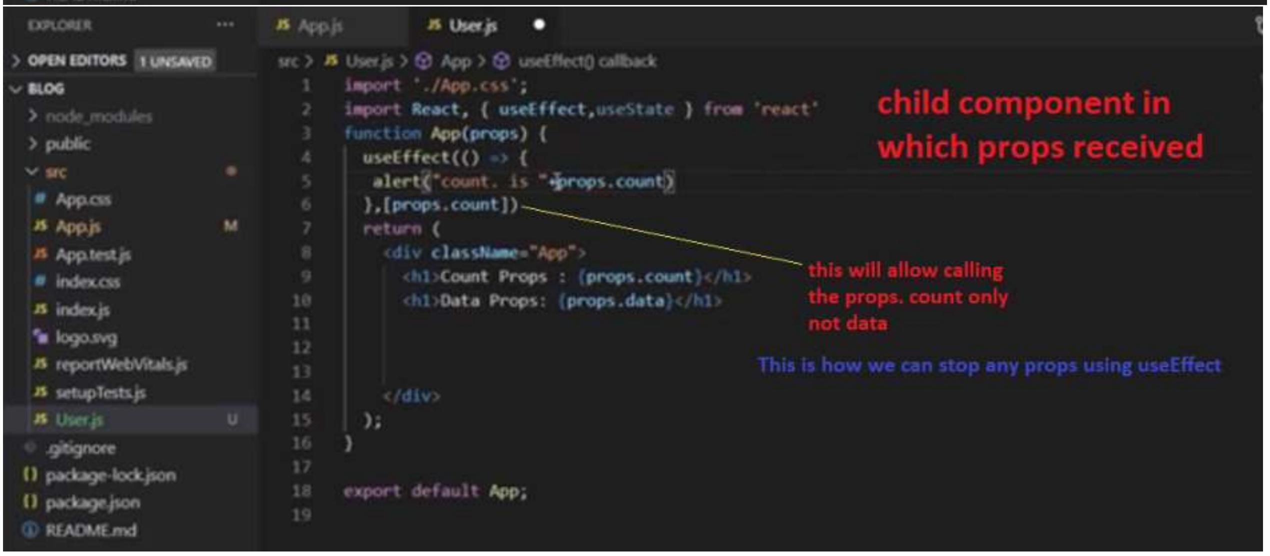

button ke click ke kuch der baad kuch karna ho to useEffect use karte hai
What does useEffect do? By using this Hook, you tell React that your component needs to do something after render. React will remember the function you passed (we'll refer to it as our "effect"), and call it later after performing the DOM updates. In this effect, we set the document title, but we could also perform data fetching or call some other imperative API.
Why is useEffect called inside a component? Placing useEffect inside the component lets us access the count state variable (or any props) right from the effect. We don't need a special API to read it — it's already in the function scope. Hooks embrace JavaScript closures and avoid introducing React-specific APIs where JavaScript already provides a solution.
Does useEffect run after every render? Yes! By default, it runs both after the first render and after every update. (We will later talk about how to customize this.) Instead of thinking in terms of "mounting" and "updating", you might find it easier to think that effects happen "after render". React guarantees the DOM has been updated by the time it runs the effects.
When we sometimes visit any website, we see any pop up after 5 seconds or please subscribe. this thing can be done using useEffect.
It means whenever you want to run any function after a period or time or any particular work, we can use useEffect.
useEffect runs after every render which means any changes made in the component will rerun the useEffect.





Props & states in useEffect





CLEANUP FUNCTION IN USEEFFECT()

```
import React, { useEffect, useState } from 'react';
function State() {
    const [data, setData] = useState(window.screen.width);

    function abc() {
        setData(window.innerWidth);
        console.log("calling");
    }
    useEffect(() => {
        console.log("add event");
        window.addEventListener("resize", abc);
        // on every resize,this abc funciton will be called and  saved on the memory that
        // may crash the window or full the server memory or can create issue in the applications,  we want it to be deleted
        // from the memory and for that we have default clean up function in use Effect as shown under return in the next line
        return () => {
            console.log("remove event");
            window.removeEventListener("resize", abc)
        }
    });
    return (
        <>
            <h3>Hello, Screen size is {data}</h3>
        </>
    )
}
export default State;
```

```
useEffect(() => {
  window.addEventListener("resize", actualWidth);

  return () => {
    window.removeEventListener("resize", actualWidth);
  }
});
```

20. Explain about types of side effects in React component.

Side effect can be everything that might be happening to your application which is not related to the UI rendering in React. For example:- sending an HTTP Request to the server, when we send HTTP Request to the server, we don't render anything in the UI.
Now, react should re-render the UI like it does when the state changes.
Another example, when we store something in the browser storage. It does not re-render the UI and this is why it is a side-effect.
There are two types of side effects.
Effect with cleanup - We get cleanup function in useEffect by default. We use it to clean the function that keeps on running which might crash the browsers.

```
useEffect();
useEffect(()=>{ effect
        Return ()=>{
        cleanup}
        }, [input])
```

and Effect without cleanup

```
useEffect(() => {
    window.addEventListener("resize", actualWidth);

    return () => {
        window.removeEventListener("resize", actualWidth);
    }
});
```

https://www.youtube.com/watch?v=5gCtW7RCtQA

21. What are synthetic events in React?

Synthetic events are those events which behave similar on all the browsers. Like prevent.default() because it does not care which browser you run your websites on.

stopPropagation()
persist()

```
import React from 'react';
function User1(){
    function sarfraz(e){
        e.preventDefault();
        document.write("You just clicked");   // e is called synthetic
    }
    return(
        <>
        <h1>Example For Sythetic Event</h1>
        <a href='https://www.google.com/' onClick={sarfraz}>Visit google.com</a>

        </>
    )
}
```


```
export default User1;
```

In the given example- e is synthetic event.

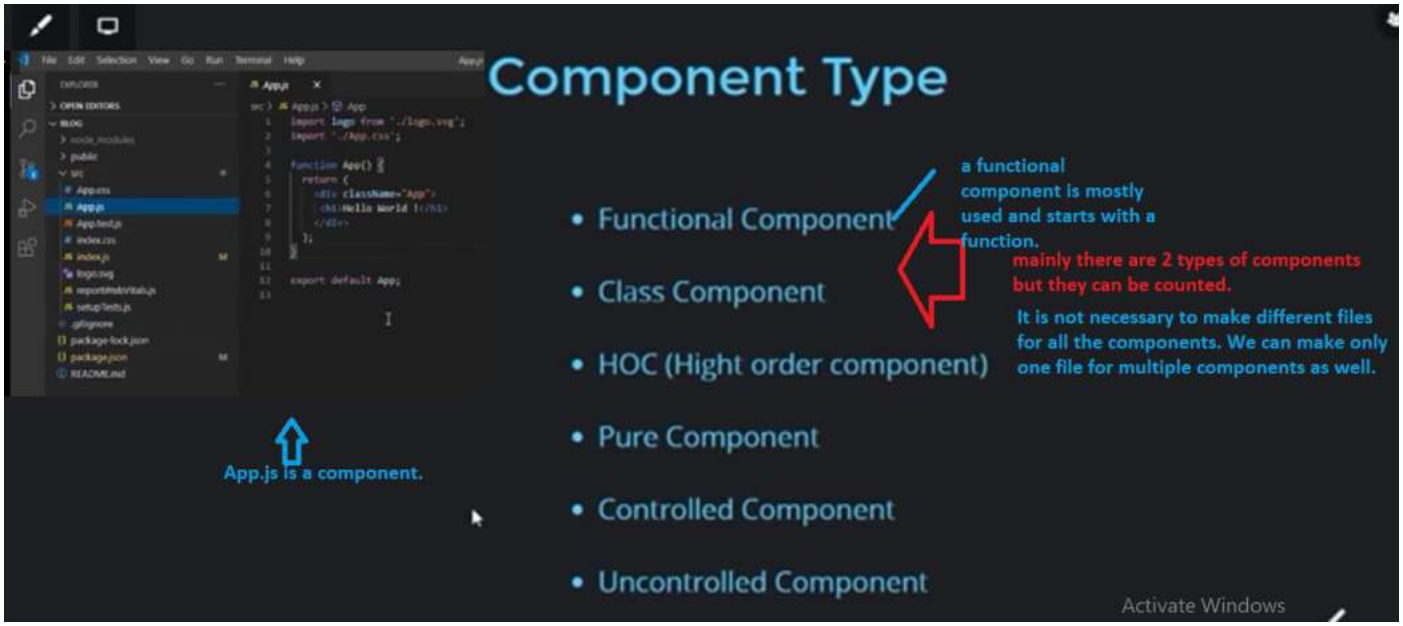
22. What are forms in React and how do you create forms in React?

Refer notes

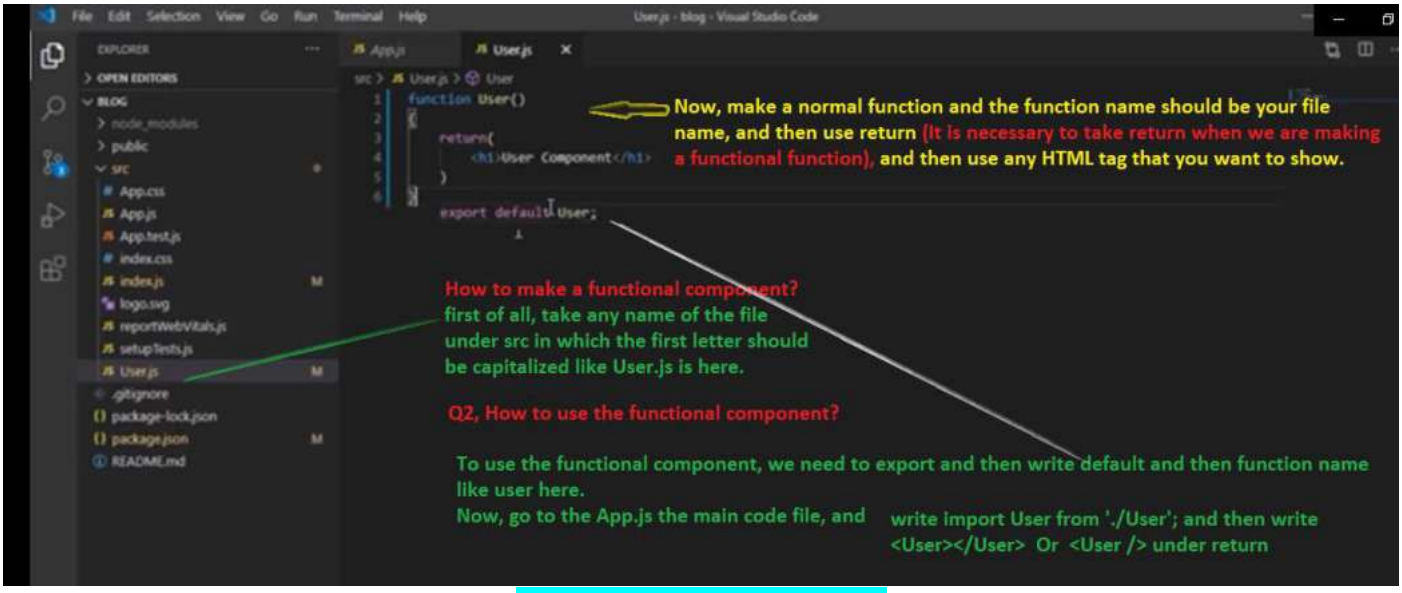
23. What are the difference between class and functional components?

Class Components	Functional Components
We should use class component when we have to use state and life cycle methods which are be default in class component and this is why it can hold or manage state and work with all life cycle methods and also be reused.	Cannot hold or manage state
It is tough to understand	Simple and easy to understand
Can work with all life cycle methods	Does not work with any life cycle method
Can be reused	Cannot be reused

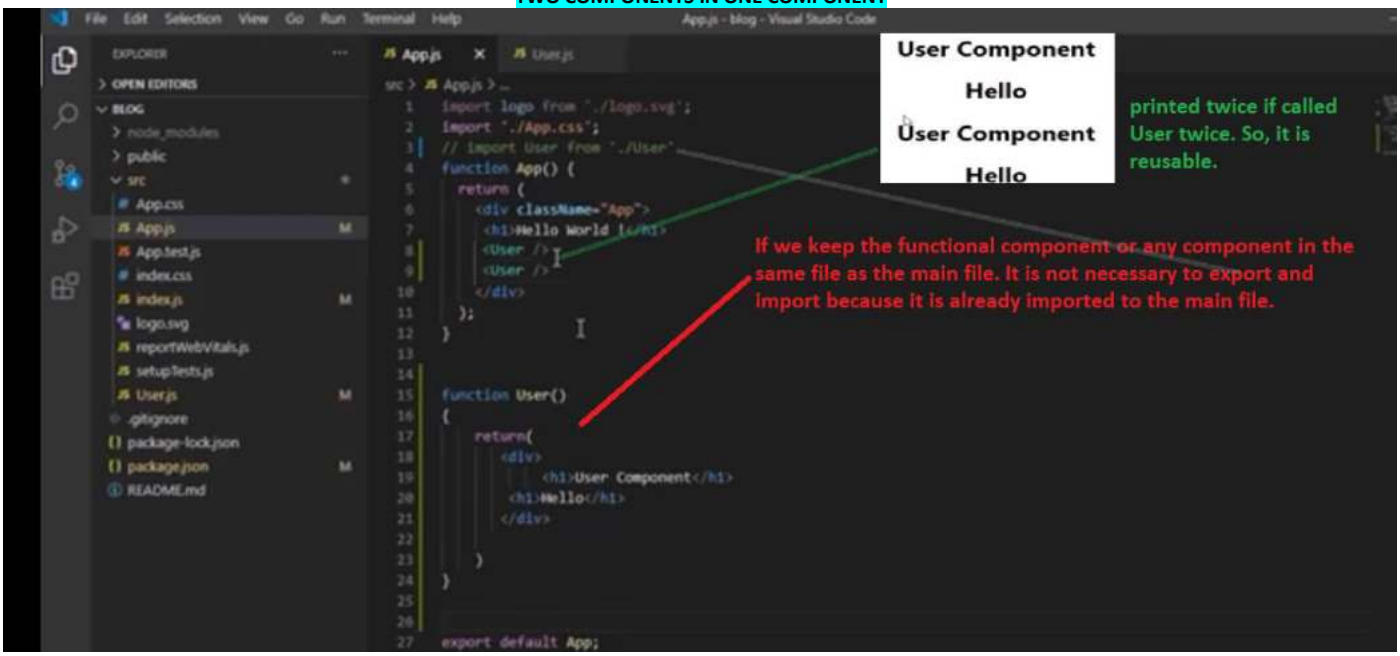
24. Define components.

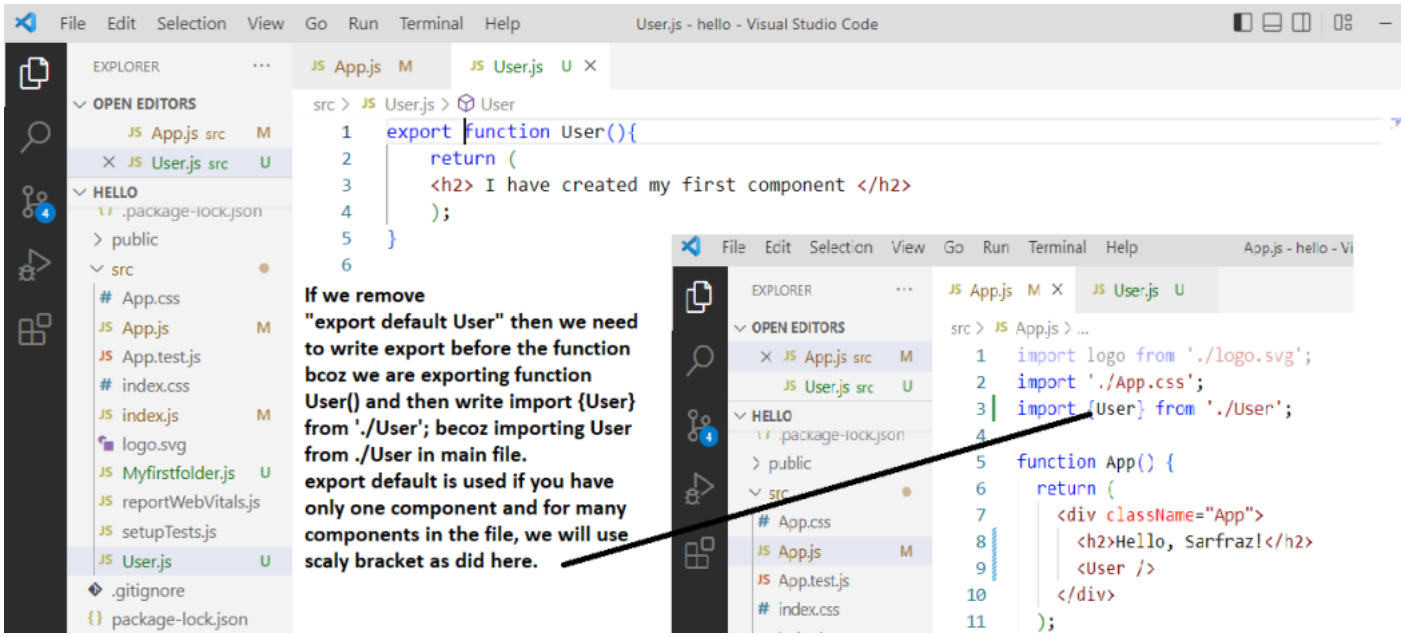


FUNCTIONAL COMPONET

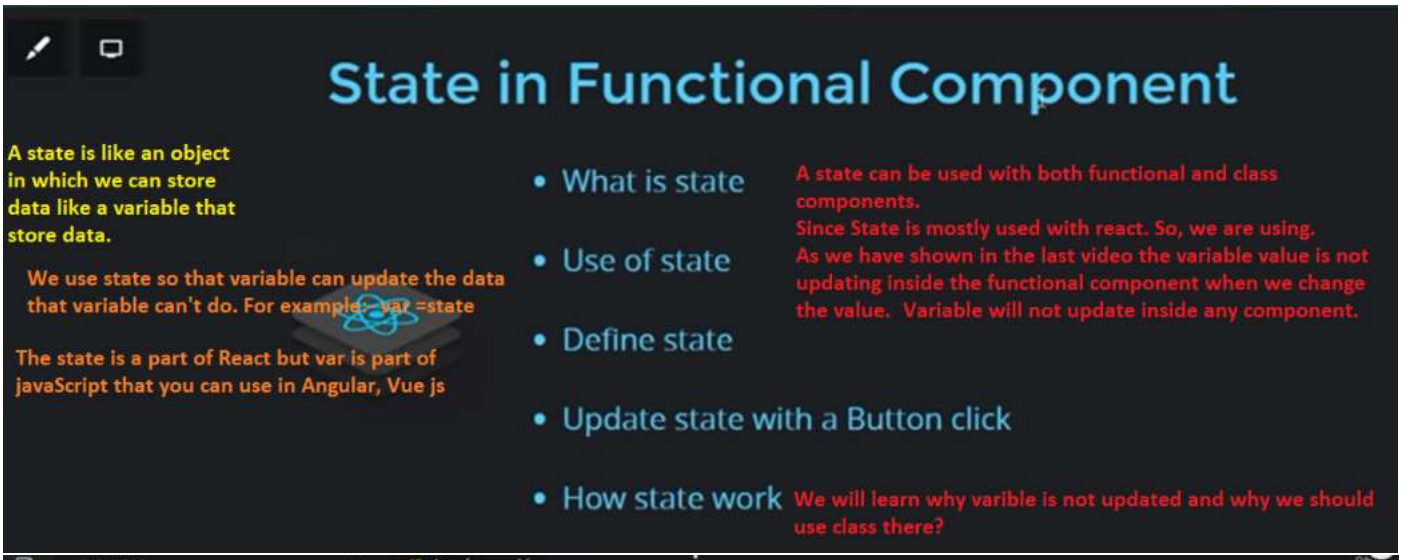


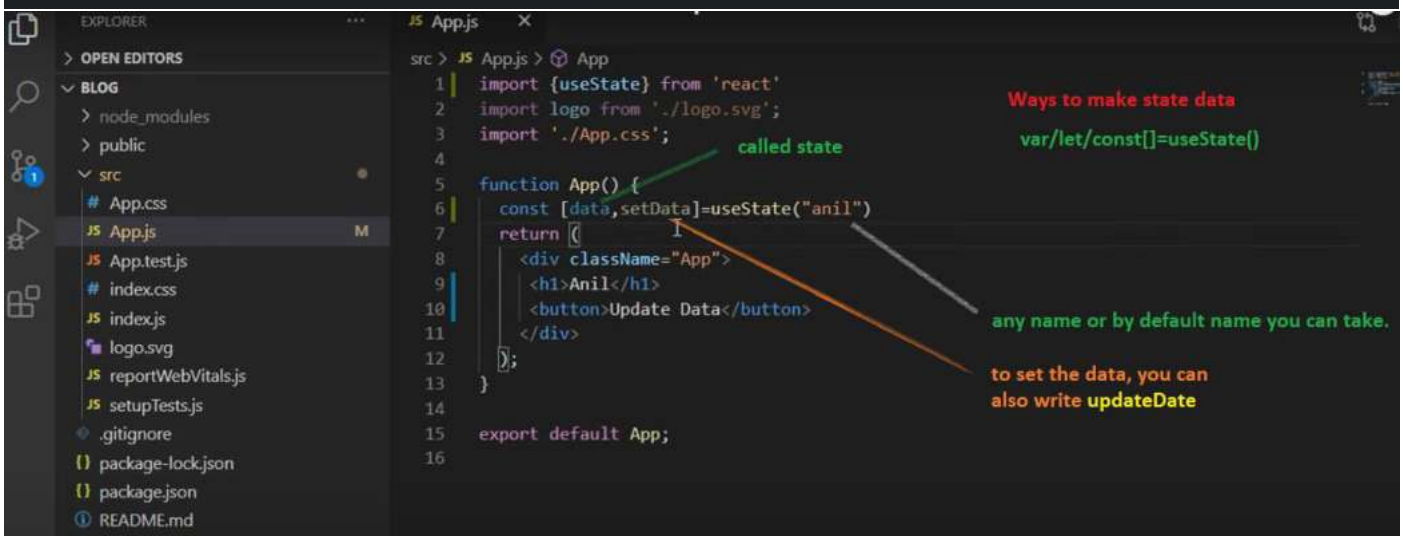
TWO COMPONENTS IN ONE COMPONENT

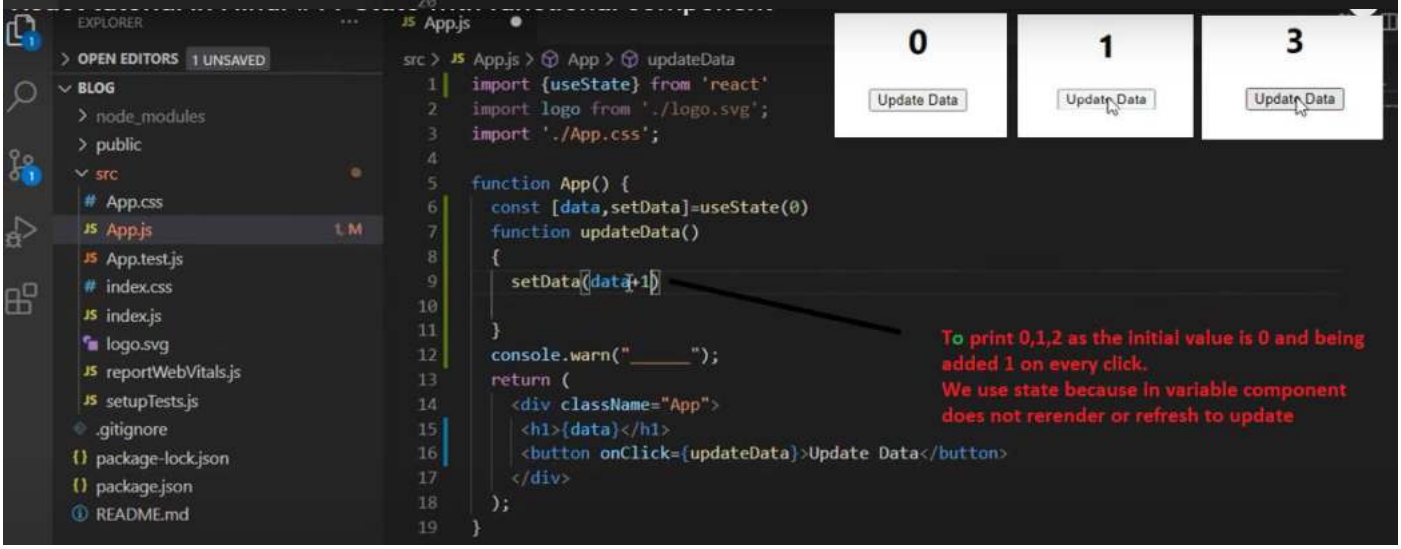




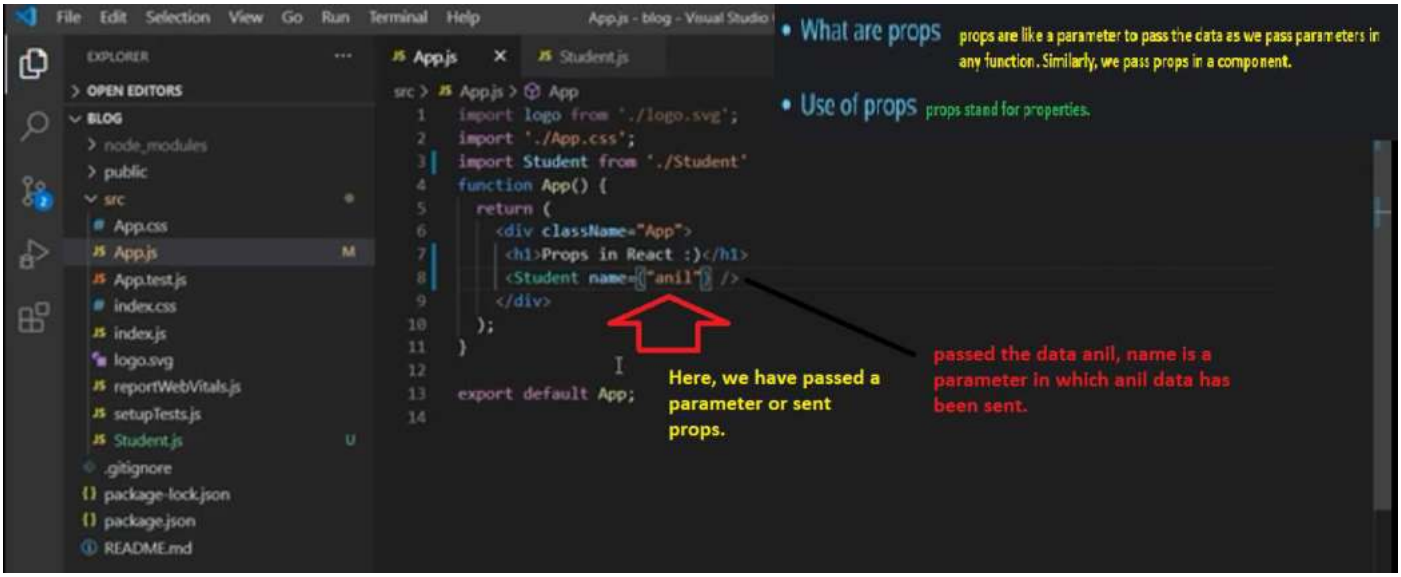
STATE IN FUNCTIONAL COMPONENT

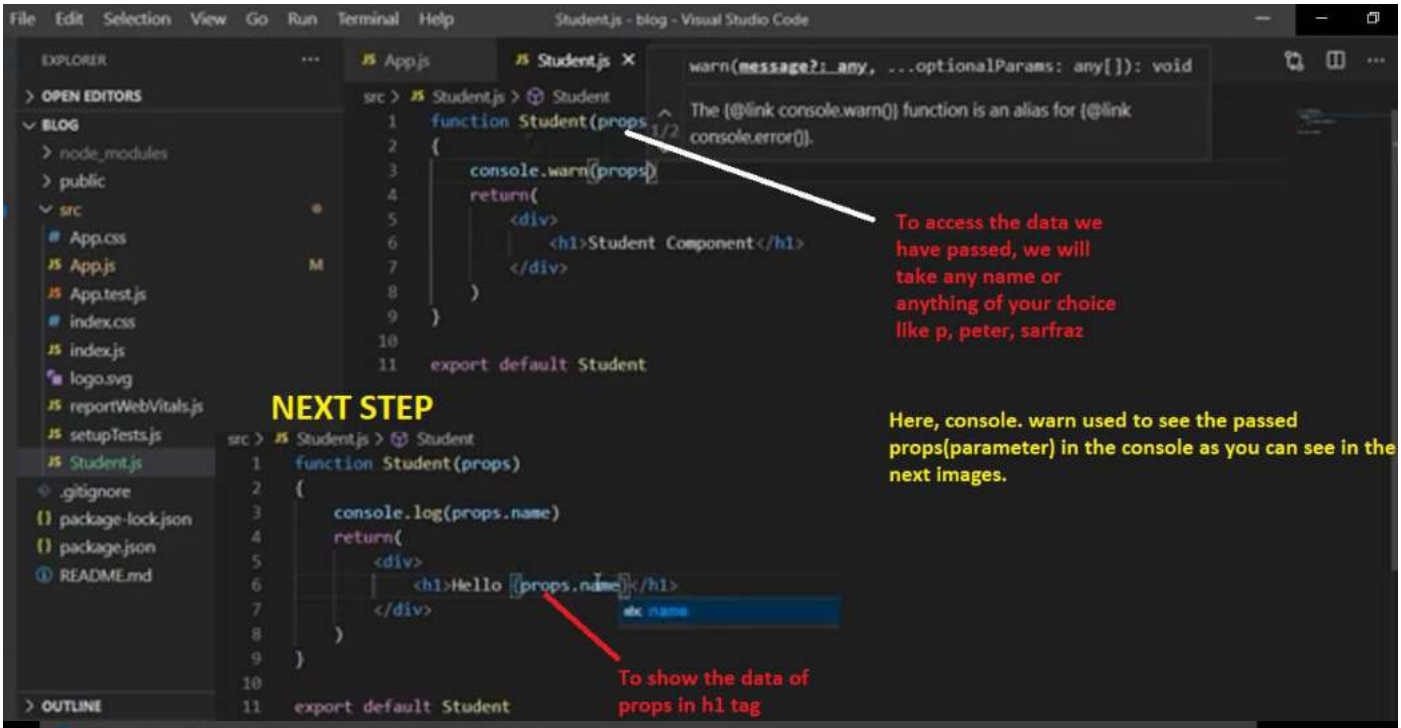


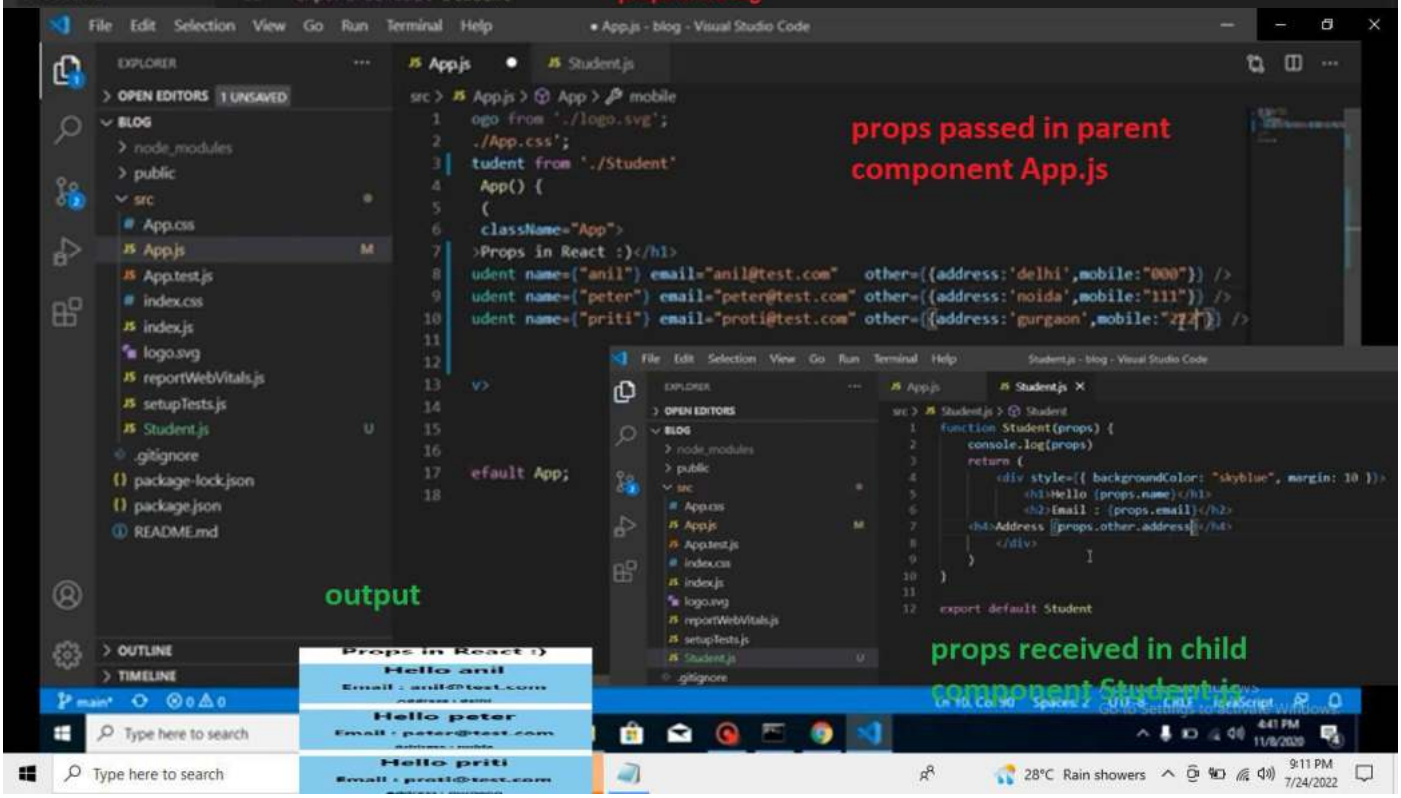




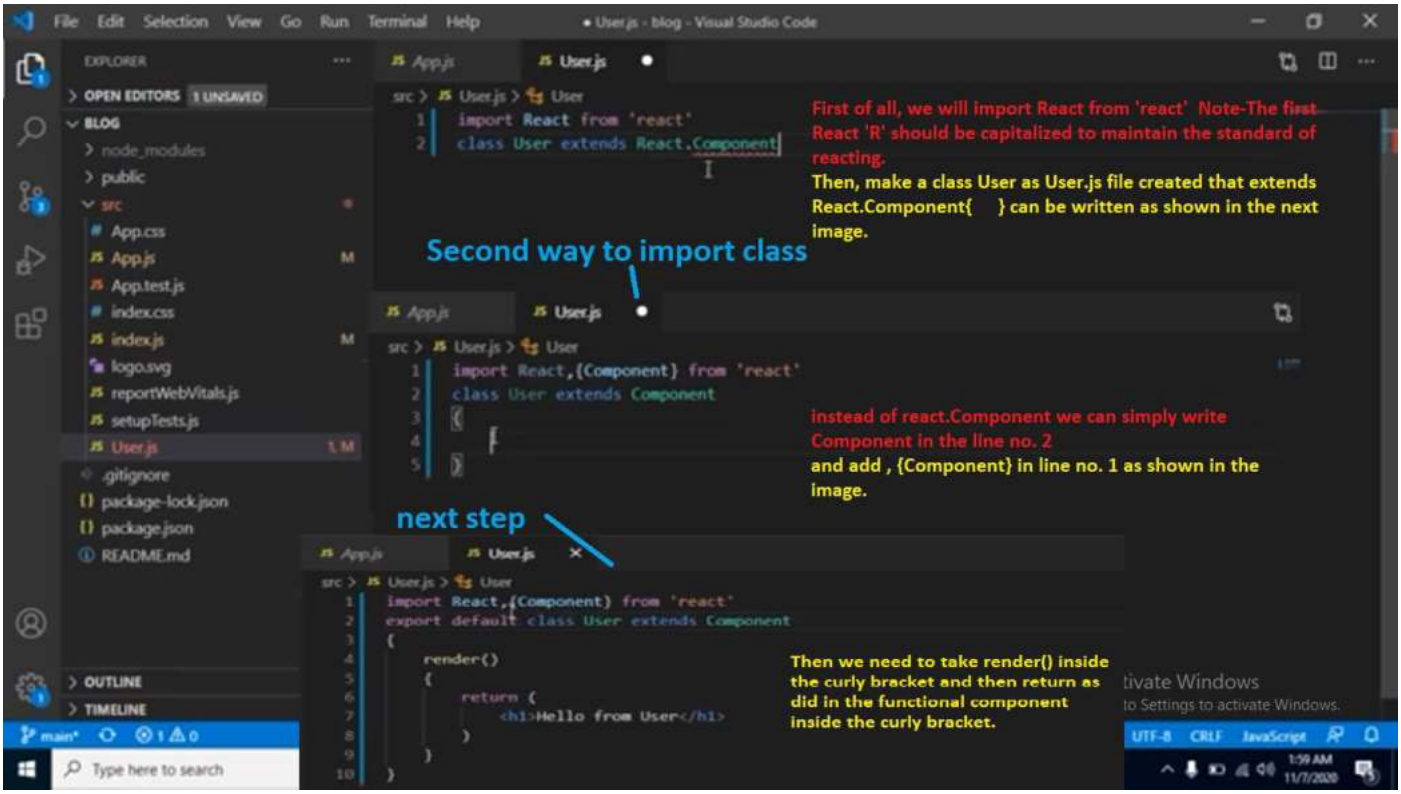
PROPS IN FUNCTIONAL COMPONENT

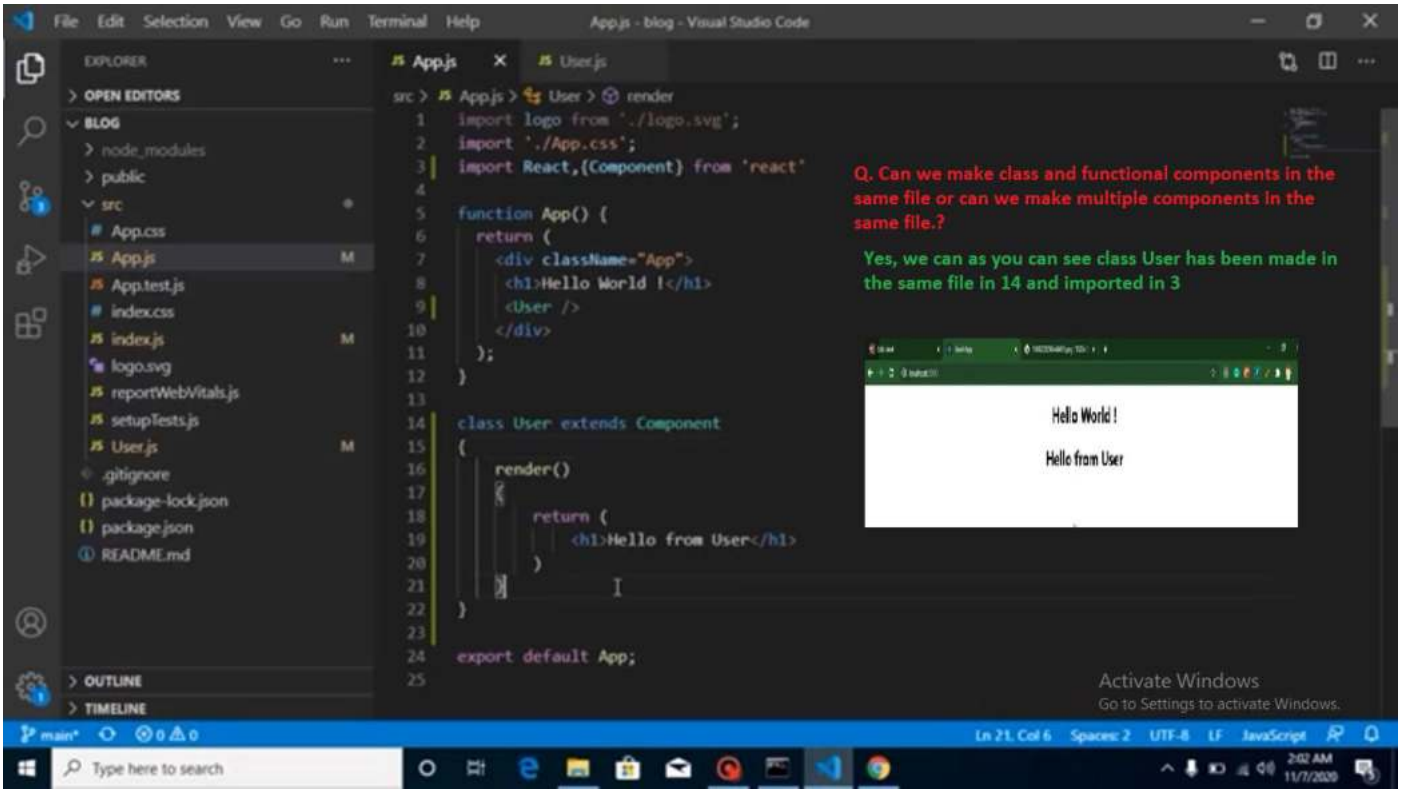


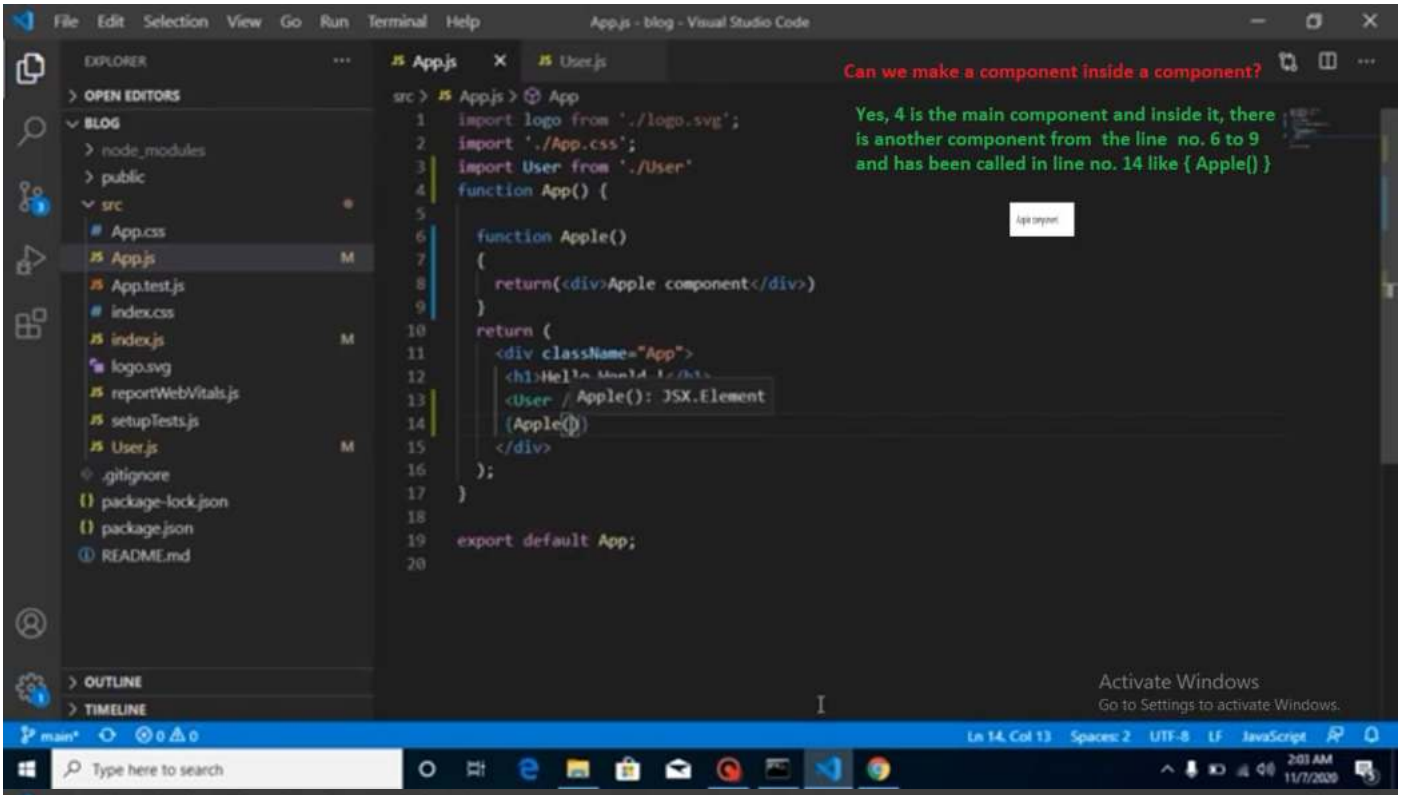


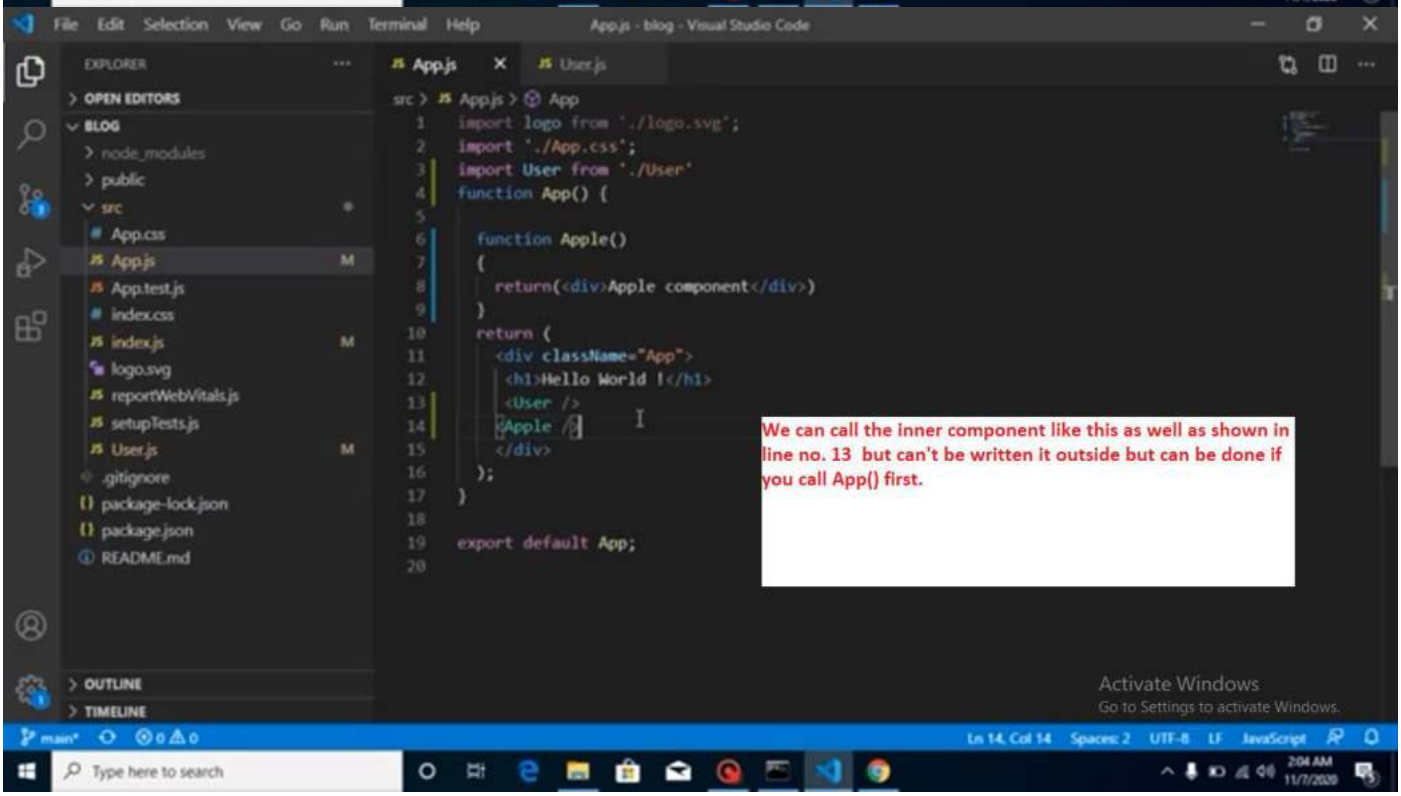


CLASS COMPONENT

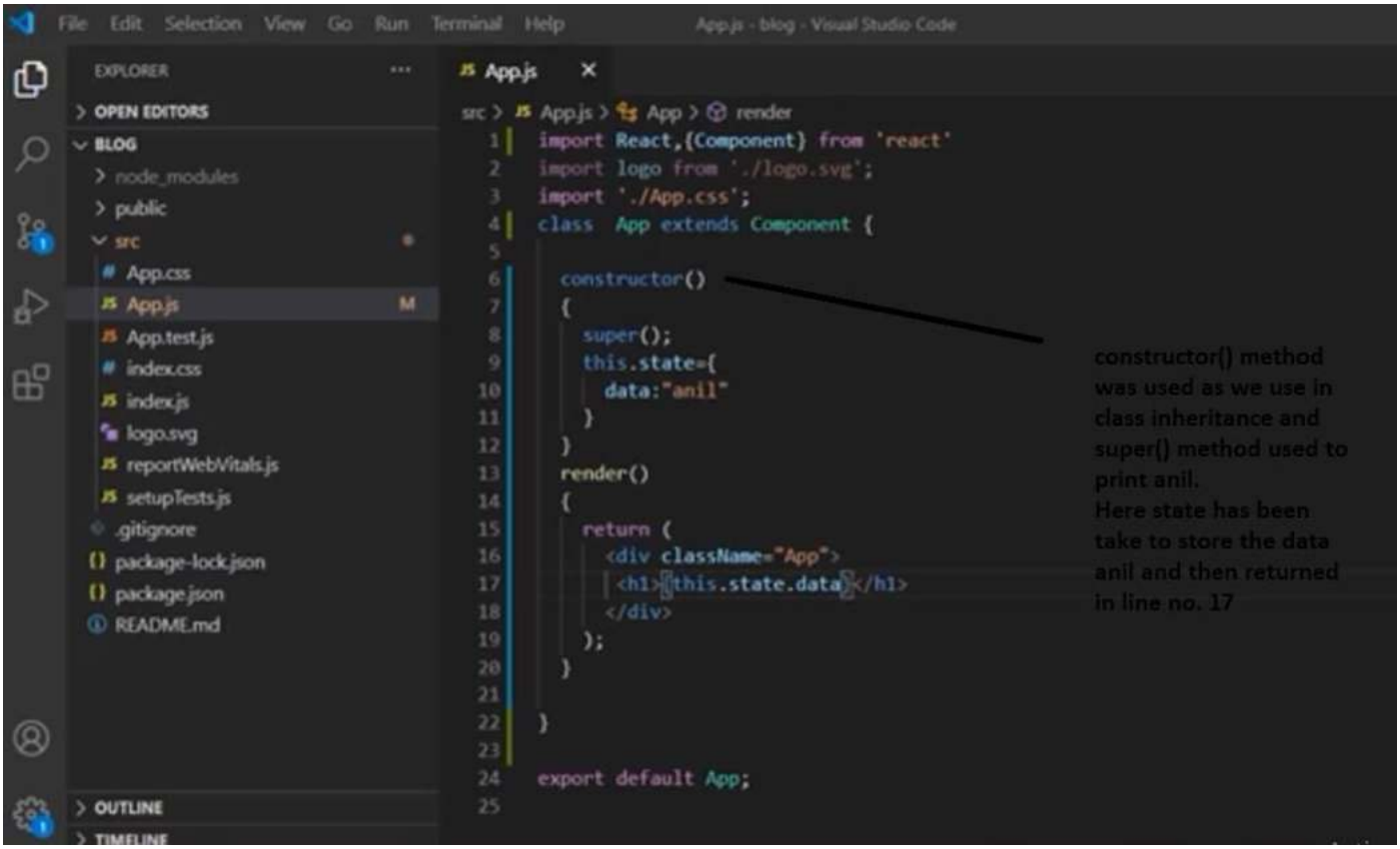

First of all, we will import React from 'react' Note-The first React 'R' should be capitalized to maintain the standard of reacting.
Then, make a class User as User.js file created that extends React.Component{ } can be written as shown in the next image.
Second way to import class
instead of react.Component we can simply write Component in the line no. 2
and add , {Component} in line no. 1 as shown in the image.
next step
Then we need to take render() inside the curly bracket and then return as did in the functional component inside the curly bracket.

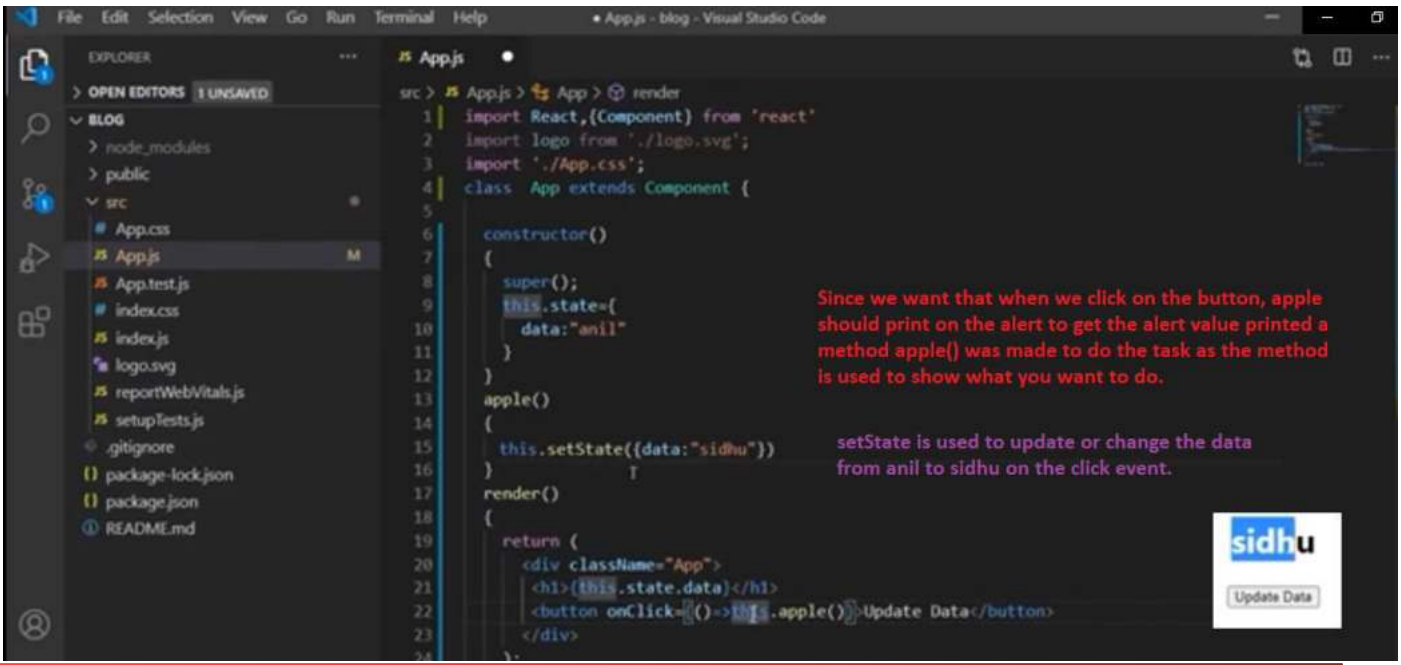

Q. Can we make class and functional components in the same file or can we make multiple components in the same file.?
Yes, we can as you can see class User has been made in the same file in 14 and imported in 3
Hello World !
Hello from User





STATE IN CLASS COMPONENT





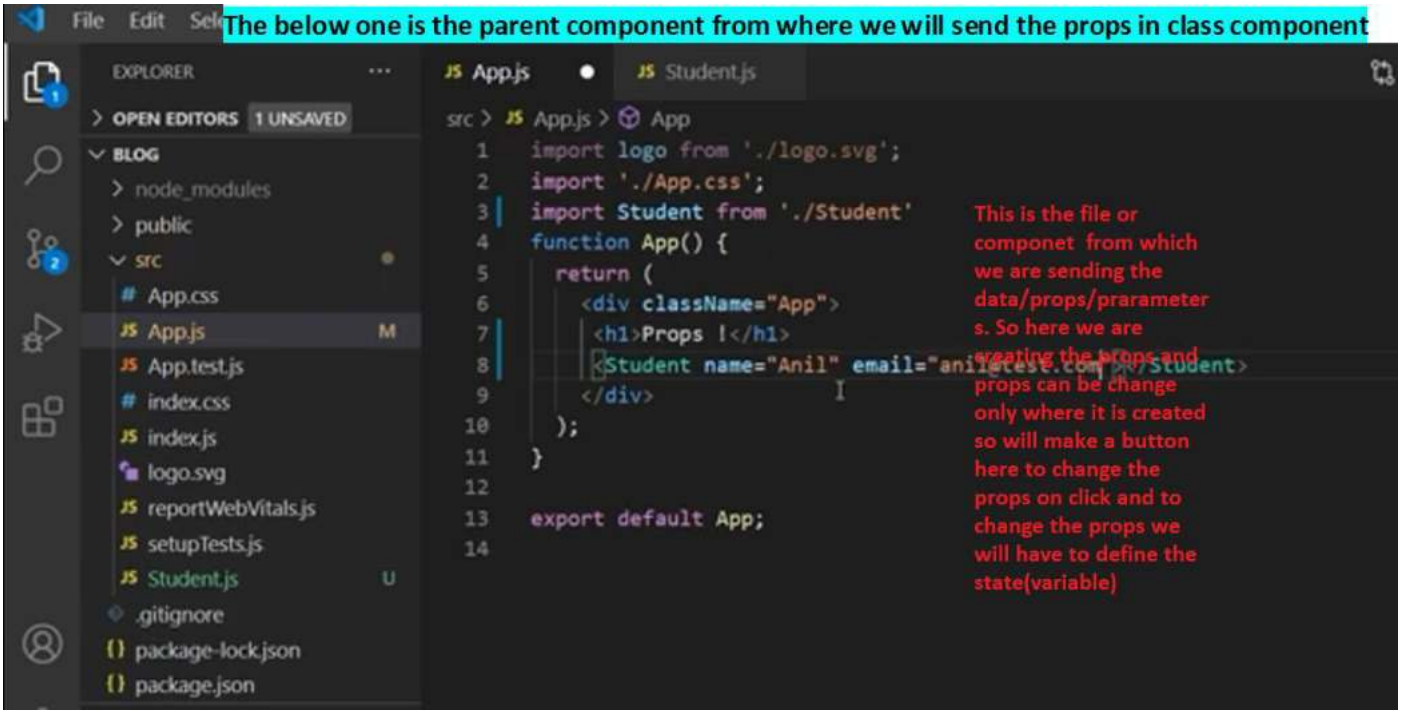
3. Do we use the super() method in react to print the value of child of sub-methods like constructor or any other method if we use the same variable or methods in the super(Parent) and sub(Child) class? Is used in using state with the class component?

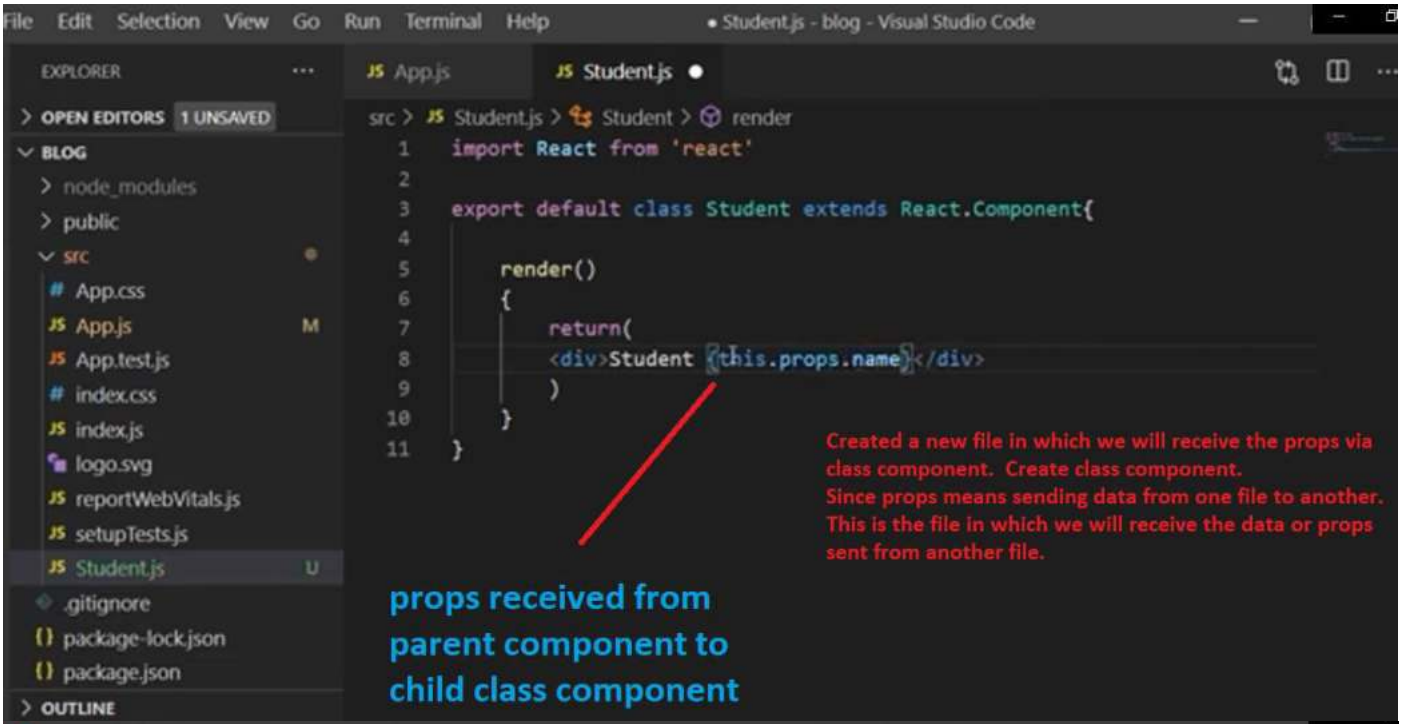
4. We can not use a state outside the component as it will not be an authentic method. Usually, it can be done as all is an object but it will not be authentic as react says not to use it outside.

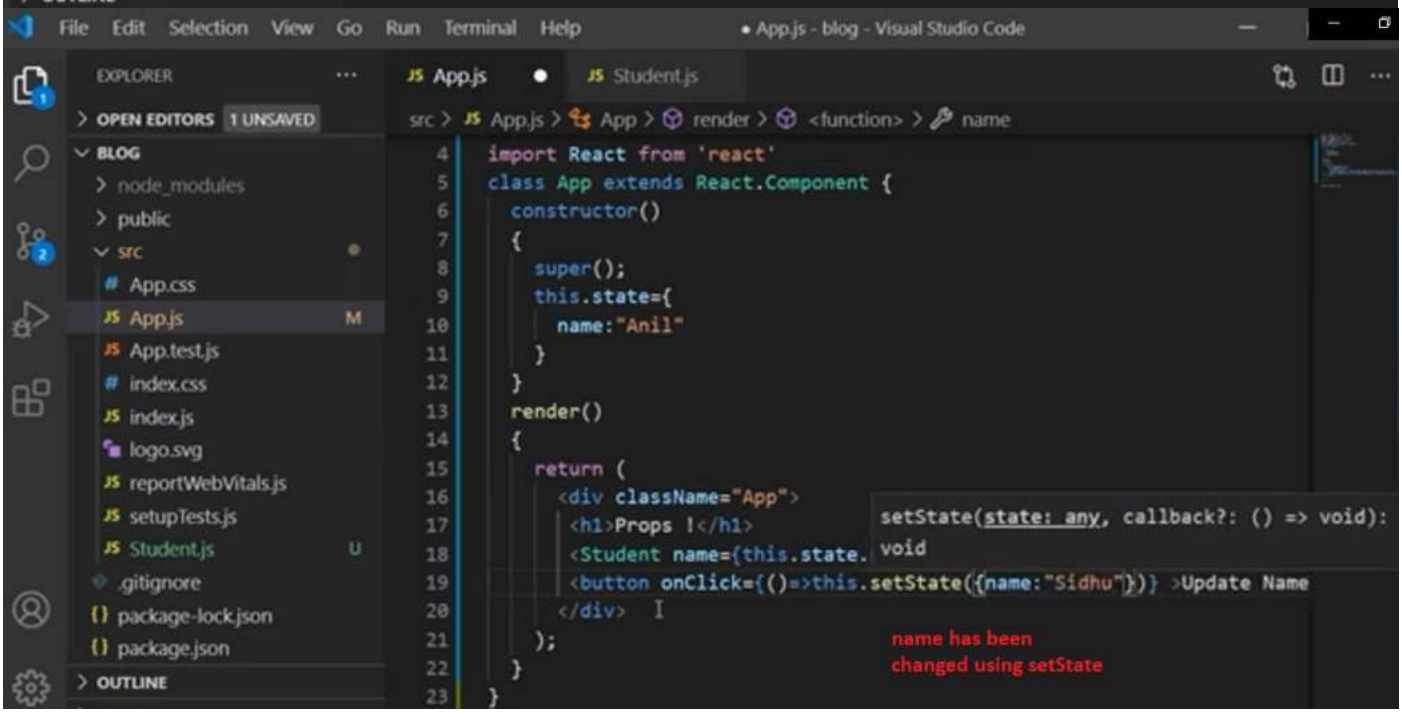
The state in react is public, not private

PROPS IN CLASS COMPONENT

The below one is the parent component from where we will send the props in class component



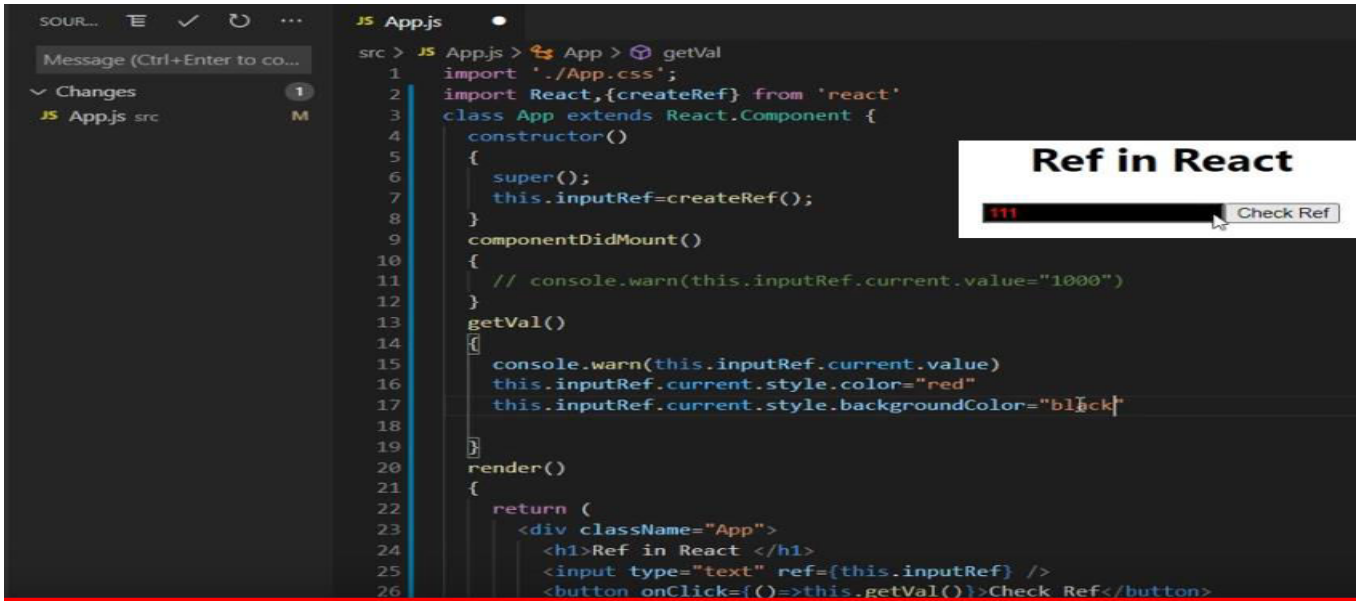




25. What do you understand by refs in React?

Ref is a unique feature and can't be used it in functional component directly. So we will first use it in class component. It is suggest by React to use 'Ref' as less as you can because it manipulates the DOM directly and if you manipulate DOM, React will be a bit slow. Ref has power to hide and show(that we do by state in React) but Ref can do it without state, giving style or changing style and adding color. So, Ref has power to manipulate the DOM which React does not suggest to do. So, it is better to use during an emergency.

Cases we use Ref with class component



It is better to use in form validation, getting form input value, setting color etc.
We use Ref in class component.

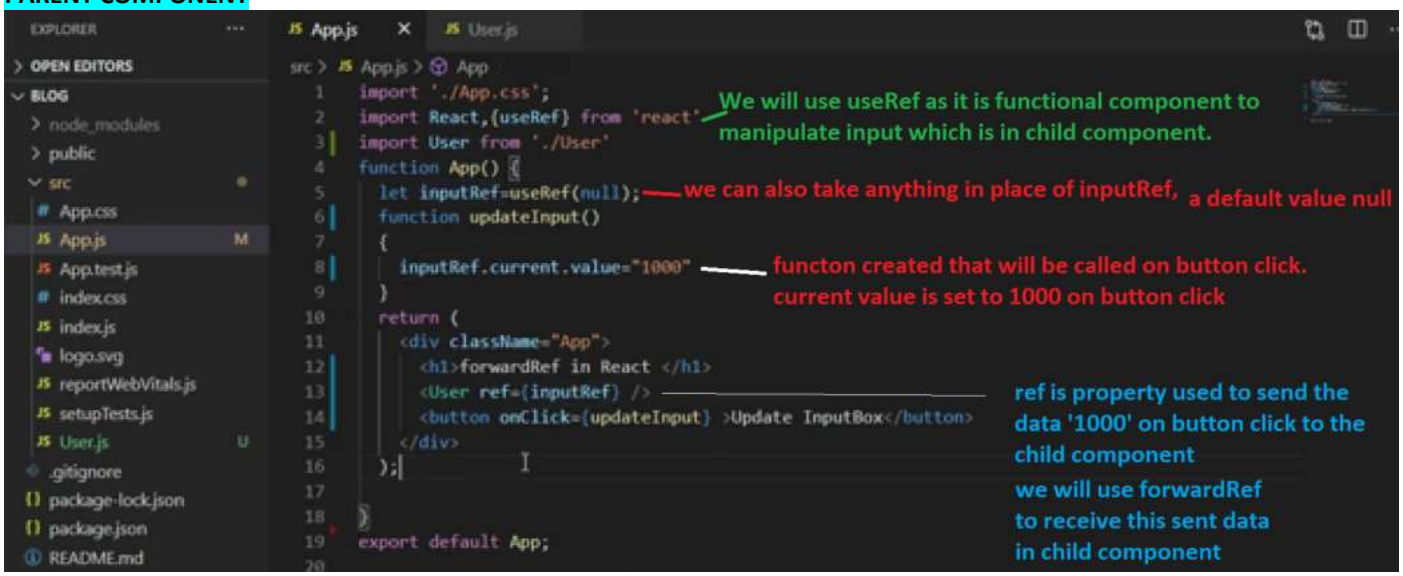
useRef:- It is used in functional component and it is a hook. We can do the same thing that can be done with Ref. The only difference is that Ref is used with class component and useRef is used with functional component. You can see this
code or the code written as **CODE FOR UC WITH useRef**

```
import React, { useRef } from 'react';
function State(){
              const inputRef =useRef();
              console.log("All I got", inputRef);
              function sarfraz(){
               // inputRef.current.value= "100"
               inputRef.current.style.color= "red";
               inputRef.current.style.textAlign= "center";

              }
       return(
           <>
           <h1>useRef in functional Component</h1>
           <input type= "text" ref={inputRef}/>
           <button onClick={()=>sarfraz()}>On Click</button>
           </>
       )
}
export default State;
```

forwardRef:- forwardRef is used when we have to manipulate DOM of an input which is in child component by clicking on a button which is on parent component using props. We usually use Ref and useRef when we have to manipulate DOM from the same component but if we have two components like parent and child then we will use forwardRef. It is like forwarding message form one to another.

In another word, we use forwardRef when we have an input tag in child component and button tag in parent component and Now, I want that when I click on the button, we can change the input value. We will use useRef in parent component and forwardRef in child component and use props to transfer the data.

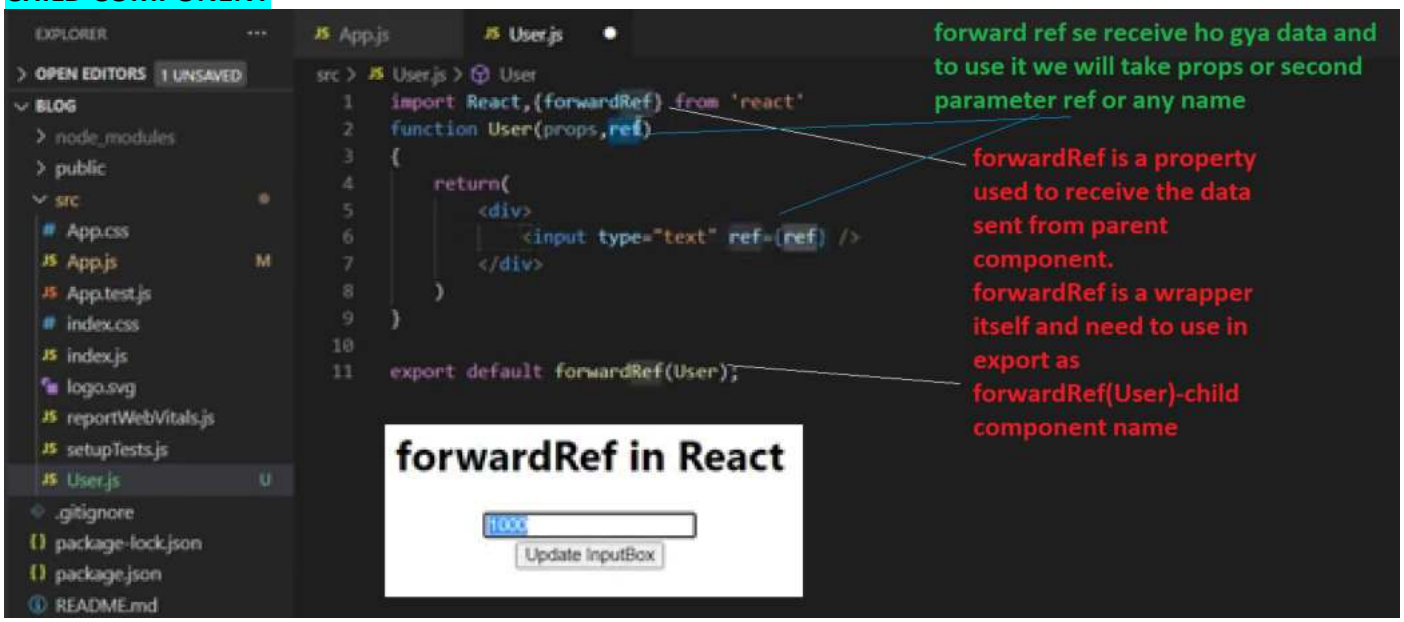
PARENT COMPONENT




```
inputRef.current.style.color="red"
```

```
inputRef.current.focus()
```

CHILD COMPONENT



28. What do you know about controlled and uncontrolled components?

When we talk about controlled component and uncontrolled components we should understand that we are referring to the component which has one or more than one input fields inside a form. Whenever we control React input fields through state, it is called controlled component and those components which are directly handled by DOM(JavaScript) or ref in React is called uncontrolled components. Like in JavaScript, we use getElementByID to directly target and make changes.

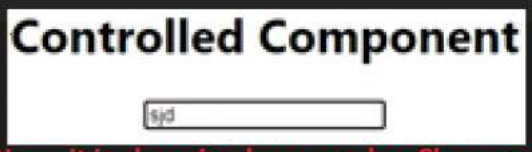
Controlled component is handled by React.js state but uncontrolled component is handled by DOM. We use useRef to directly manipulate DOM in React.

CODE

```jsx
import './App.css';
import React,{useState} from 'react'
function App() {
  let [val,setVal]=useState("")
  return (
    <div className="App">
      <h1>Controlled Component </h1>
      <input type="text"  value={val} />
    </div>
  );

}
export default App;
```

We will not be able to write anything in the input area because we have given value = null or " " or nothing which is a fixed value like if we write here value=100 then 100 will be shown in the input area but can not be changed and to change it we will use onChange method

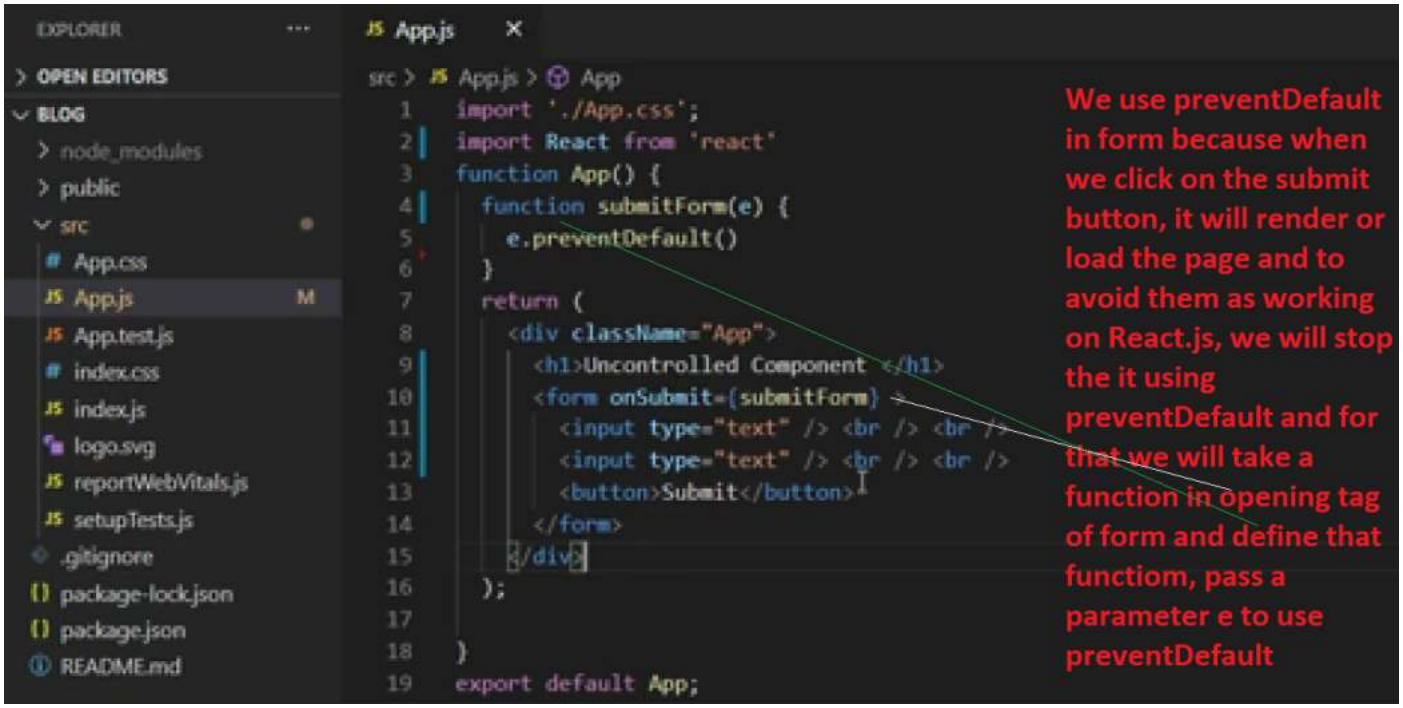
```jsx
import './App.css';
import React,{useState} from 'react'
function App() {
  let [val,setVal]=useState("")
  return (
    <div className="App">
      <h1>Controlled Component </h1>
      <input type="text"  value={val} onChange={(e)=>setVal(e.target.value)} />
    </div>
  );

}
export default App;
```



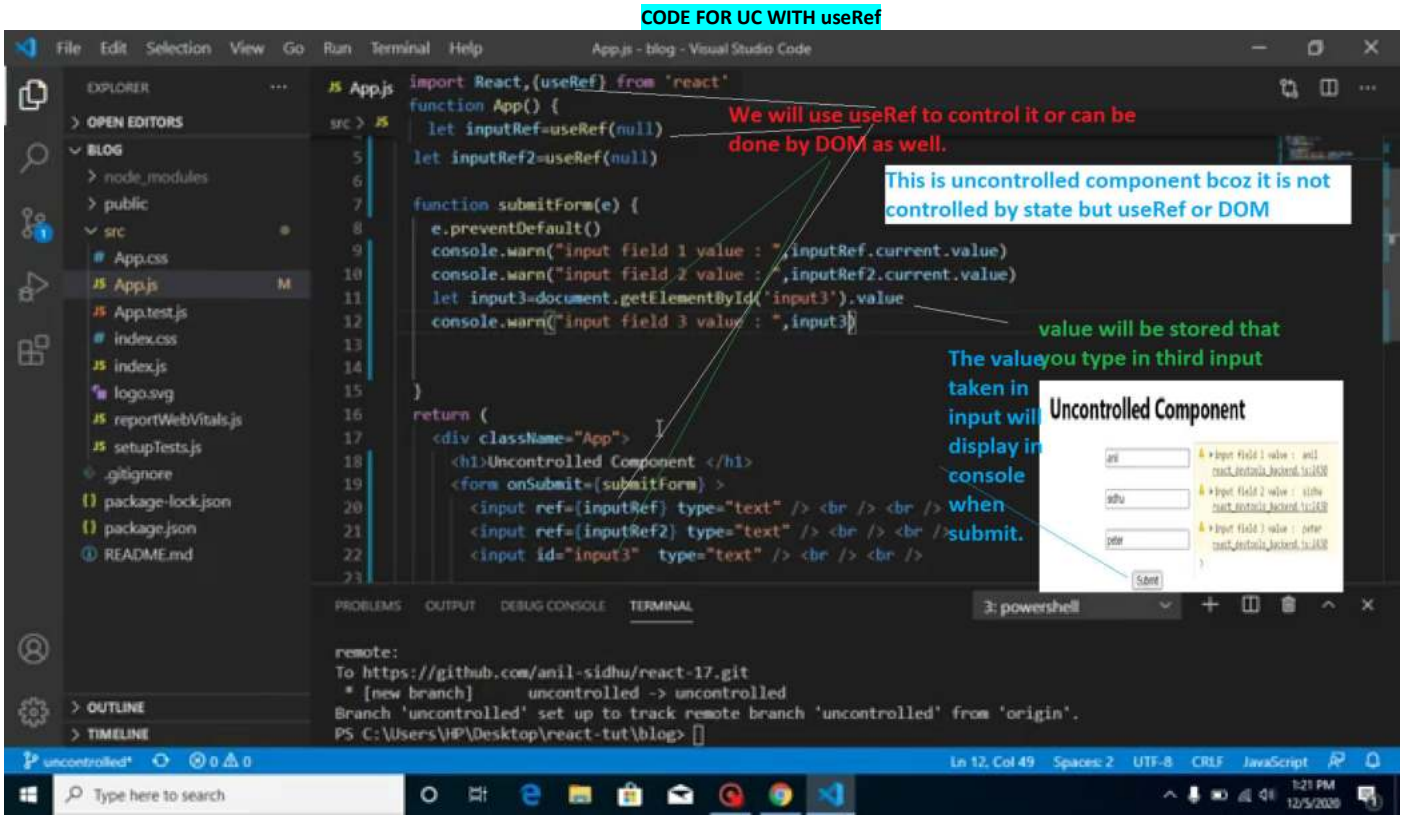
Now, it is changing bcoz used onChange and it is a controlled component becoz it is being controlled by component state and it has input tag, e is parameter that we take with onchange

```jsx
import './App.css';
import React,{useState} from 'react'
function App() {
  let [val,setVal]=useState("")
  let [item,setItem]=useState("")

  return (
    <div className="App">
      <h1>Controlled Component </h1>
      <input type="text"  value={val} onChange={(e)=>setVal(e.target.value)} />
      <input type="text"  value={item} onChange={(e)=>setItem(e.target.value)} />
    <h3>Value {val}</h3>

    </div>
  );

}
export default App;
```

CODE FOR UNCROLLED COMPONENT



CODE FOR UC WITH useRef



29. What are Higher Order Components(HOC) and What can you do with HOC?

A component that takes another component as props and return a component as a result.
It acts as a container for other components. It helps in keeping component simple and reusable. We use it when we have to apply a common logic in multiple components.

HOC can be used for many tasks like:

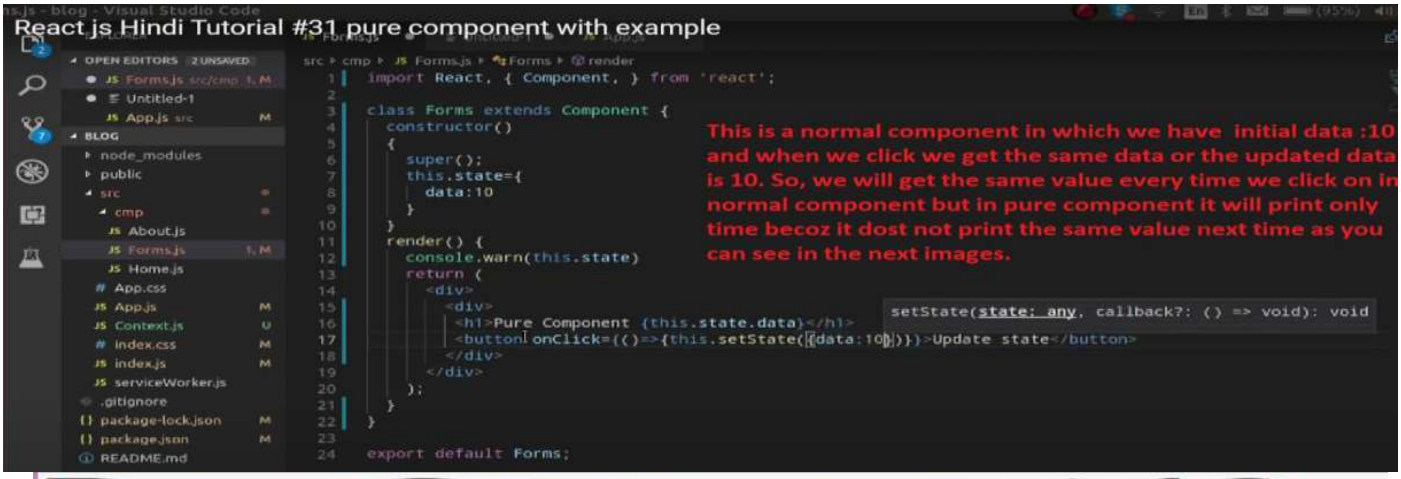
We use higher order components **to reuse logic in React apps or for reusing component logic such as first component background should be red, the second one should be blue etc.**

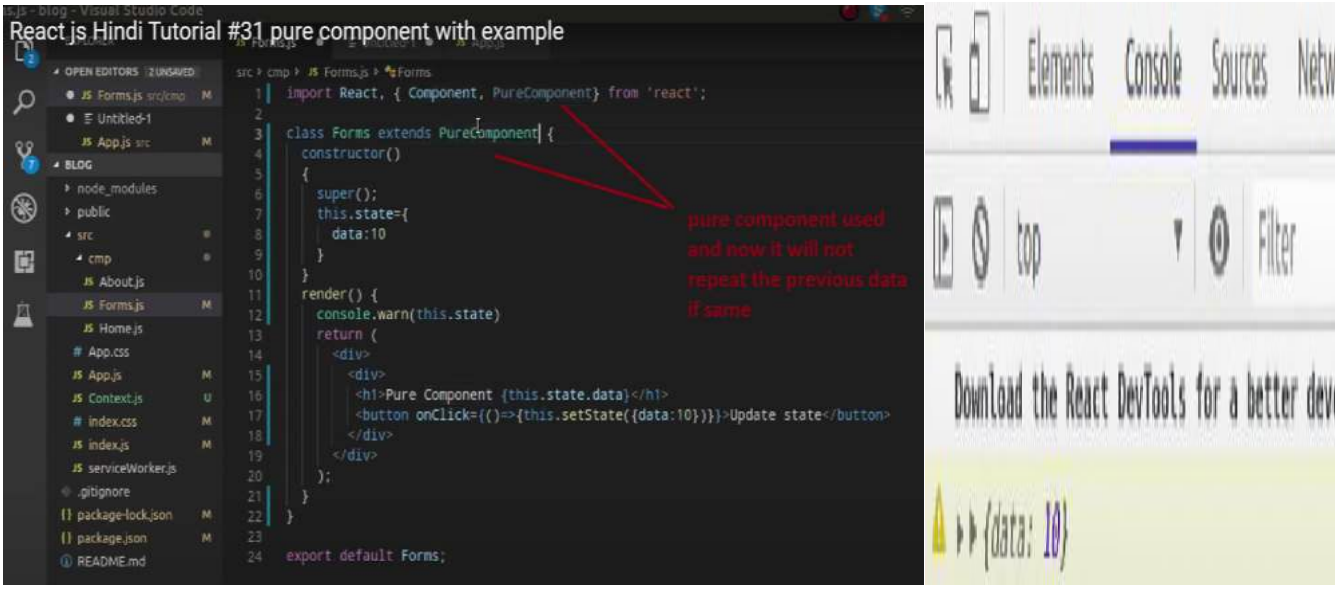
A higher-order component (HOC) is an advanced technique in React .

- **Code reuse, logic and bootstrap abstraction**
- **State abstraction and manipulation**
- **Props manipulation**

30. What are Pure Components and useMemo?

Pure component is just a feature that we need to import from React and use . It is used in class component and to use this in functional component we use a feature called useMemo which is hook.
Pure component stop the re-rendering of your component like it stops to execute again and again when it check and find that the same state is being updated which was updated last time. It actually checks the previous and next state and if the value of the state is same then it stops to execute that value. It happens as it uses shouldComponentUpdate(). We can check the same in state and props.
We should use pure component when we have the same value of state and props on clicking on something again and again so that it does not increase the load unnecessarily.





useMemo:- It is a hook and important because it enhances the performance of your application.The way we use pure component in class component. Similarly, we use useMemo in functional component. The work is same for both pure component and useMemo.

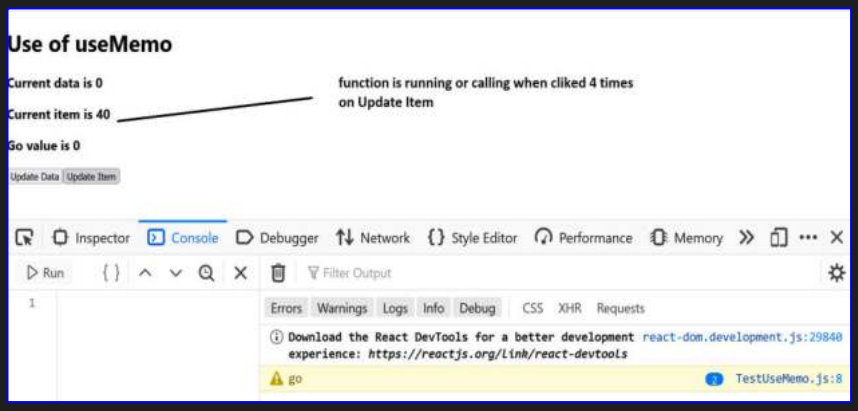
We use it because most of time when we work with state and props, our state and props same value keeps on updating with the same value and it affects the performance and to stop this we use useMemo like pure component.


```
import React, { useMemo, useState } from 'react';
function TestUseMemo() {
    const [data, setData] = useState(0);
    const [item, setItem] = useState(10);


    function go(){
        console.warn("go");   // Kindly click on console then click on warnings to see the rerendeing.
        return data*5
    }
    return (
        <>
            <h1> Use of useMemo</h1>
            <h3> Current data is {data}</h3>
            <h3> Current item is {item}</h3>
            <h3> Go value is {go()}</h3>
            <button onClick={() => { setData(data + 1) }}>Update Data</button>
            <button onClick={() => { setItem(item + 10)}}>Update Item</button>
        </>
```

```
    )
}
export default TestUseMemo;
```

Here we have taken a function go which runs on the button click on Update Data because go fuction return data*5.
So, on every click on update data function go is running, that is fine but when we click on the update item button this function go should not run because it is not linked to the updated item. However, it is running on every click of Update Item which is a problem, Here, we will use useMemo to stop this. To stop such issues, we first need to import useMemo then write useMemo() before the return function and put the function inside the parenthesis of useMemo(function) and then we may store it as let x = useMemo(function) and after function use comma and then use dependency like , [data] which means it(function) will run only data updates because data is taken in the function. Now when we click on Update item, function will not run.

```
import React, { useMemo, useState } from 'react';
function TestUseMemo() {
    const [data, setData] = useState(0);
    const [item, setItem] = useState(10);


let x = useMemo(function go(){
              console.warn("go");
             return data*5
}, [data]   )
    return (
        <>
            <h1> Use of useMemo</h1>
            <h3> Current data is {data}</h3>
            <h3> Current item is {item}</h3>
            <h3> Go value is {x}</h3>
            <button onClick={() => { setData(data + 1) }}>Update Data</button>
            <button onClick={() => { setItem(item + 10)}}>Update Item</button>
        </>
    )
}
export default TestUseMemo;
```



React Router - React Interview Questions

1.What is React Router or Routing?

React Router is a routing library. It is used to create routes in a React application.
Whenever we have to make pages in React we use React Router. We have different Router for Angular and Vue.js . We can make different pages without Router as well but we need to write too much codes and logic. We can also use link but it will refresh or reload the page. So, we don't use link in React. React Router is a powerful routing library.
In order to install React Router, Go to new terminal in vs and write npm i react-router-dom
Here, 'I' means install.
To check if React Router is installed, we need to go to package.json and check under dependencies.
"react-router-dom" : "6.2.2", will be written.
 6 is major update,, 2 minor update, 2 package.
In order to use router we need to import the router:-

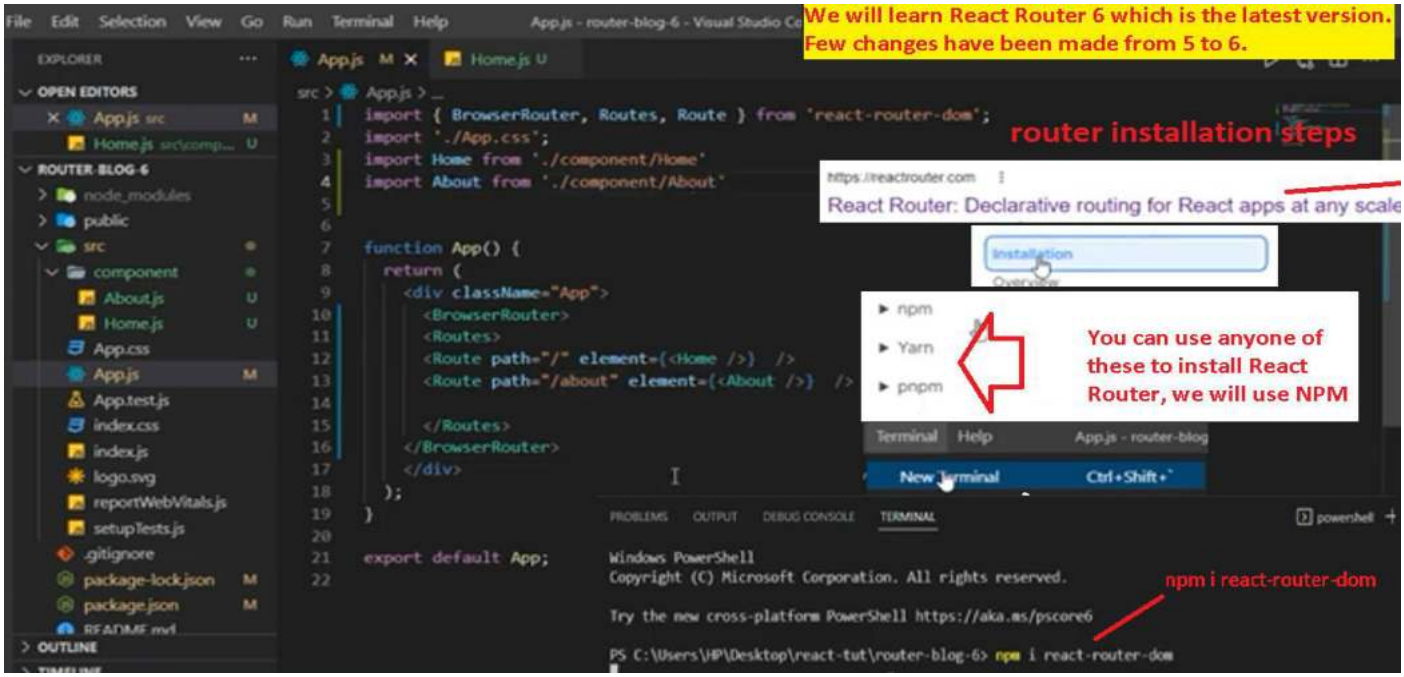
```
import {BrowserRouter} from 'react-router-dom'
```

 and make a wrapper or box under return and inside this all links will be added for different pages. We need to make a wrapper or box and inside the box, React Router will work not outside. We need to make two wrappers and in the third one page will be called.

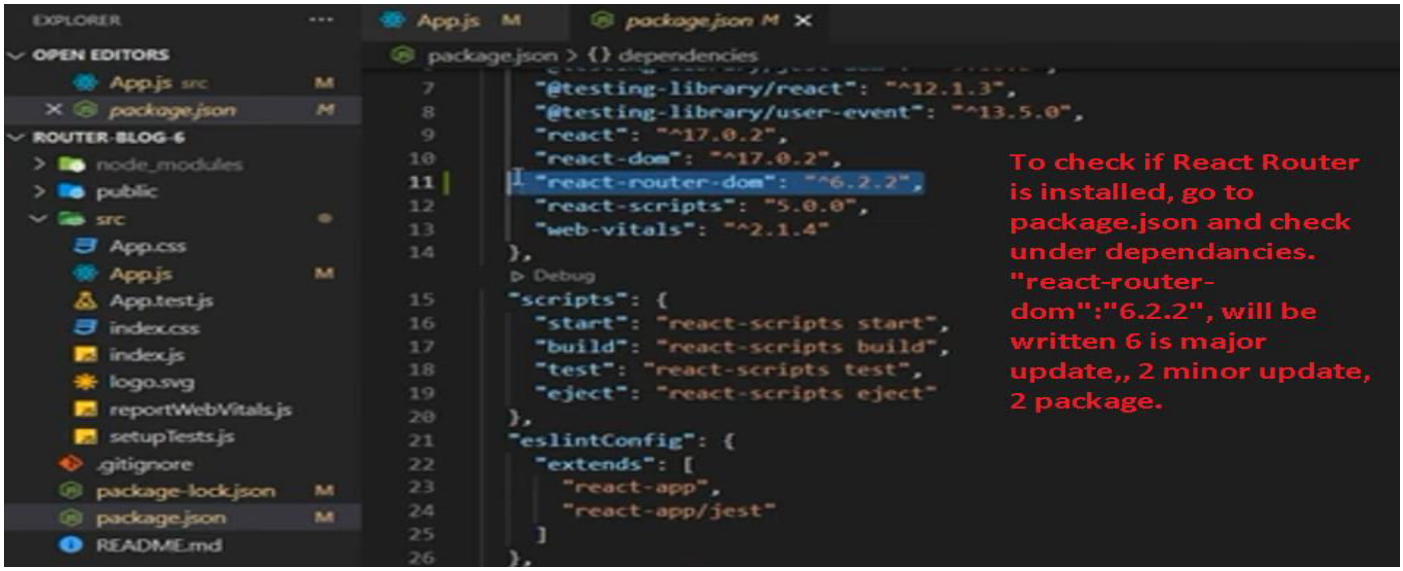
2. Why do we need to React Router and how to implement it?

We need to make different pages which will look like single-page web applications as does not reload the entire web page.
Enables multiple views in a single application by defining multiple routes in the React application.

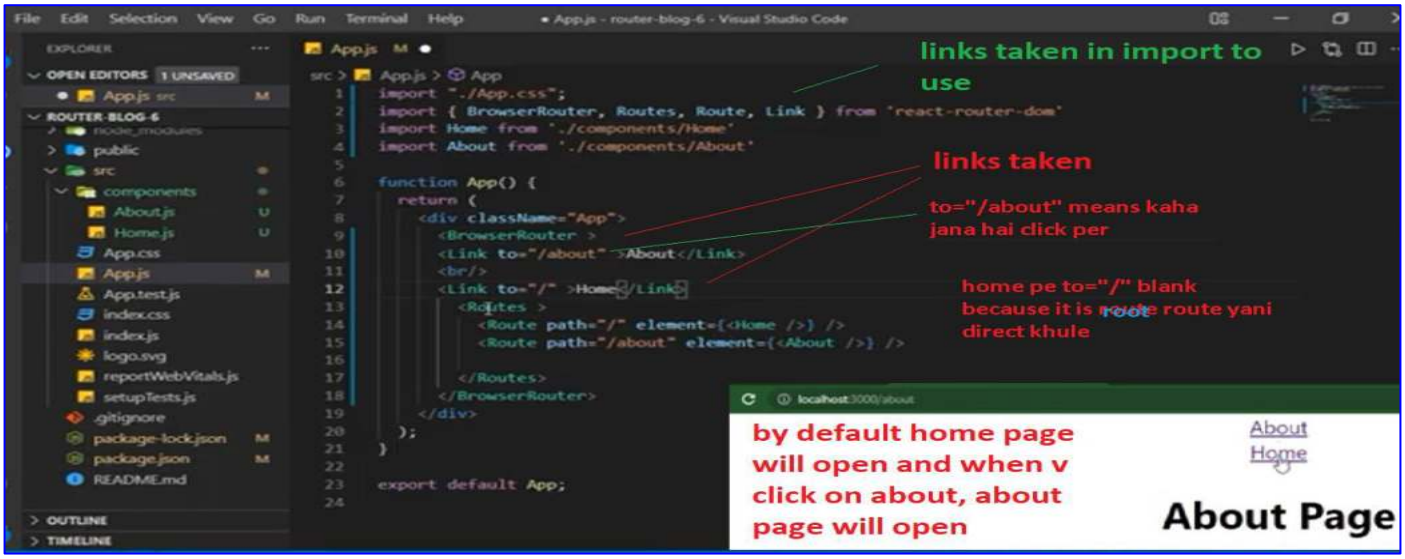
Here, by default home page will open(as path="/" taken) and if we write /About in the URL then About page will display.



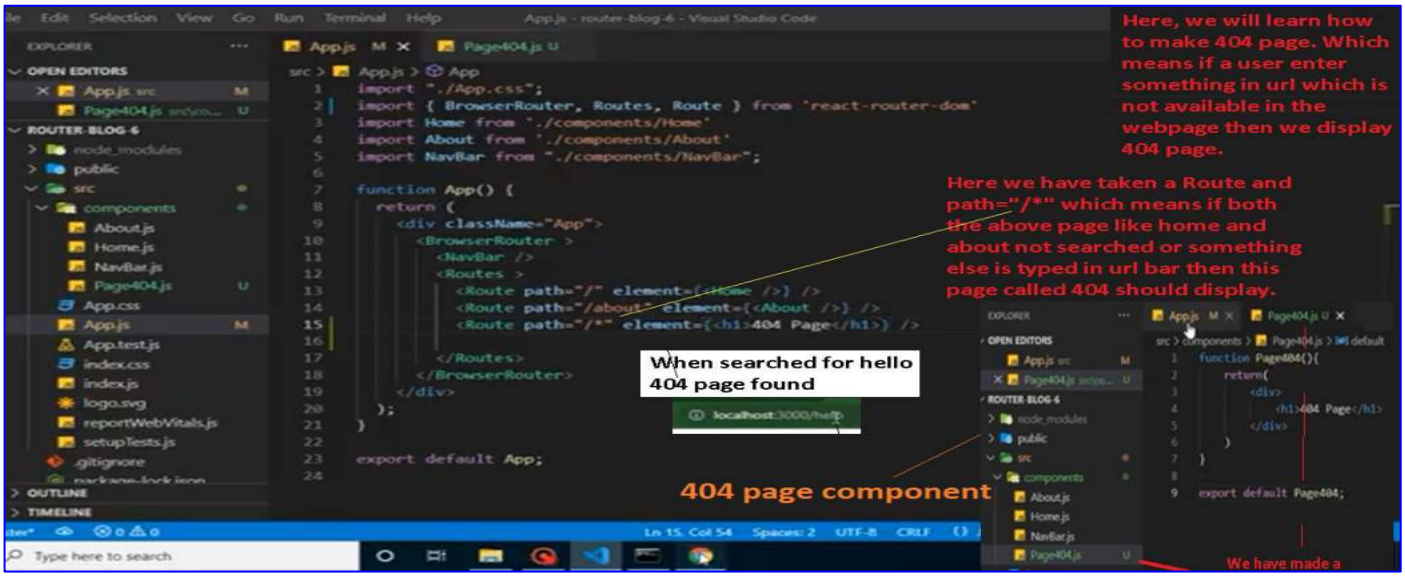
To check if router installed or not



We use links instead of href in React becoz href will reload the page



PAGE NOT FOUND

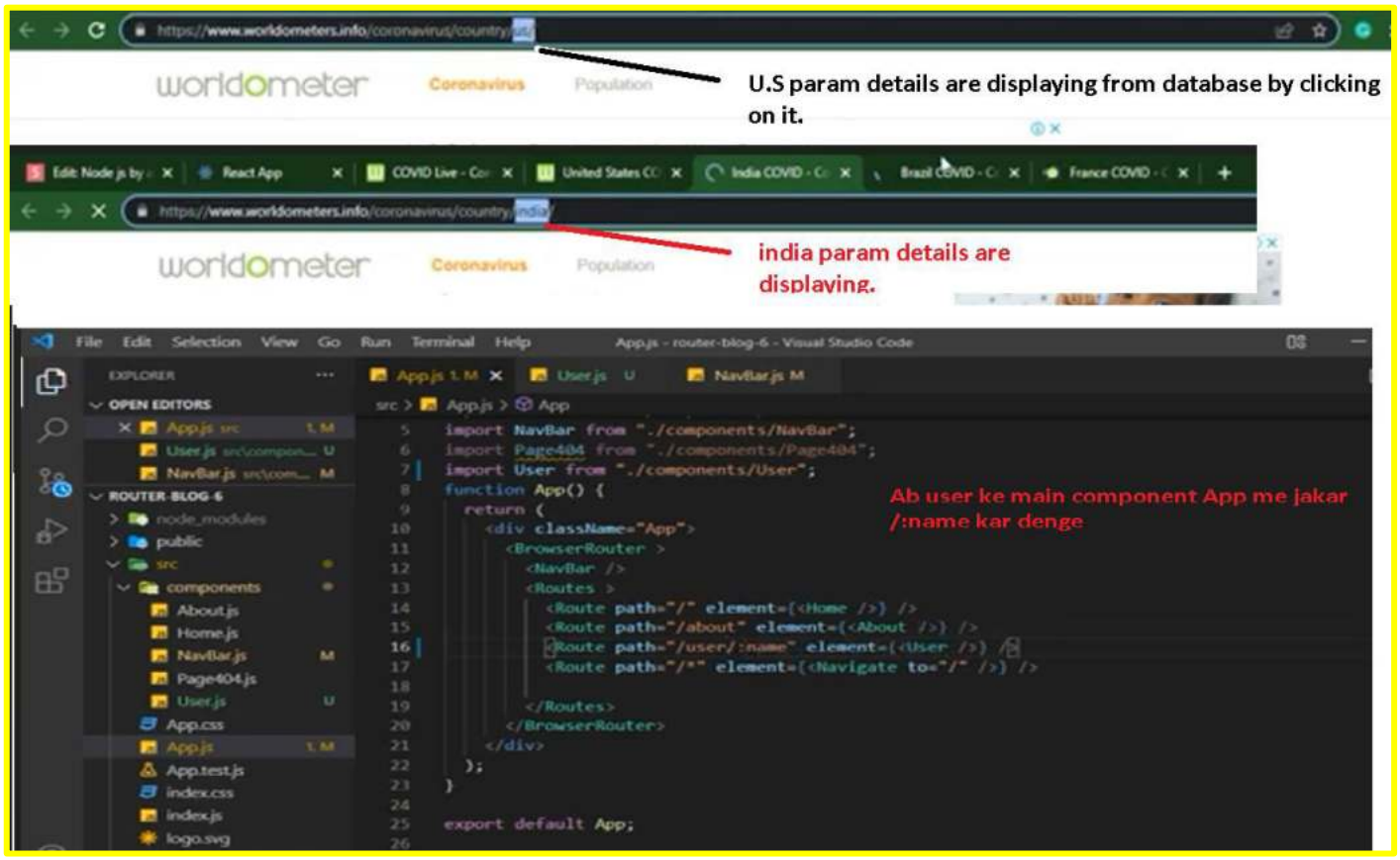


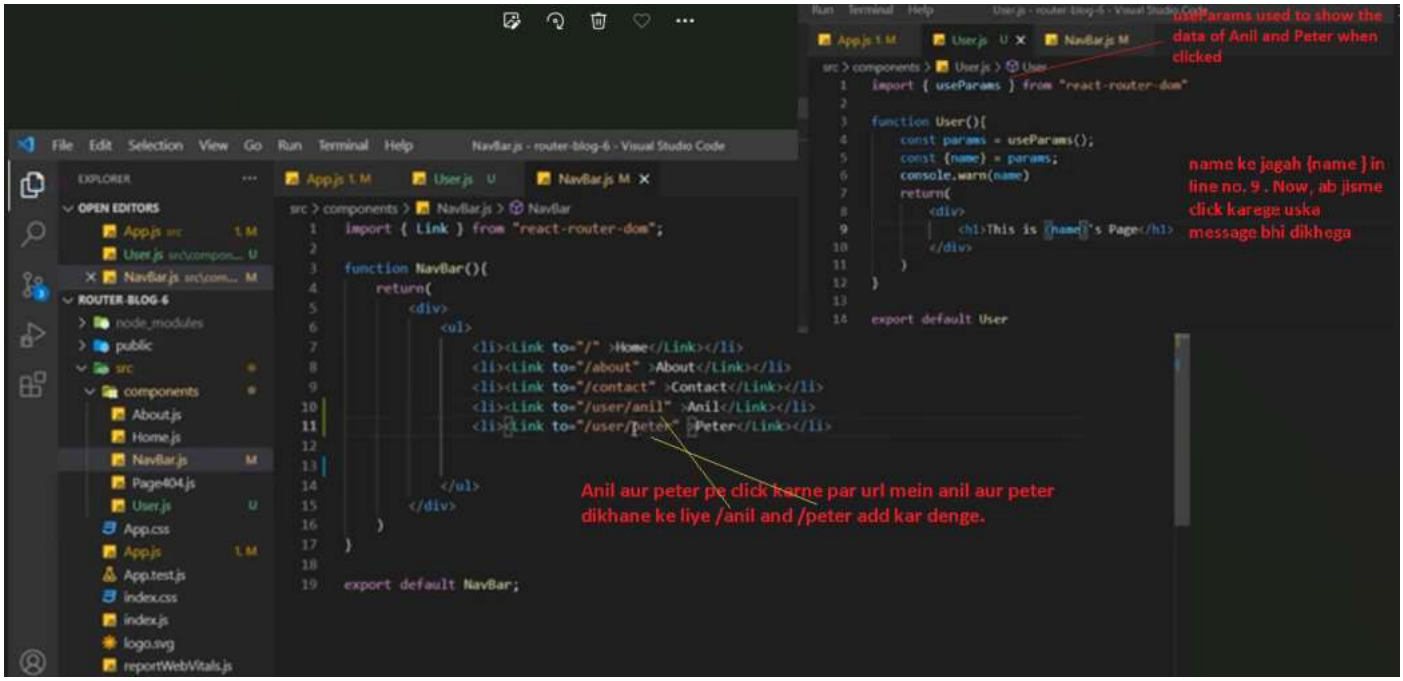
3. How is React routing different from conventional routing?

SN	React Routing	Conventional routing
1.	React Routing has single HTML page	Each view is a new HTML file
2.	The user navigates multiple views in the same file	The user navigates multiple files for each view
3.	The page does not refresh since it is a single file	The page refreshes every time user navigates
4.	Improved performance	Slower performance

4. Define params and why do we need this with Route?

Whenever we need dynamic data, the data that we need to send to the data base and on the basis of that data we have to get the details on the entire page then we use param or parameter in React Router. Suppose we have a list of users like Anil, Sarfraz, Johny etc. and on click of user like Sarfraz , we have to open a new page and show various details of Sarfraz. Since it will be very tough to make page for all the users if we have many users like 1000. In this case, we send parameters and on the basis of that parameter, data matches from the database and we get details from the database of that particular user.





OR DYNAMIC ROUTING

32. Explain how lists work in Reat.

List is used to make a list from a given array i.e array= [10, 12] or from a given array of objects like arrayOfObject=[{name:"sarfraz", age=25 }, {name="Alam", age= 26}] or Lists are used to display data in an ordered format and the traversal of lists is done using map() function.
In another word, Map method is used to iterate or traverse through the each element of an array and provide a new array. It has a call back function as a parameter that runs for each element available in the array and return a new array.
Syntax- map(callback(cuurentValue, index, array)
Arr = [10, 20]
map(num, index)=>{return num}

```
<script>
      let arr = [10, 20, 30, 40];
      let a = arr.map((num) => {
         return document.write(num)    // 10 20 30 40
      })

      document.write(a);
   </script>
   <script>
      let arr1 = [10, 20, 30, 40];
      arr.map((num, i) => {
         document.write(`${i} : ${num} ` + "</br>");  // 0: 10 1:20 so on
      })
   </script>
```

Why is there a need for using keys in Lists? OR Define key. Each child in a list should have a unique "key" prop. Explain this.

A "key" is an attribute of string that we need to include when we create lists of elements. It is used to identify which items have changed, added or removed/ deleted from the lists.

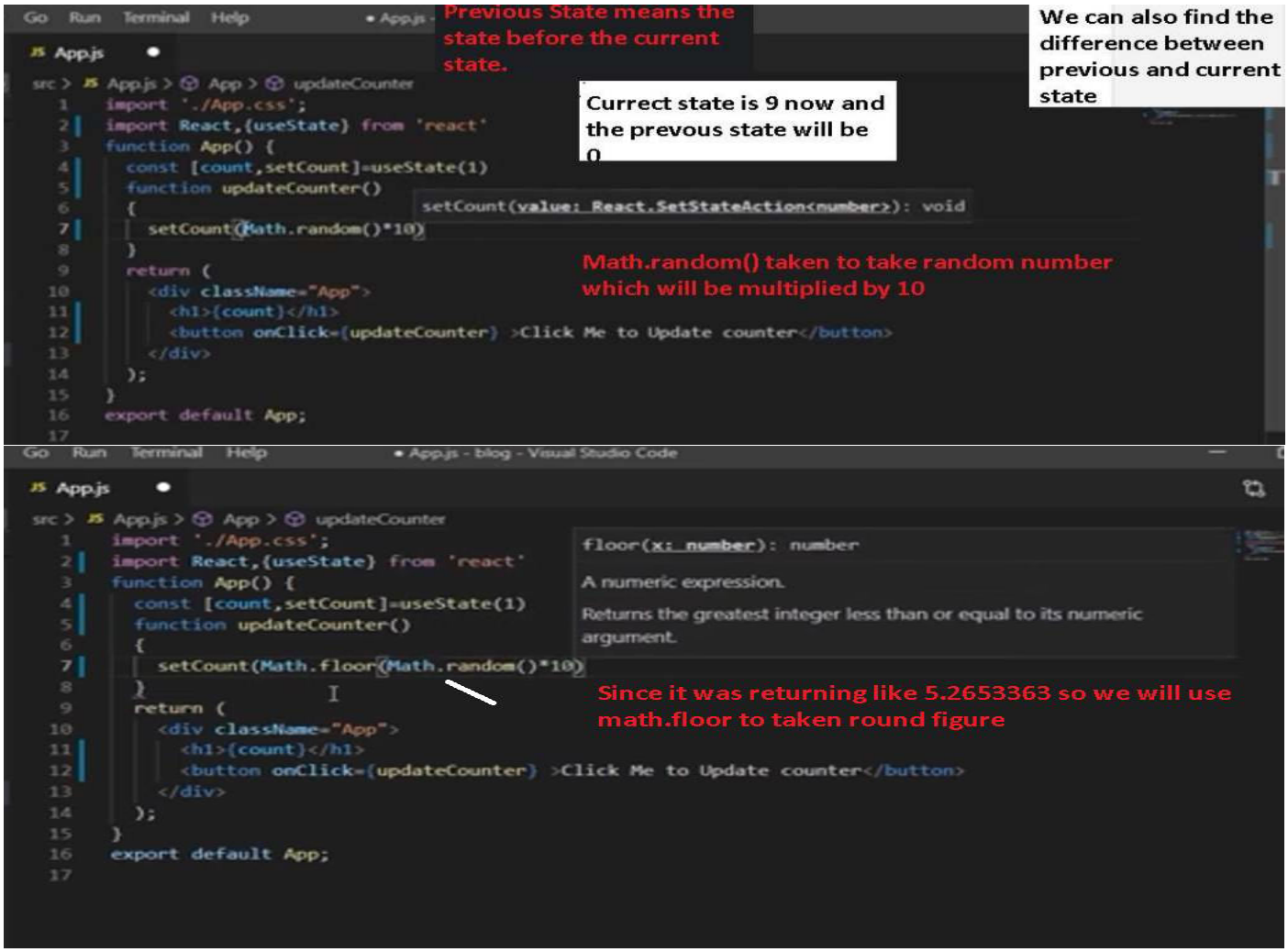
It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered

Whenever we render the list the virtual DOM needs a unique key to identify in which children changes have been made and then update with the only child on which changes have been made. Here you can see <tr> used twice so two children were

created and because of that virtual DOM is unable find on which child changes were made and give error so avoid this error we need pass a unique key to make both the <tr> different.

```
src > JS App.js > App > users.map() callback
  function App() {
    const users = [
      { name: 'Anil', email: 'anil@test.com', contact: '111' },
      { name: 'Burce', email: 'bruce@test.com', contact: '222' },
      { name: 'Peter', email: 'peter@test.com', contact: '333' },
      { name: 'Sam', email: 'sam@test.com', contact: '444' },

    ]
    return (
      <div className="App">
        <h1>List With Bootstrap Table</h1>
        <table>
        <tr>
            <td>Name</td>
            <td>Email</td>
            <td>Contact</td>
          </tr>
        {
          users.map((item,i)=>
          <tr key={i}>
            <td>{item.name}</td>
            <td>{item.email}</td>
            <td>{item.contact}</td>
          </tr>)
        }
        </table>
```

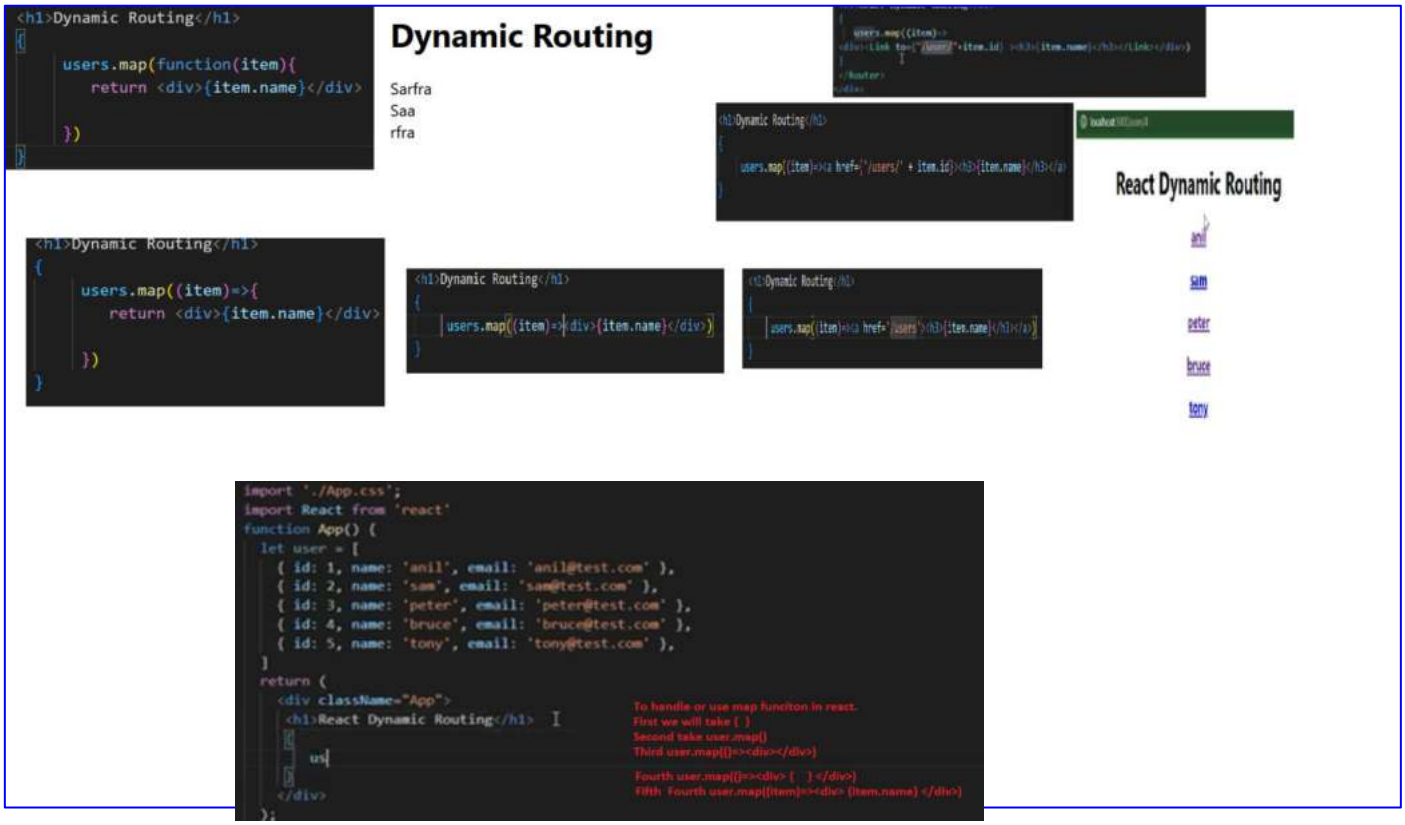
33. Previous state



34. Why do we use map not for loop in react?

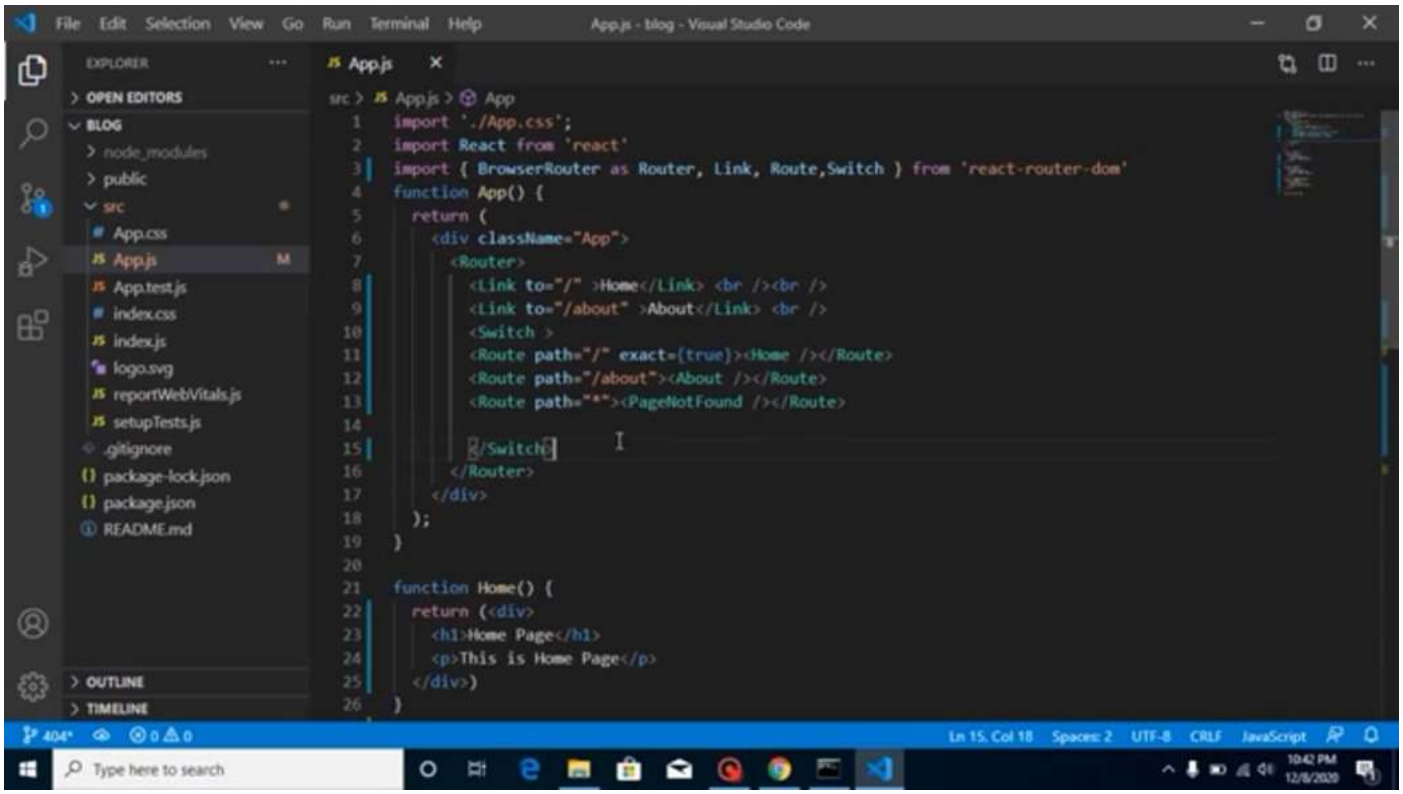
For loop will not work under return

When we don't know how many routes we have make for example we have user or product list and you have 10 products in product list then we will make 10 routes but what if we add more products later then how you will manage the routing in that and then we will have to use dynamic routing.



5. Why is switch keyword used in React Router v4?

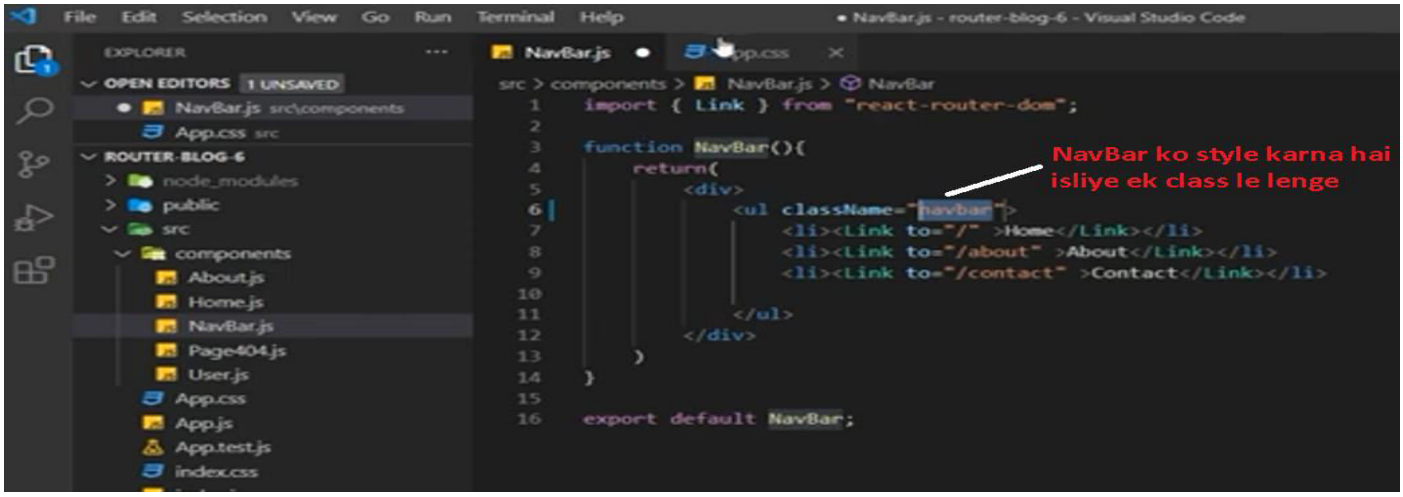
Although a ** ** is used to encapsulate multiple routes inside the Router. The 'switch' keyword is used when you want to display only a single route to be rendered among the several defined routes. The **<switch>** tag when in use matches the typed URL with the defined routes in sequential order. When the first match is found, it renders the specified route. Thereby bypassing the remaining routes. Switch is used as wrapper and exact will make the default page as it is exact true.



49. List down the advantages of React Router.

STYLE IN ROUTER

We need to follow the steps to style the route:-


File Edit Selection View Go Run Terminal Help
• NavBar.js - router-blog-6 - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NavBar.js src\components
App.css src
ROUTER-BLOG-6
node_modules
public
src
components
About.js
Home.js
NavBar.js
Page404.js
User.js
App.css
App.js
App.test.js
index.css
NavBar.js
App.css
src > components > NavBar.js > NavBar
import { Link } from "react-router-dom";
function NavBar(){
return(

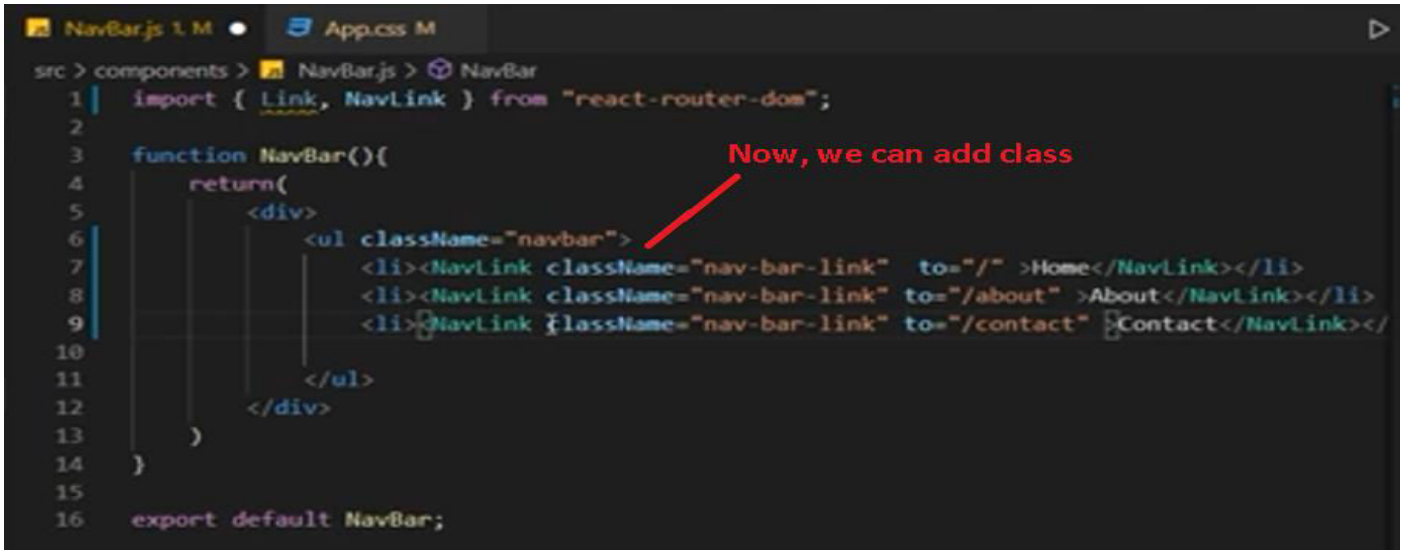
<ul className="navbar">
<Link to="/" >Home</Link>
<Link to="/about" >About</Link>
<Link to="/contact" >Contact</Link>

)
}
export default NavBar;
NavBar ko style karna hai isliye ek class le lenge


NavBar.js 3, M
App.css M
src > components > NavBar.js > NavBar
import { Link, NavLink } from "react-router-dom";
function NavBar(){
return(

<ul className="navbar">
<NavLink to="/" >Home</Link>
<NavLink to="/about" >About</Link>
<NavLink to="/contact" >Contact</Link>

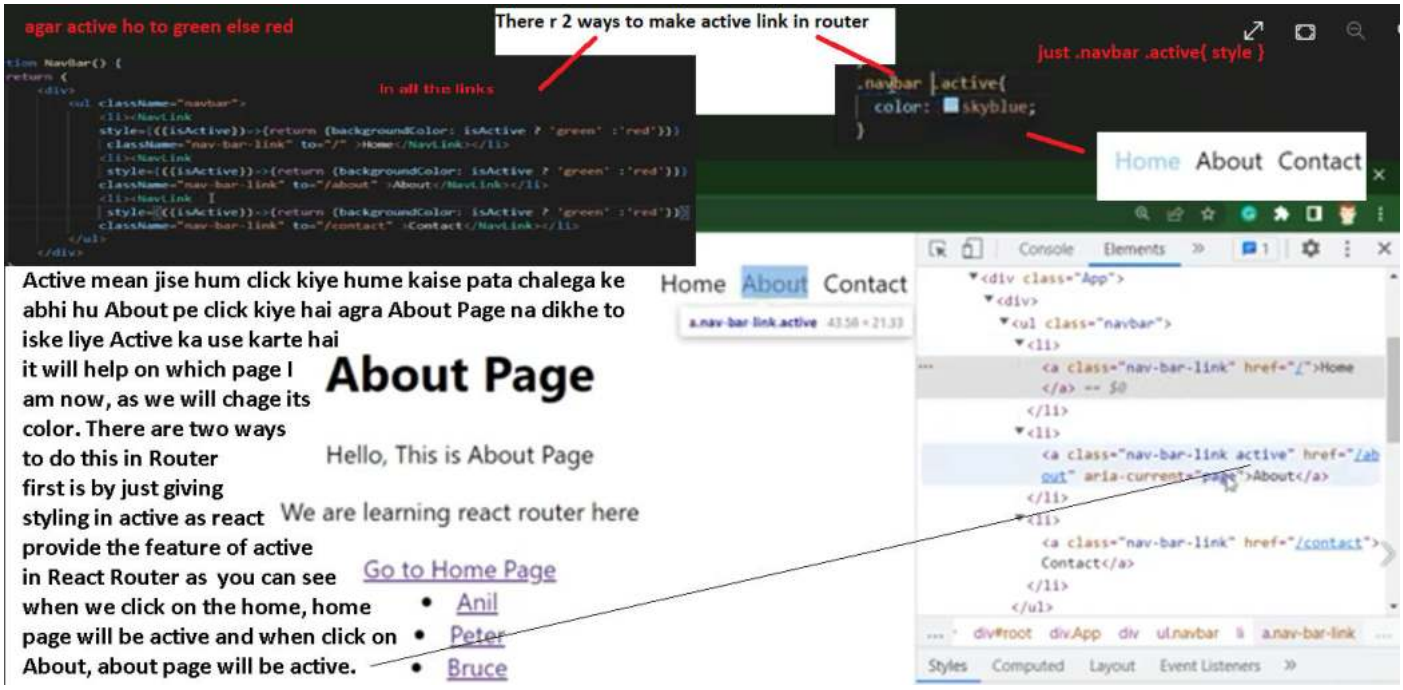
)
}
export default NavBar;
Since link pe class nahi laga sakte hai jo ke react ke rule ke against hai isily lik ko NavLink se replace kar denge


NavBar.js 1, M
App.css M
src > components > NavBar.js > NavBar
import { Link, NavLink } from "react-router-dom";
function NavBar(){
return(

<ul className="navbar">
<NavLink className="nav-bar-link" to="/" >Home</NavLink>
<NavLink className="nav-bar-link" to="/about" >About</NavLink>
<NavLink className="nav-bar-link" to="/contact" >Contact</NavLink></

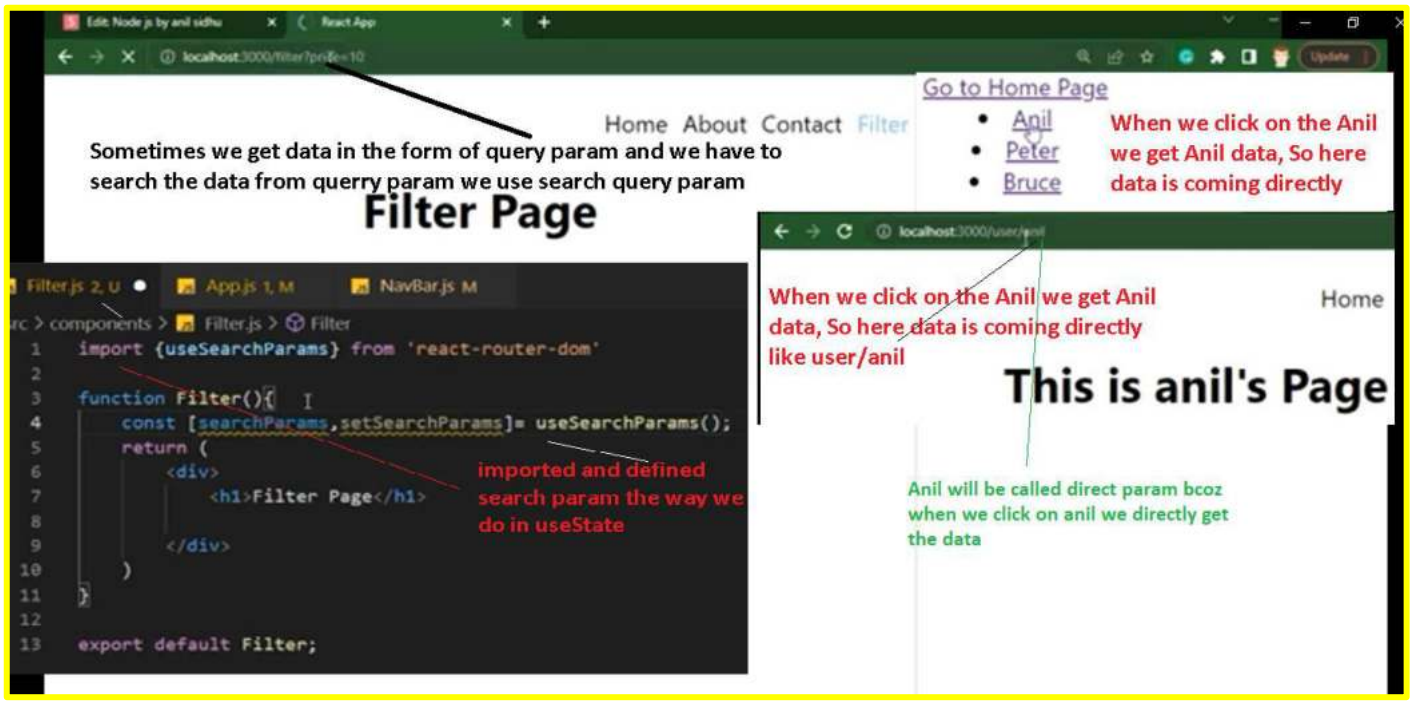
)
}
export default NavBar;
Now, we can add class

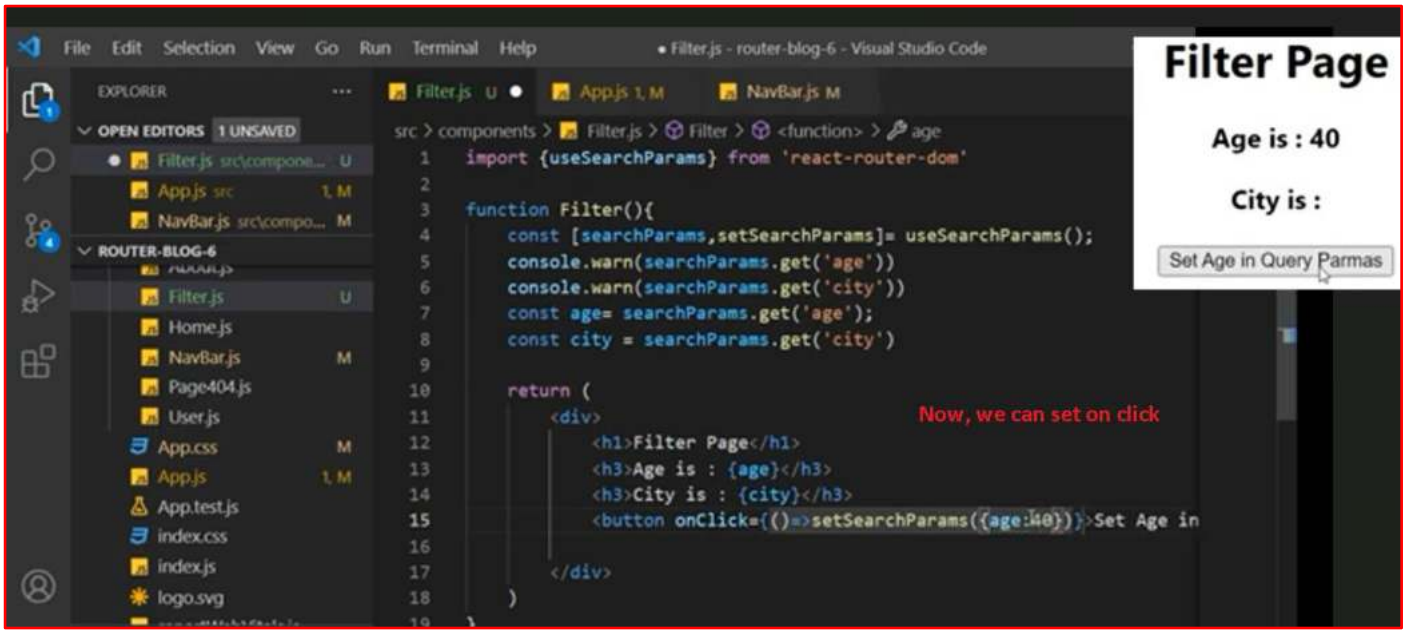
ACTIVE LINK IN ROUTING OR ROUTER



6. What is searchParams Hook.

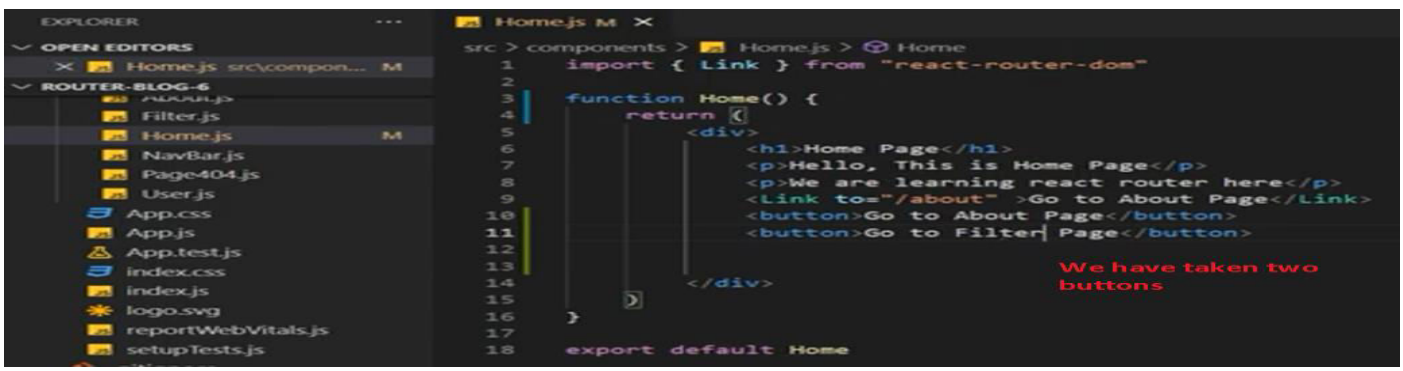
SearchParams Hook is hook of React Router and it is used to search the query param and get the data of that particular searched data from database. Like if url is like www.google.com?age=10 and we have to get the data of 10 which is a query param then we will use searchParams Hook. We can get and set age=10(get) and age=20(set). www.google.com/users/sarfraz. Here sarfraz is direct param. Here is the step given to get and set the searchParams. We can use param insead of params for single param.

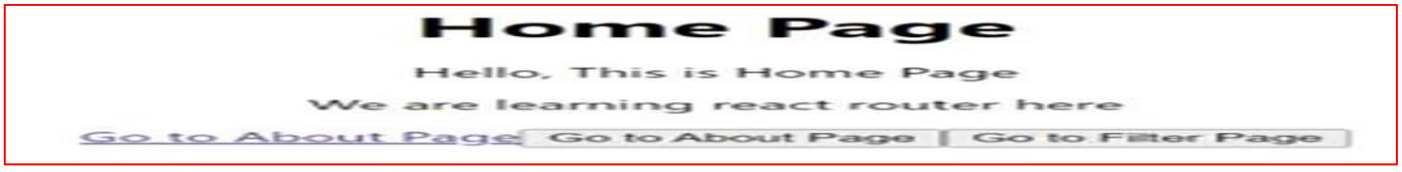




7. What is useNavigation? Which hook will you use to navigate on button click and function call.

We use useNavigate to go to the other page on button click or on function calling.





```
import { Link, useNavigate } from "react-router-dom"

function Home() {
    const navigate = useNavigate();
    return (
        <div>
            <h1>Home Page</h1>
            <button>Go to About Page</button>
            <br />
            <button>Go to Filter Page</button>
        </div>
    )
}

export default Home
```

We will use useNavigate to navigate on button click and fuction call. So, imported and defined in line no. 1 and 4

```
import { Link, useNavigate } from "react-router-dom"

function Home() {
    const navigate = useNavigate();
    return (
        <div>
            <h1>Home Page</h1>
            <p>Hello, This is Home Page</p>
            <p>We are learning react router here</p>
            <Link to="/about" >Go to About Page</Link>
            <br />
            <button onClick={()=> navigate('/about')}>Go to About Page</button>
            <br />
            <button>Go to Filter Page</button>
        </div>
    )
}
```

arrow method used to call the navigate on clik and about to open about page

navigate(to: To, options?: NavigateOptions): void

Navigation on calling the function

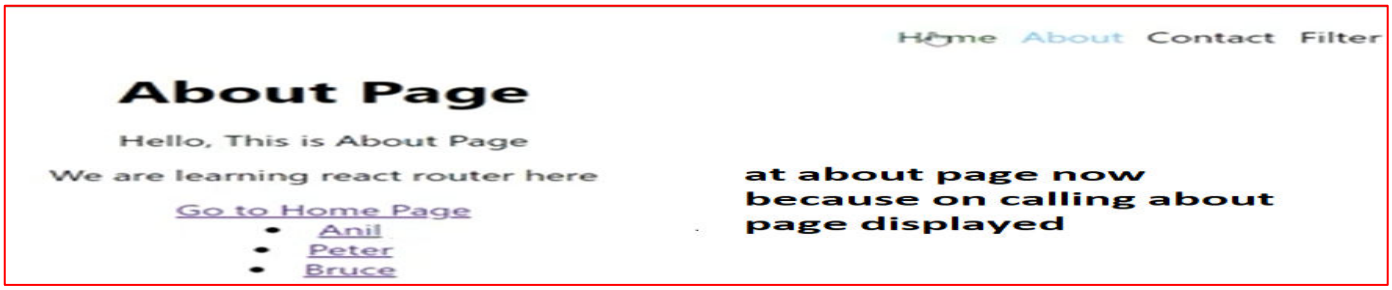
```
import { Link, useNavigate } from "react-router-dom"

function Home() {
    const navigate = useNavigate();
    return (
        <div>
            <h1>Home Page</h1>
            <p>Hello, This is Home Page</p>
            <p>We are learning react router here</p>
            <Link to="/about" >Go to About Page</Link>
            <br />
            <button onClick={()=> navToPage()}>Go to About Page</button>
            <br />
            <button onClick={()=> navToPage()}>Go to Filter Page</button>
```

Now, we will navigate on fuction call so two functions taken

```
import { Link, useNavigate } from "react-router-dom"

function Home() {
    const navigate = useNavigate();
    const navToPage =()=>{
        navigate('/about')
    }
    return (
        <div>
            <h1>Home Page</h1>
            <p>Hello, This is Home Page</p>
            <p>We are learning react router here</p>
            <Link to="/about" >Go to About Page</Link>
            <br />
            <button onClick={()=> navToPage()}>Go to About Page</button>
            <br />
            <button onClick={()=> navToPage()}>Go to Filter Page</button>
```

navigate(to: To, options?: NavigateOptions): void

About taken to show on calling the function



8. Nested Routing. Do we need to use <outlet> to show the message nested routing?

Refer notes , Yes(it will only show the route like user/sarf but sarf content will not display and for that we need to import and use <outlet> in which making nested routes.

9. useLocation hook in React Router.

In React Router, whenever, we go to another route from one route and check what we have got in this route when we load that route then we can use useLocation. useLocation gives all the data like params, hash, state and search in an object.

For example, when we click on About, we go to about page so we are routing from about to about page. Now, In about page we have also a route(link) Anil. When we click on Anil, we route(go) from About to Anil. Now, if we want to check that from About page, in Anil page what all data we have got like param, hash, state etc then we will use useLocation.

Whenever we go to another route from one route then a unique is generated called key.

Refer notes for more explanation.

10. Define protected route.

protected route is not a feature of React Router but it is a logical part. We use protected route to redirect to the user to log in page whenever any user tries to visit your website before log in .
In short, it is used to authenticate the account.

Here, I can visit to the About page when I click on login but now, I want that I can't visit to About page until I login which means whenever I click on Home, About, Filter etc. It will take me to the log in page where I need to put the credential and then I can navigate.

11. Why we are making protected route?

We can simply go to each page put condition like when click on home page take to the log in page and ask to authenticate but we will not do this because if we have to many pages to buttons then it will be difficult so we will use protected route in which we will keep all the buttons/pages in one common component and this is why we use this.

REACT STYLING QUESTIONS

1. How do you style React components? AND DEFINE STYLE COMPONENT

There are three ways to style React components.

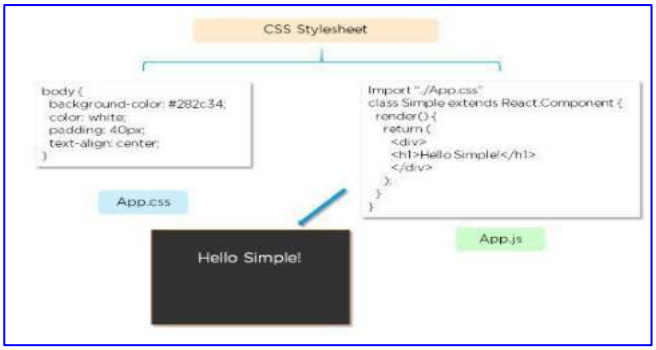
1. **Inline Styling (In the tag using style)**
2. **External Styling (Making external sheet with extension name.css)**
3. **CSS Module (Making external sheet with extension name.module.css)**

INLINE STYLE SHEET

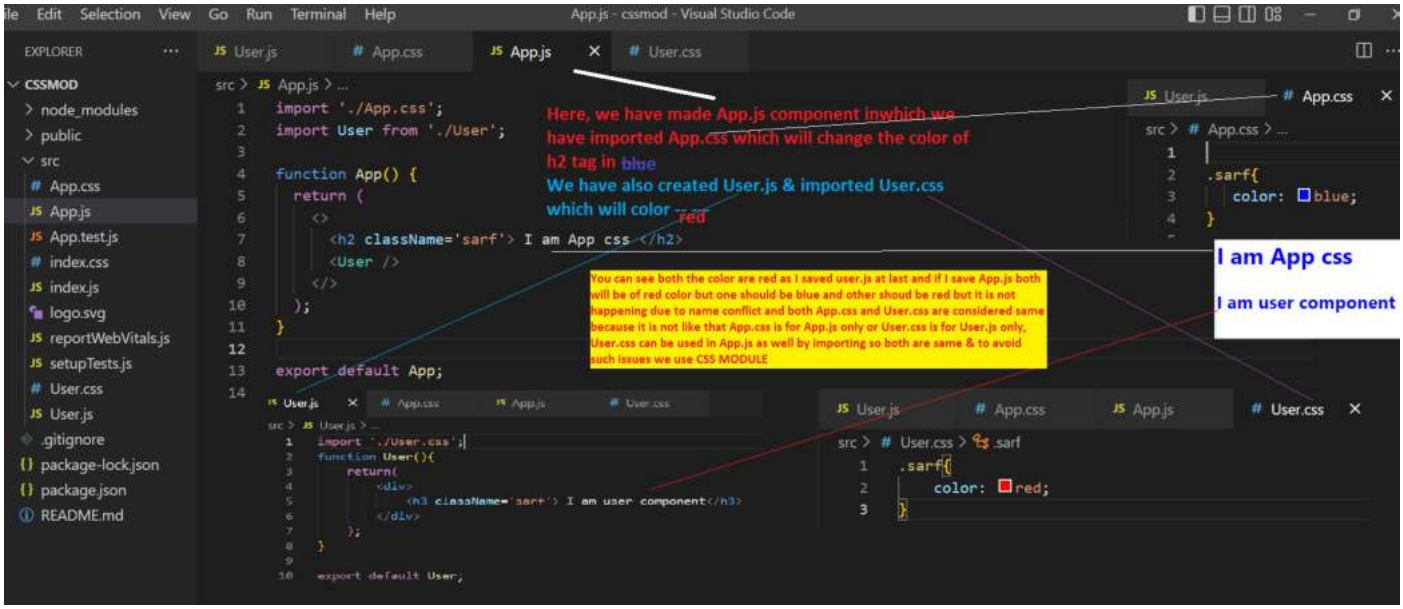
```
import React from "react";
function InlineStyle(){
    return(
        <>
        <h2 style={{color:"blue"}}> I will be used for inline style</h2>
        <h2 style={{color:"blue", background:"red", fontSize:30}}> I will be used for inline style</h2>
        </>
```

```
    )
}export default InlineStyle;
```

- *JavaScript Object* *CSS Stylesheet*



2. Explain the use of CSS modules in React.



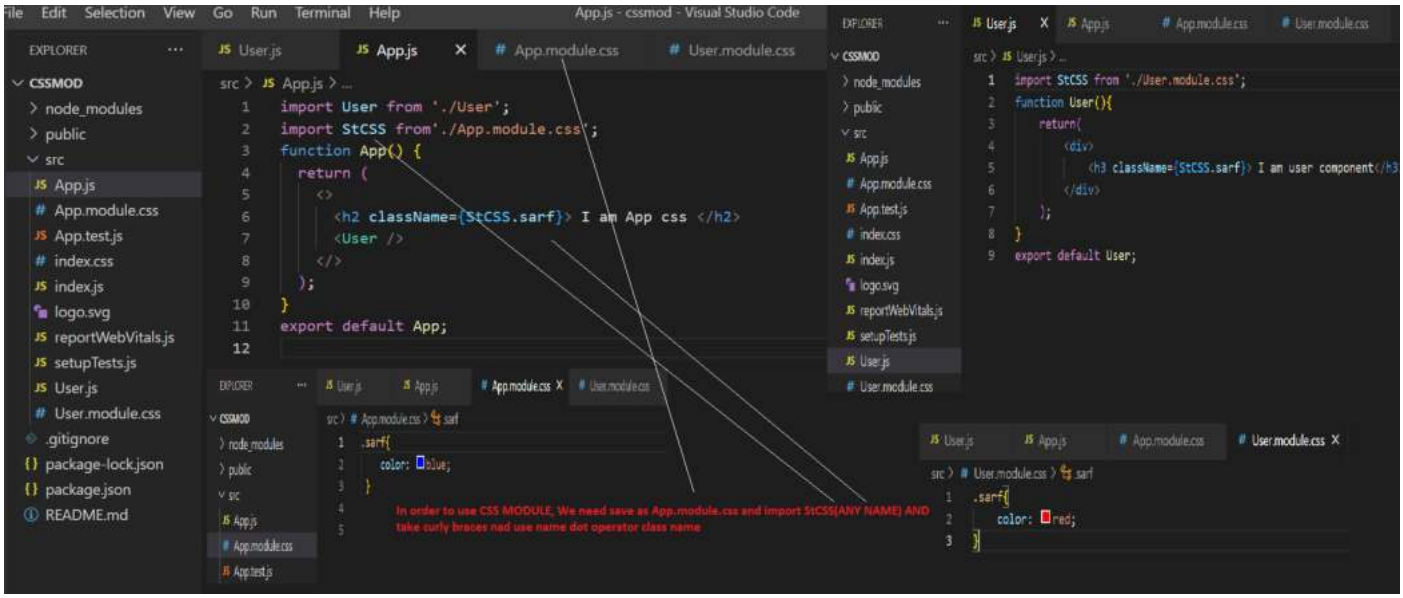
We need CSS modules in React because we can't not use pseudo classes in inline CSS. So we need external style sheet for this. However, in external style sheet creates naming conflicts.

CSS modules allow us to use the same class name in different files without naming conflicts issues. In CSS modules name classes are local scope so will be used where it is imported only not globally like we had in external sheets and that was why naming conflicts were happening.

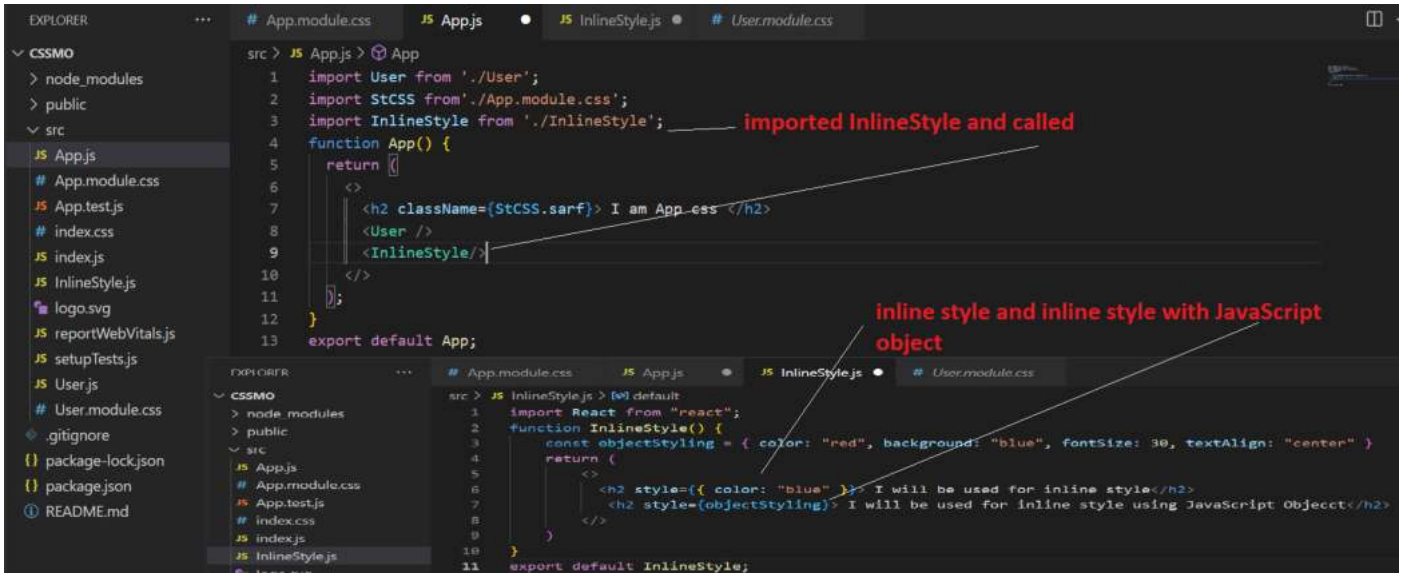
Syntax of CSS MODUE:- Name.module.css (name can be anything)

It will work only if you use react-scripts@2.0.0 and higher.

We can use both CSS MODE AND REGULAR CSS TOGETHER.

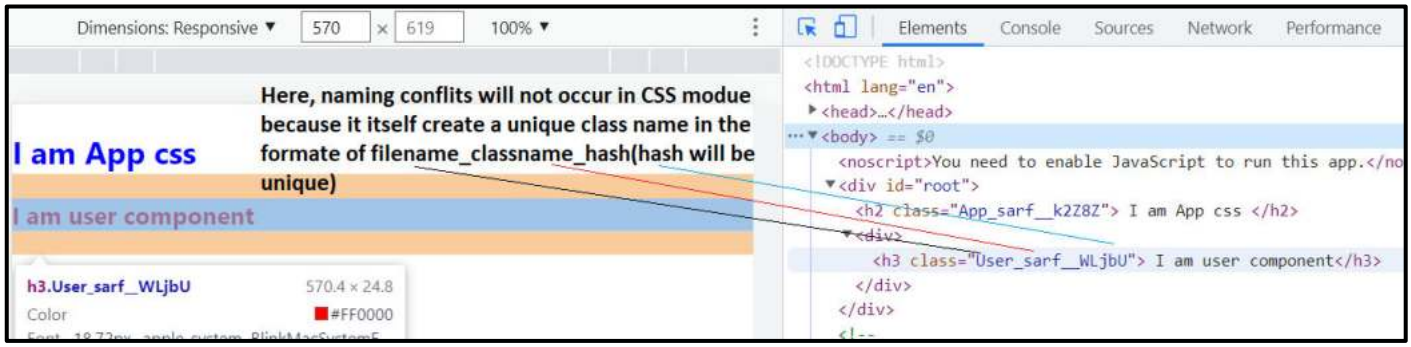


Inline style uses



C:\Users\pione\OneDrive\Desktop\FED\React.js\cssmod

- The CSS module file is created with the **.module.css** extension
- The CSS inside a module file is available only for the component that imported it, so there are no naming conflicts while styling the components.



https://www.youtube.com/watch?v=7onGdDzU-Fg

Few advantages are:

50. Define Postman and how would you store user's input data in the database?

Refer Notes

51. **Difference between PUT and PATCH.**

In PUT Method, we send many data but in PATCH Method, we send a particular thing or single thing that we have to change to update.
But some other definition says:-
In PUT Method, we send and modify the data but PATCH are just instructions to modify the data and in this data will not be changed as it gives just instruction that you have to change this data.

52. Can we delete using PUT method?
It depends on the back-end team, if they have used PUT method to delete, PUT can be used to DELETE. Method name is for our understanding and name can be changed by the back-end team(code has been written in such a manner that even PUT can delete) to do some actions like GET, POST, PUT OR DELETE

Redux Interview Questions

35. How to transfer data from parent to grand child or grand child to parent directly?

There are three ways to do it.

context API(It is the inbuilt in React)

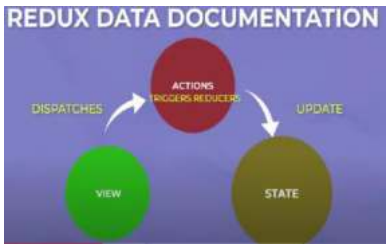
Redux (It is an external library that we need to install in React and use)

Redux toolkit

What is Redux(State Management System)?

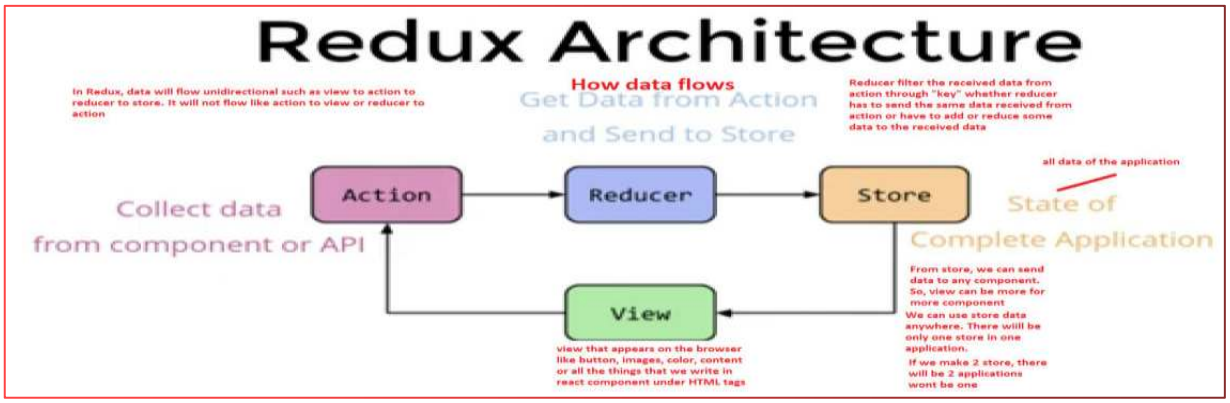
Redux is an open-source, JavaScript library and it is also called state management tool, here state means application data managed by the Redux of your application or project. Redux is like a container where you can store your whole application data. It is not a kind of a database but a kind of an array where you can store your whole application data. For example, I have a project related to the employee management system, we can store all the data of employees in that container.
Here in the container, we can store user's current data, API data so that you will not have to call the API again and again. The Redux data will disappear when you refresh but if you have some sort of data that you want to keep even after refreshing the page like login & logout details of users, token etc. Then you will have to keep them locally not in Redux
In Redux, it becomes easy as we keep all the data in a container and take out whatever, we require.
In simple word, It is like a store room at our home from where we can put anything and then use anywhere.
Redux is available inside the application in the front-end where we have react apps.

Describe basic data flow in Redux or How does Redux work or How the update get back into my UI?



If we talk about Redux data-flow, we have view, aactions, state and then circle and this is how data flows in one direction from view to actions and action to Reducer and then Reducer to state and then state to UI.
View actually dispatches actions which are event handlers and actions trigger reducers which update the state and the state update the UI.

What are the components of Redux?



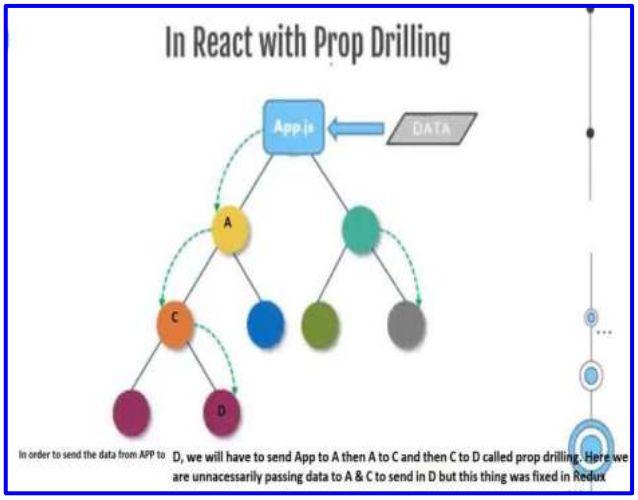
How it works

We have a wrapper of Redux that we get after installing the Redux makes a wrapper of the entire application which helps to control the application data through Redux.

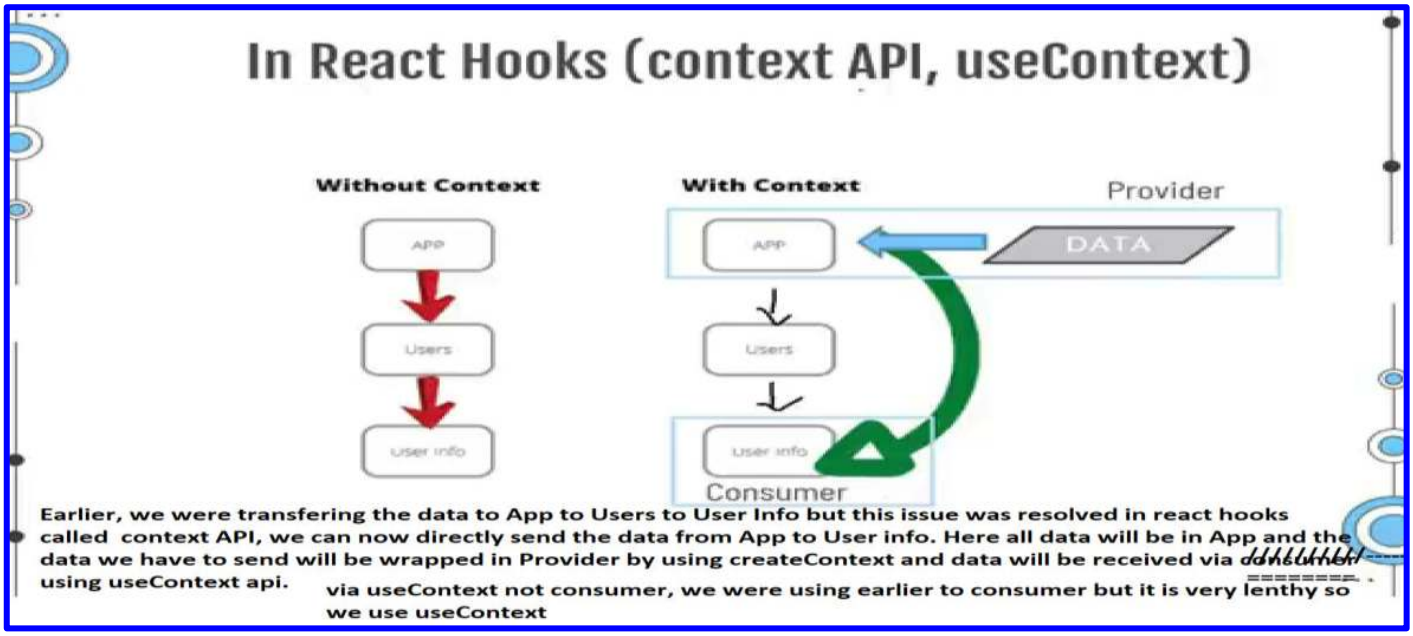
Why we use it

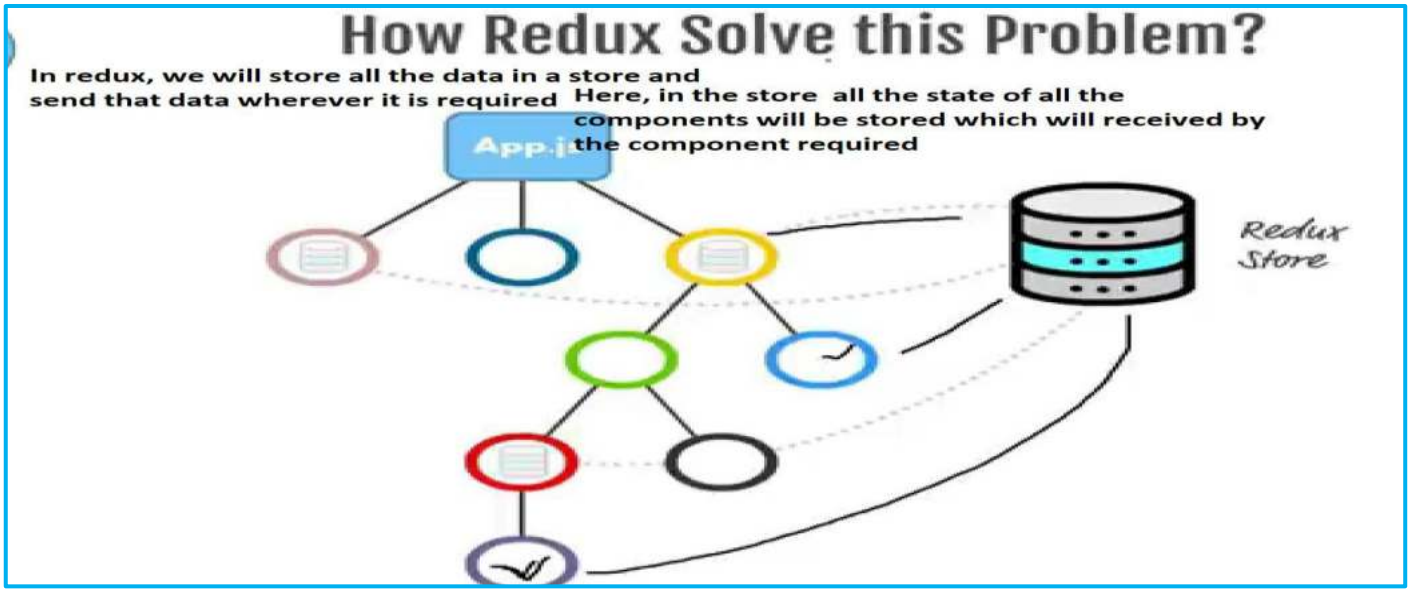
We know that React creates component based applications and it becomes very tough to communicate from one component to another component in React. For example if we are creating any e-commerce website in which we have to create a checkout page. It becomes very tough to show the check-out amount on selecting end number of of items but in Redux, it becomes easy as we keep all the data in a container and take out whatever, we require.

In fact, It is used to transfer the data from parent to child or sub-child or sub-child to parent that we can also do with context API.



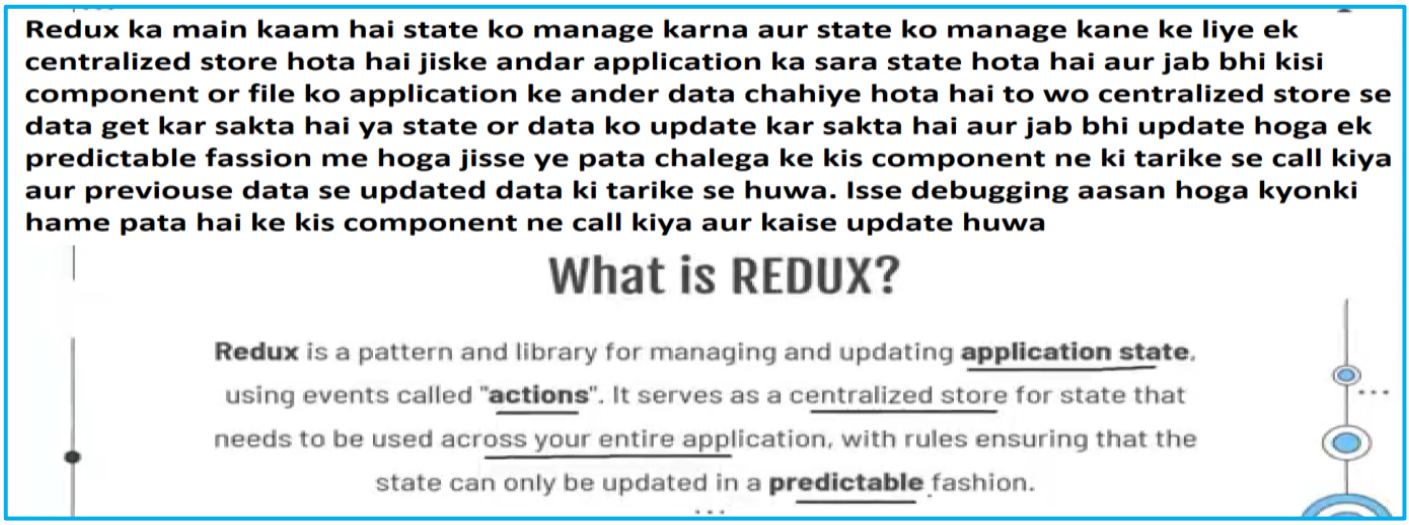
To resolve the prop drilling (which means using props again and again from a to b to c to d but It was fixed in useContext and Redux



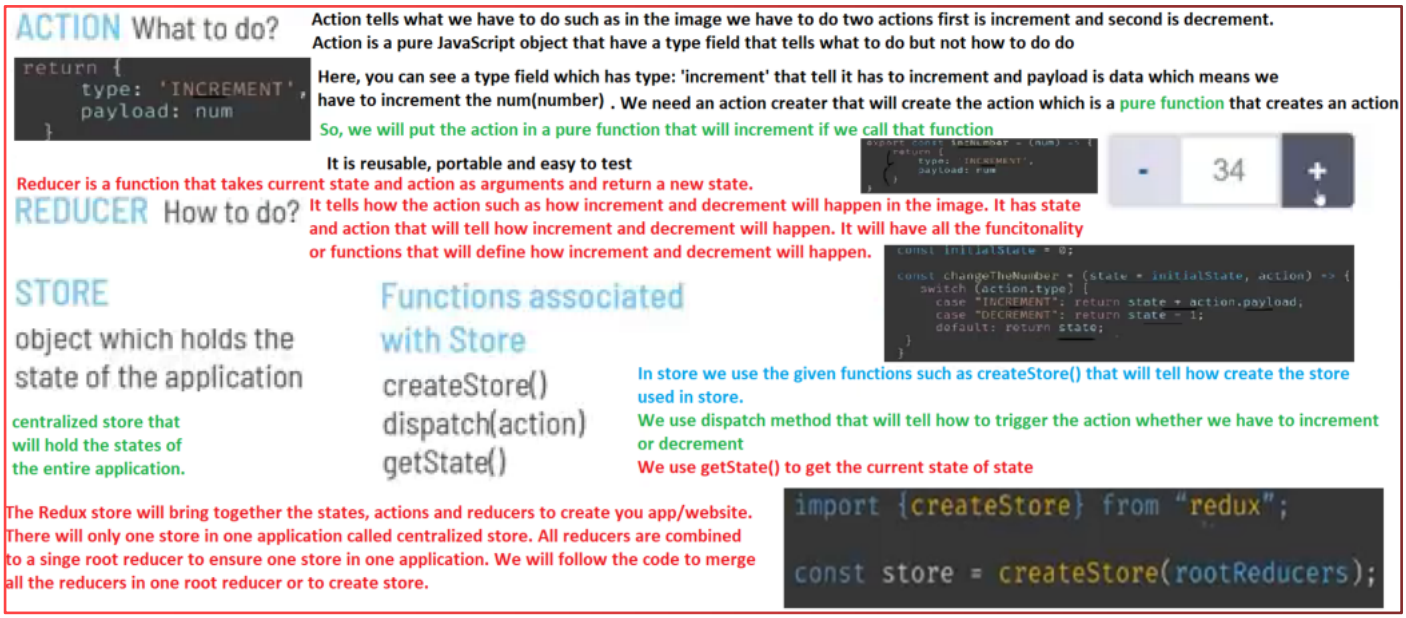


Here, jis kisi component ko data chahiye hoga store ko call karega aur data ko get kar lega
How calling the Data, Creating the store, How data is storing, How state updating
Define Redux.

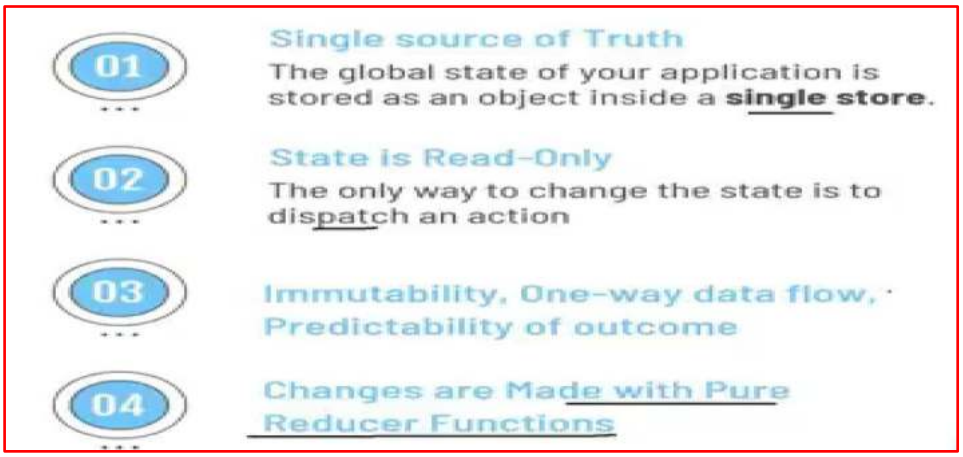
Redux ka main kaam hai state ko manage karna aur state ko manage kane ke liye ek centralized store hota hai jiske andar application ka sara state hota hai aur jab bhi kisi component or file ko application ke ander data chahiye hota hai to wo centralized store se data get kar sakta hai ya state or data ko update kar sakta hai aur jab bhi update hoga ek predictable fassion me hoga jisse ye pata chalega ke kis component ne ki tarike se call kiya aur previouse data se updated data ki tarike se huwa. Isse debugging aasan hoga kyonki hame pata hai ke kis component ne call kiya aur kaise update huwa



Define Reducer, action, store and stores method in Redux and write a code for the same.



Define Redux Principle.



Single source of truth- The global state of your application is stored in an object tree within a single store.
State is read-only- The only way to change the state is to dispatch an action.
Changes are made with pure functions

1. Redux will have only one state or store.
2. We will use dispatch(that trigger the action) to change the state .
3. Immutability will not let the data change that will not others misuse the data and code can be well organized.

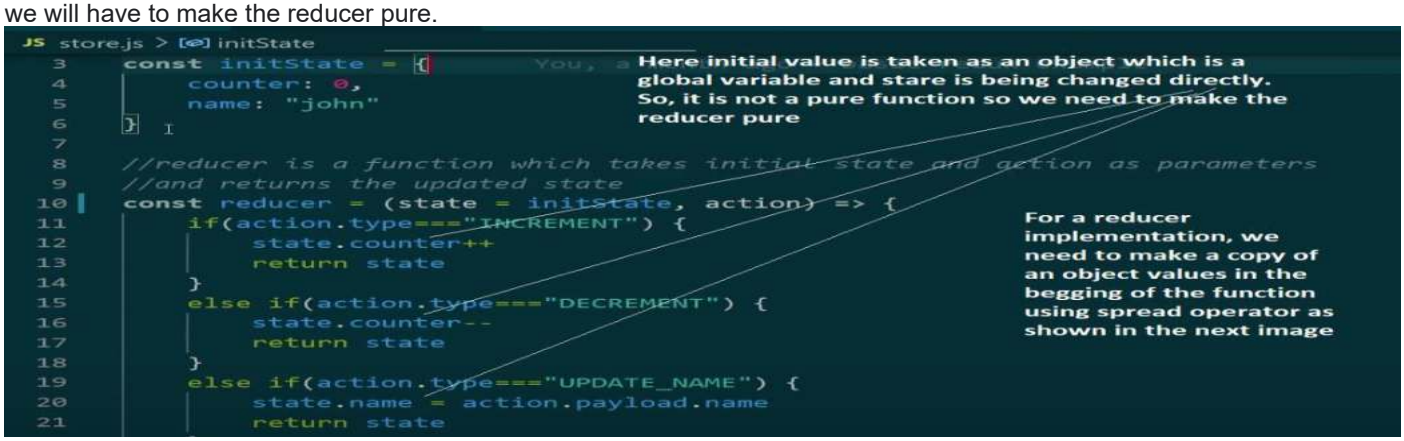
Do we have multiple reducers in Redux?

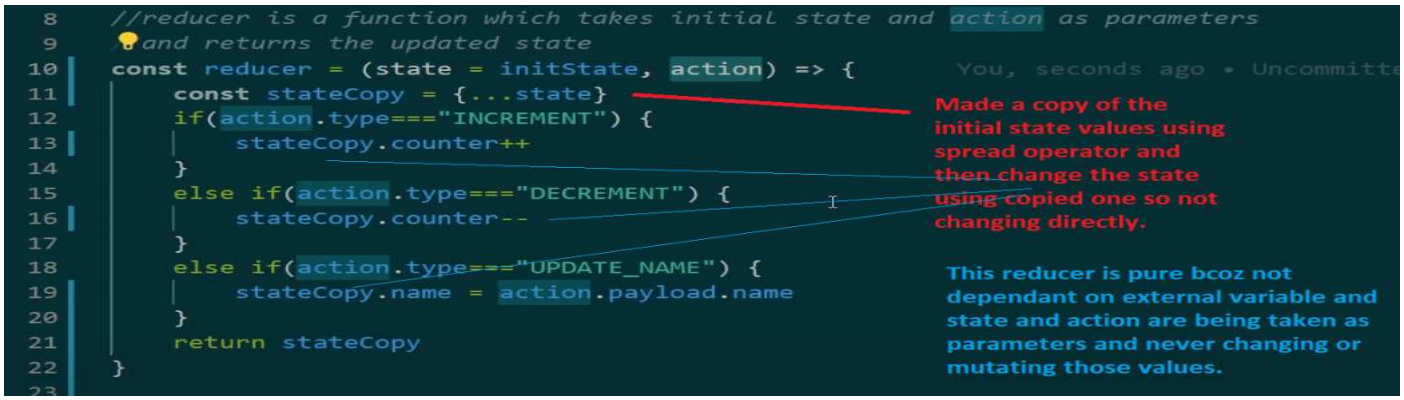
Yes, we make different reducers to perform different tasks such as increasing and decreasing number on button click, posting a form and a different reducer for authentication system. So, we have different reducers for different features of the application.
Suppose, we have two feature in our application.
First is Number Change feature on button click and Second is nameChange feature. So, we will two reduers inside reducer folder.
We will import combineReduers from Redux and then use combineReduers and then pass both reducers in an object. We need to store this in a variable that we will export and we will also import both the reducers from created file.
Follow the notes for next step for step 4

```
import {combineReducers} from 'redux'
import changeTheNumber from "./Reducers";

 const rootReducer = combineReducers({
    changeTheNumber
})
export default rootReducer;
```

Do we have multiple stores in Redux?
Yes, It is possible to create multiple stores in Redux. However, it can make the things difficult to understand. In a react-redux application, it is a good practice to have only one Redux store.

Why reducer is a pure function? First answer what pure function is and then why reducer is a pure function
Whenever we are dealing with state values in React, the state values can not be mutated or changed directly and this is why we use pure function in Reducer.
Since we use store to store redux values. So, we need to make sure our reducers are not directly updating the state values and for that we will have to make the reducer pure.





Why state is immutable in Redux and React?
Here, state is immutable means we can not change the state like any variable x = 10 or any object = {} values directly. In order to change that we need to first copy if copy object use spread operator and then change but question is why or why do we need to follow this rule or what if we break this rule or update directly means without copying as shown in the above image.

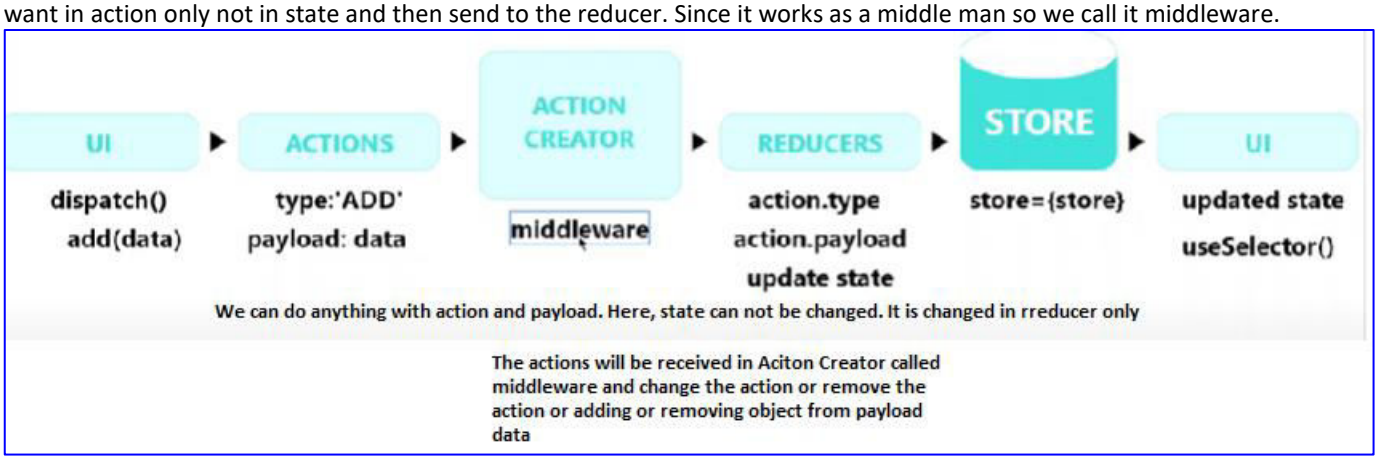
Define Redux Middleware.
Redux thunk and Redux Saga in Redux are two libraries of Redux called Middleware that provides lots of benefits to make our life easy.

Difference between Redux thunk and Saga.

Both Thunk and Saga are function.
Redux returns a function but Saga returns action within Redux.
We use **npm I redux-thunk** to install Thunk and **npm I redux-saga** to install Redux Saga.
They both are used to handle side effects. Redux thunk uses Promises to handle the side effects whereas Saga uses Generators.

When we use these two middleware, React becomes more powerfull.

The purpose middleware such as thunk and saga is to catch the action before it reaches to the reducer and make the whatever changes I want in action only not in state and then send to the reducer. Since it works as a middle man so we call it middleware.



How to connect React components with Redux.

React-Redux provides us a connect function that we can use to connect the React components to the Redux. We need to import connect from 'react-redux as shown below:-

Import { connect } from 'react-redux';

We will also need to import the component that we will connect to the Redux.

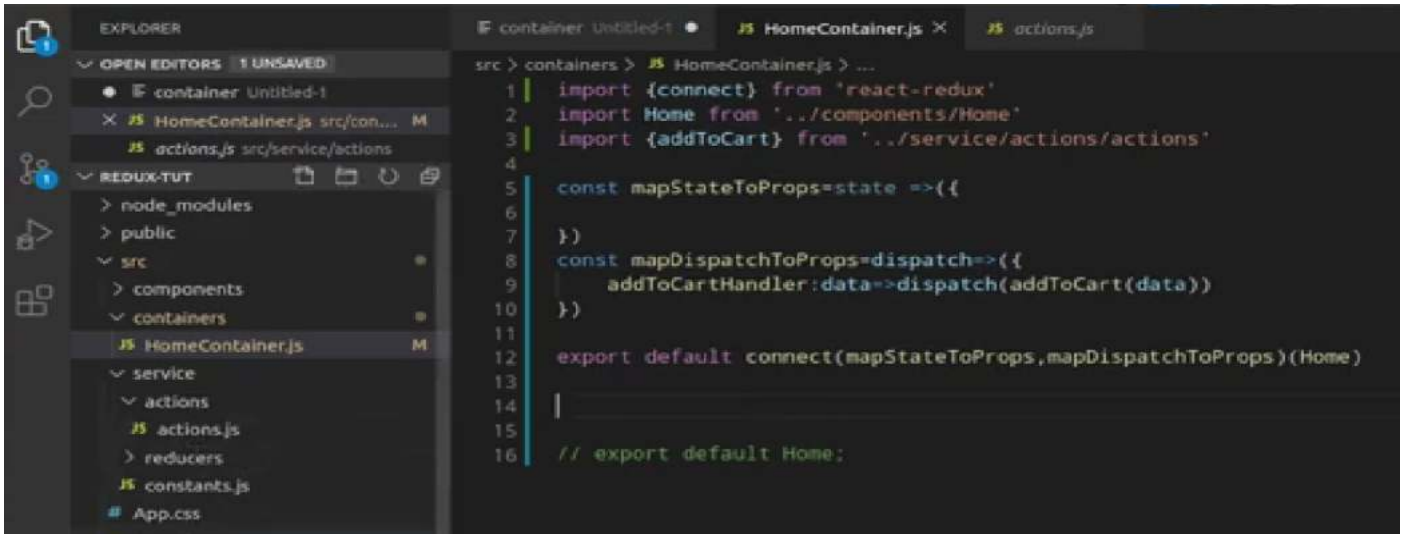
Then we will use connect function and then use parenthesis and write the component name we will connect as shown below:-

connect(mapStateToPorps, mapDispatchToProps) (component that will connect)

Connect has two parameters first is mapStateToPops and second is mapDispatchToProps. Both parameters are call back functions which are called by React-Redux later and their name can be anything. This mapStateToProps selects the state that we need from the state management tool and make state to props to send to any component where it is required. In short, it sends data from store to the React component and whenever state changes, prop's value will also change and when prop's value changes our component will re-render and show the changed value.

We use mapDispatchToProps to send the data from action(button click) to store and it has an argument dispatch that trigger the action and this action is created by action creator. You can see below in the image mapStateToProps has argument state.

Like connect(mapStateToPorps, mapDispatchToProps)



We have also useSelector and useStore hooks that we can use to take out the state values through store and selector.

Difference between mapStateToProps and mapDispatchToProps

We use mapDispatchToProps to send the data from action(button click) to store and from store to elsewhere(any component) we will use mapStateToPorps. We can give it name x or y as well.

How can we access a redux store outside a react component?

We just need to export the store from the module where it created with createStore and we also need to ensure that it doen not pollute the global window object

```
store = createStore(myReducer);

export default store;
```

How can we connect multiple middleware to Redux?
We need to import createStore and applyMiddleware from Rudux
We need to use applyMiddleware and to pass ReduxThunk and logger as an argument

```
import { createStore, applyMiddleware } from 'redux'

const createStoreWithMiddleware =
applyMiddleware(ReduxThunk, logger)(createStore);
```

Define Redux Devtools.

Redux DevTools is a live-editing time travel environment for Redux with hot reloading, action replay, and customizable UI. If you don't want to bother with installing Redux DevTools and integrating it into your project, consider using Redux DevTools Extension for Chrome and Firefox.

What are the features of Redux DevTools?

Below are the major features of Redux devTools
1. Lets you inspect every state and action payload
2. Lets you go back in time by "cancelling" actions
3. If you change the reducer code, each "staged" action will be re-evaluated
4. If the reducers throw, you will see during which action this happened, and what the error was
5. With `persistState()` store enhancer, you can persist debug sessions across page reloads

How is Relay different from Redux.
Difference between React Context and React Redux?
React Context is an inbuilt feature of React that we don't have to install from outside to use. It is good for passing the data to deeplly nested compnoents. It does not provide a large number of features.

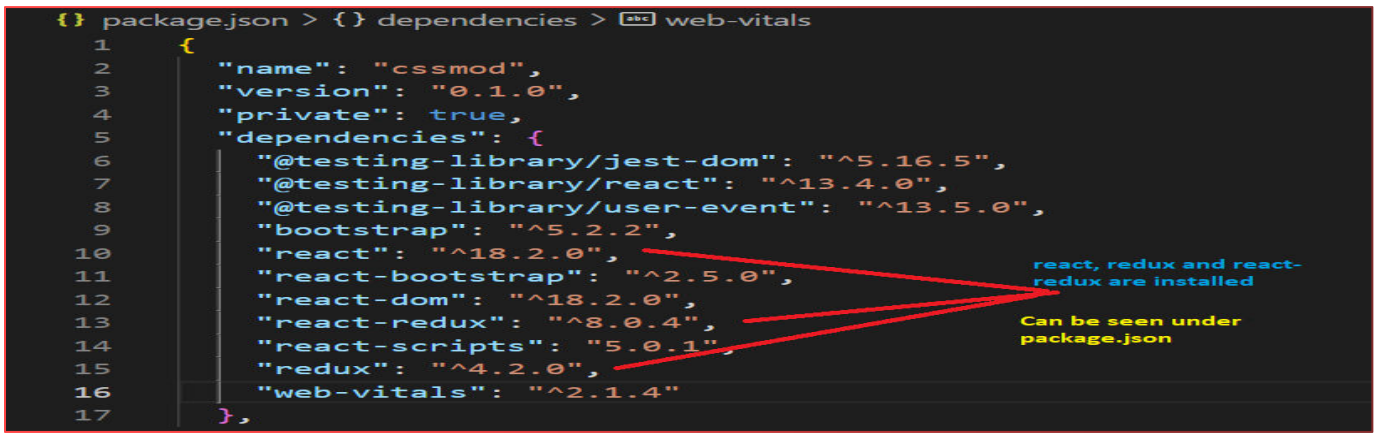
Is Redux state similar to component state?

This state is not similar to the state of any component. Here, state means how many data you currently have and how many data added later managed by this state.
Redux can be used with Angular.js and Vue.js as well. However, most of the developers use it with React.js. React uses Redux to build the user interface.

Installation of Redux

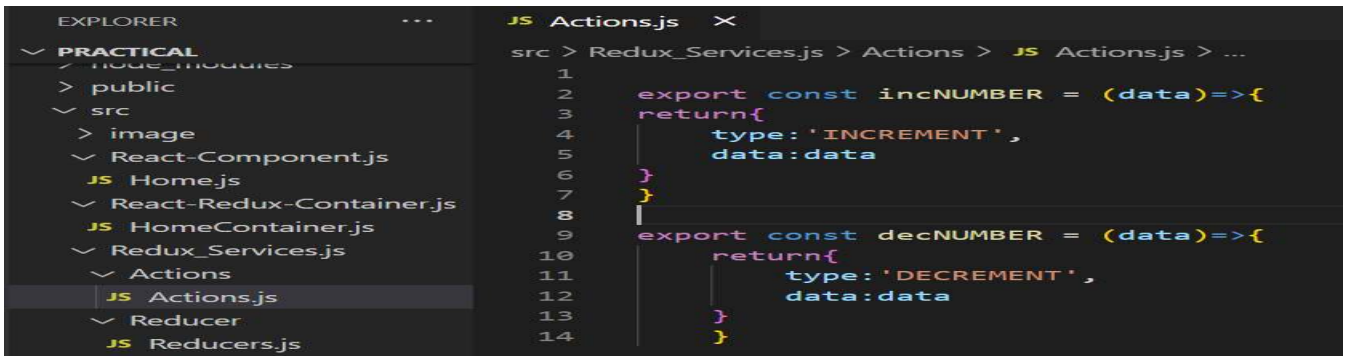
PS C:\Users\pione\OneDrive\desktop> npm install redux

PS C:\Users\pione\OneDrive\desktop> npm install react-redux (For bridging between React and Redux)

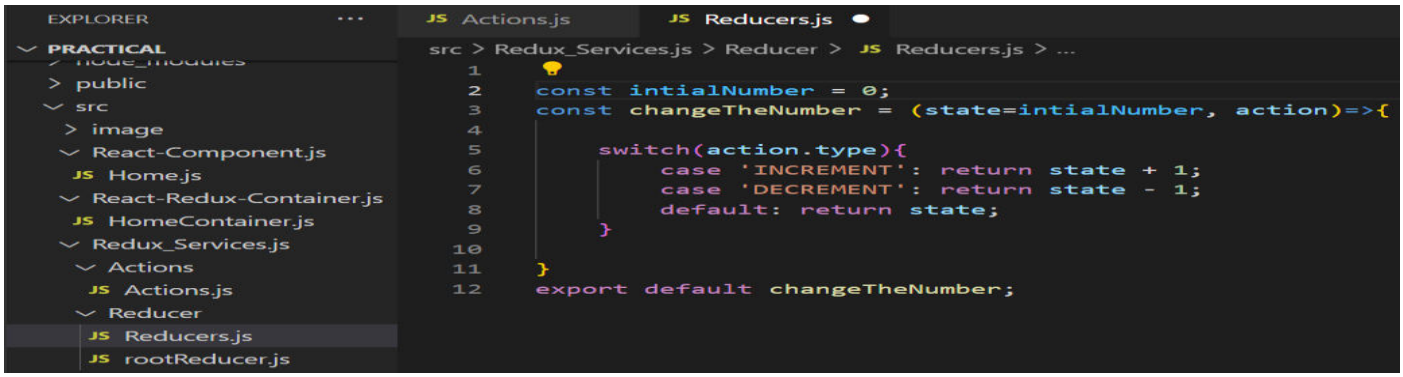


STEP TO USE REDUX.

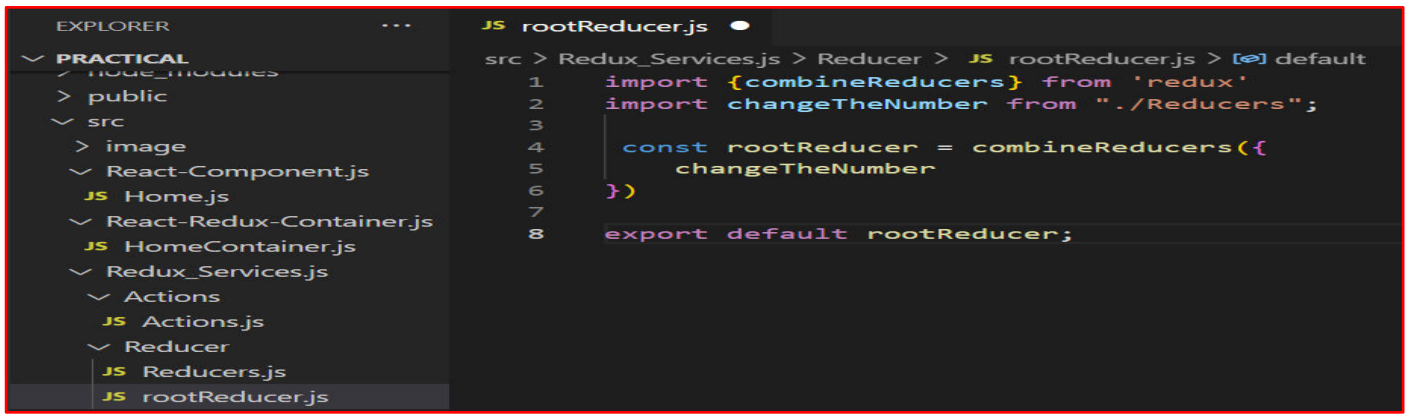
Step 1
FIRST OF ALL, WE WILL CHECK HOW MANY ANCTIONS WE HAVE TO PERFORM AND THEN CREATE UNDER ACTION AND EXPROT TO USE ELSEWHERE. IncNUMBER funciton will incremet and decNUMBER function will decrement when we call them.



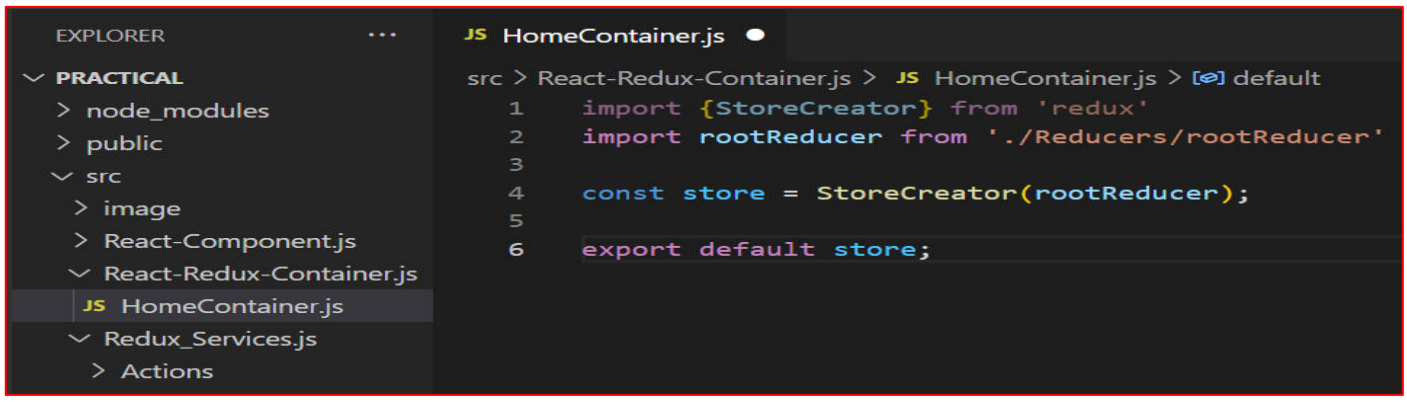
Step 2: Since we send data from action to reducer. So, we will go to the reducer file and write code for how increment and decrement will happen in reducer and use export default to use it elsewhere like in root reducer.



Step 3: We will import reducer to rootreducer. We usually use it when we have mutliple reducers. Here, we have one reducer so we can pass that one reducer to the store directly but we will not do that. We will first import in rootReducer and then pass it to the central store. We will take out or import combineReducers from redux and reducer from reducer file and store it in combineReducers({changeTheNumber}). We can take second reducer if we have using comma inside combineReducers like combineReducers({changeTheNumber, hello})and also need to import if take. The rootReducer is the complete state.

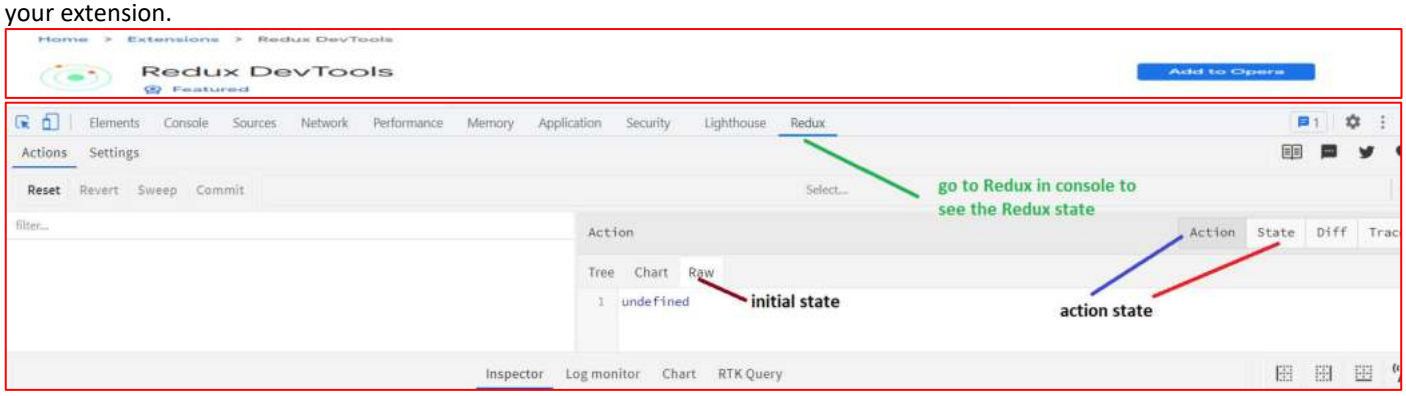


Step 4:- Since we send data from reducer to store. Here, we in order to send the reducer to the store, we need to import the rootReducer to the store.Here home-container under React-Redux-Container is my store where we will import the rootReducer.
Now, we know that we have three methods in store. First is createStore() to create the store so we will import it from Redux.
Now, we use StoreCreater to create the store and we will put the rootReducer inside the store like x = StoreCreater(rootReducer)
X is the global or centralized store/state which has all the states and whomsoever call this state will get state or data.
We also need to add this main store to the main application so that all the components can use this centralized state and for that will export it.

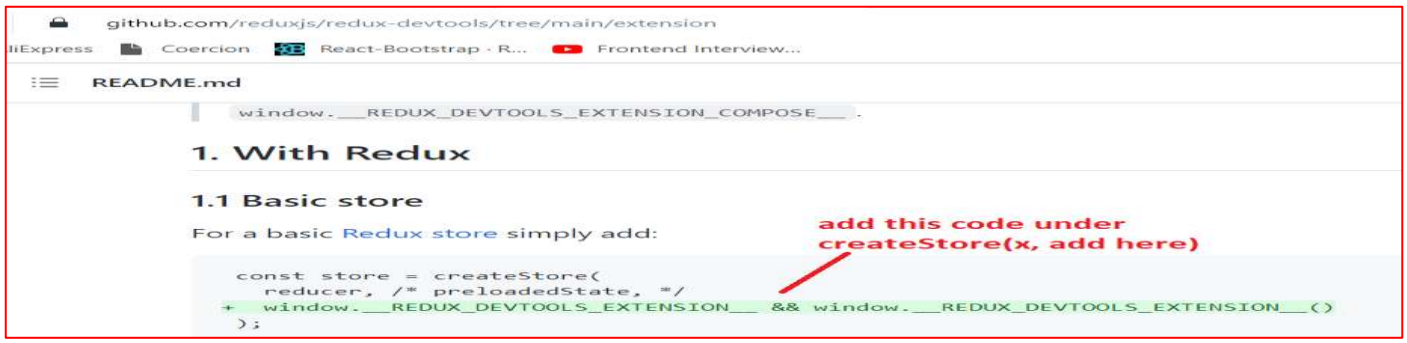


Step 5:- Now, we need to import store to the main component of the app and that is index.js

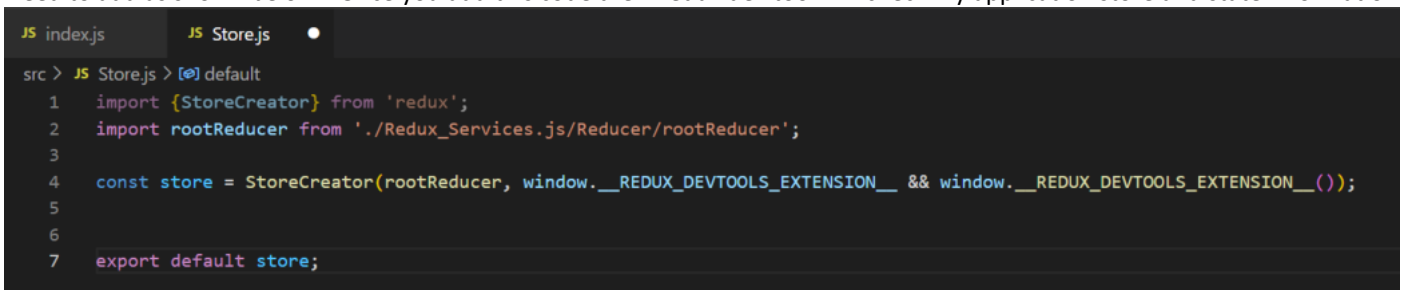
Step 6:- It is good to install redux dev tool We have component like App.js where can see the state when we create any component in React and similary we have redux dev tool when we use redux. We need to install it by searching in google Redux Dev Tools and add to your extension.



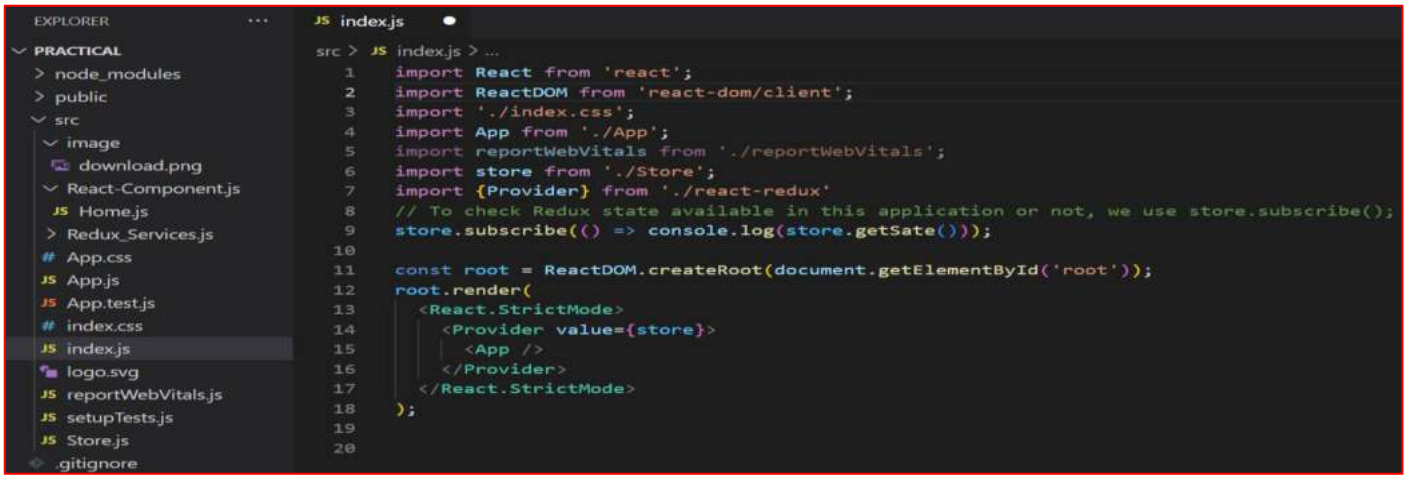
Once installed go to the git hub link just below the Redux Dev Tool in google search and add a code as shown below:-



Need to add as shown below:- Once you add this code then Redux dev tool will check my application store and state information.



Step 7:- Now, we need to import Provider in index.js so that the main App which is App.js can be provided to all the components available in the centralized store and once it is done we will connect React to Redux and for that we will wrap <App/> by provider like <Provider> <App/> </Provider> and then pass store as props in Provider. We wrapped App because it has all other components or it is the parent of all the components and App has all the data now so any component can now get data from store.



Step 8:- Now, we have to know how our individual component or child component will get the data from other component through main store(Management tool) once we put all the data in management tool. In another word, how we will receive the data. In Redux to receive the sent data we use a hook called useSelector the way we use useContext API in context API.

```
const updatedState = useSelector((argu)=>{argu.changeTheNumber});
```

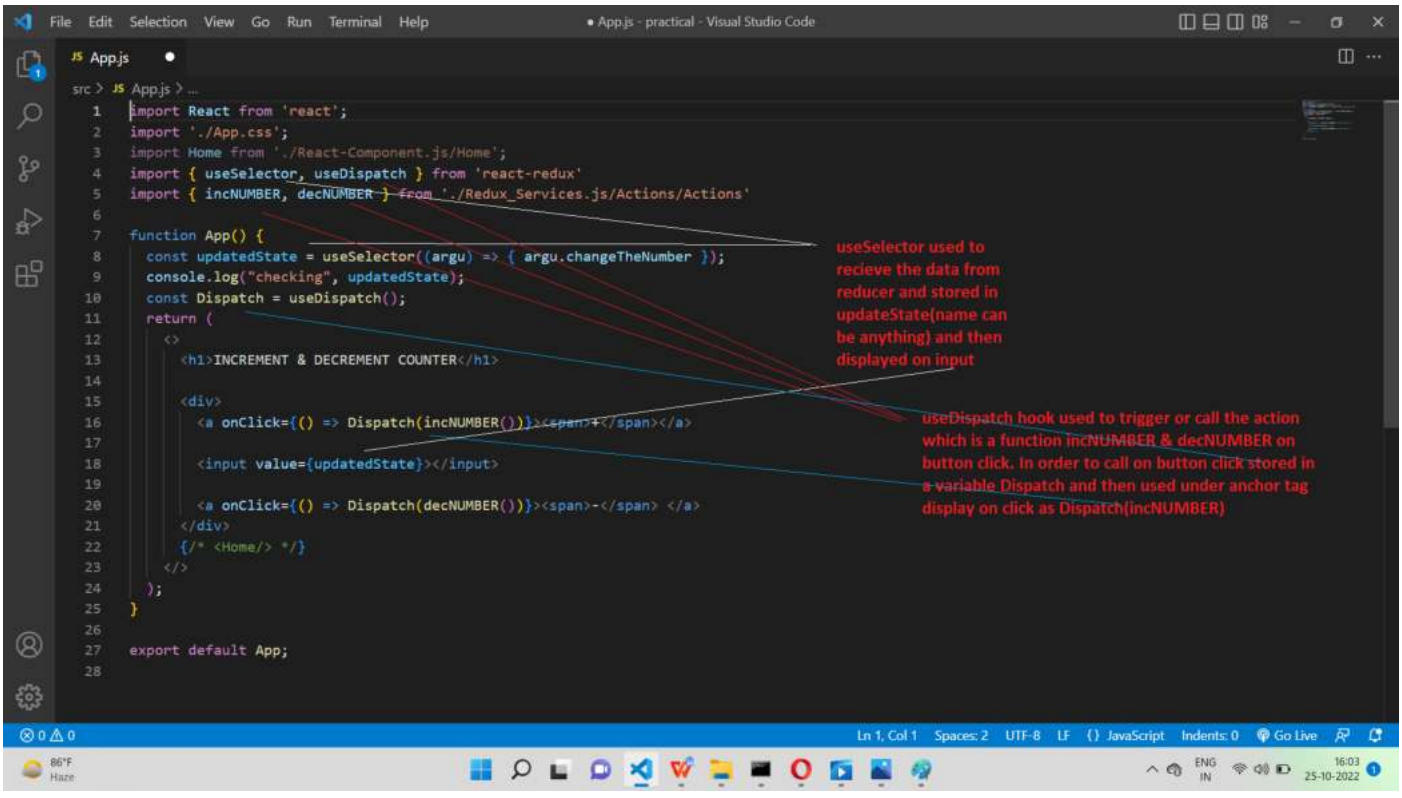
updatedState will have the sent data that we have received like here it will be 0 because our initial data is 0 that we had in Reducer, we have received initial data using useSelector hooks(This is how we get data from store or central store using useSelector) and now we will change this data on click using useDispatch hook and we also need to import both the functions incNUMBER and decNUMBER to use them from Actions.

```
import {incNUMBER, decNUMBER} from './Redux_Services.js/Actions/Actions'
```

Now, we need to call these two function so that on button click of + we can add by 1 and on - we can subtract by 1
and to call these functions we will store useDispatch value in any variable like

```
const Dispatch= useDispatch();
```

Now, Dispatch.incNUMBER to increase the number and Dispatch.decNUMBER to decrease the number on button click.



34. What is the Flux?

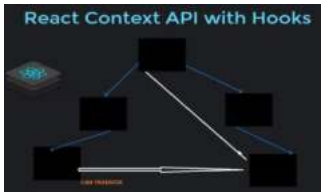
Flux is the application architecture that Facebook uses internally when working with React for building web applications. It is neither a framework nor a library but a new kind of architecture that complements React and the concept of Unidirectional Data Flow.
It is a method of handling complex data inside a client-side application and manages how data flows in a React application.
In Flux, we use dispatcher instead of Reducer and data flows from View-Action-Dispatcher-Store-View

35. How is Redux different from Flux?

Redux	Flux

Redux is an open-source JavaScript library used to manage application State	Flux is an architecture and not a framework or library
Store's state is immutable	Store's state is mutable
Can only have a single-store	Can have multiple stores
Uses the concept of reducer	Uses the concept of the dispatcher
View-Action-Reducer-Store-View	View-Action-Dispatcher-Store-View
	MVC was replaced by Flux for unidirectional data flow

36. Define context API. (v)



When we have to send the data from parent to child component, we can easily do it using props but if we have to send the data from parent to sub-child or grand child or grand child to child we use Redux in a big project but if we have small project we can do the same thing using context API. It is not necessary to send parent to child first and then from child to sub child if we have to parent transfer the data from parent to sub child or grand child. We can directly transfer data from to grand child or grand child to parent using Redux and context API.

We can do the same thing using Redux and context API. It is better to use context API because context API is internal or inbuilt hooks that we don't need to install as already installed but Redux is an external library that we need to install separately in Reat.js project to use.

When we use Redux, it will apply to all the components used in that project . Hence all the components will be under the Redux also called management tool but in context API, it is not like that, you can take any component you want to apply context API. It is not necessary to use all the components in context API.

It is not used by most of the developer because they think it is not so capable that Redux but it is not like that we can make small project using this.

It is better to use context API in a functional component because we don't have to write multiple codes to use it.

Step to use:- Step 1 - First we will make a child component.

Step 2 - Send data from parent(app component) to child(child component)

Step 3- write createContext under curly braces in main component(aap.js) to import

```
import React, {createContext} from "react";
```

and just above the component [Here App()] take a variable like const global= createContext();

Now make wrapper wherever you want to use, we will make a wrapper in the main component in which all components will be linked so this way we can transfer data from any to any component.

```
//STEP 4 taken state to store data  //step 5 taken value to set green color in all the components used under the wrapper
```

Using step 4 and 5 we sent the data {appColor:color} and now we need to access the data.

The child component in which you want to receive the sent data we will use hook {useContext}

Now sent data is in useContext so we will store it in a variable

```
const appColor = useContext(GlobalInfo);
```

Now data is in appColor so used in h2 to print

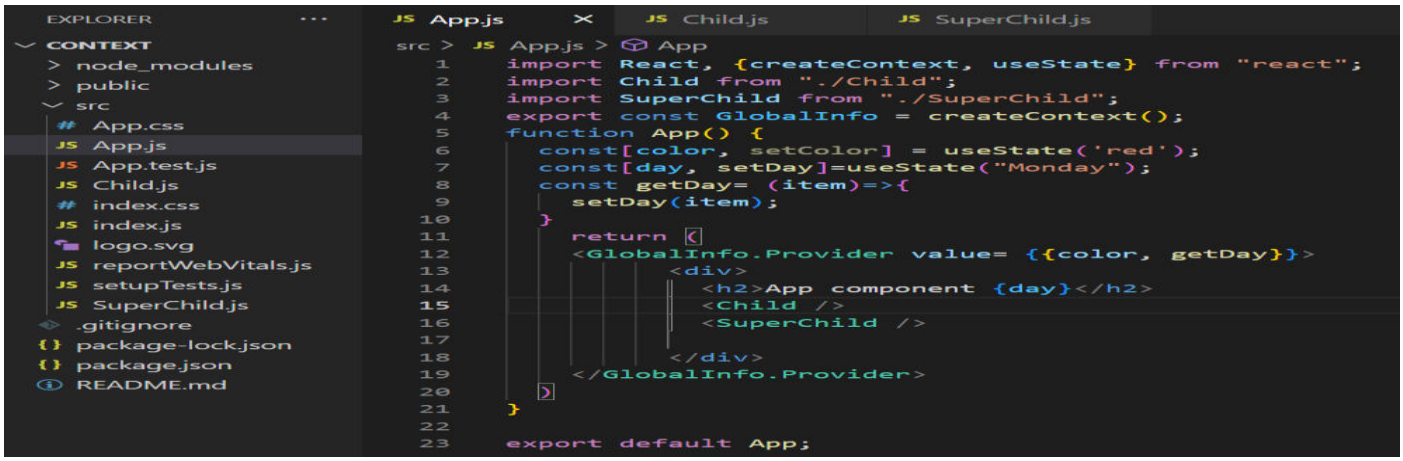
```
<h2 style={{color:appColor}}>Child Component</h2>
```

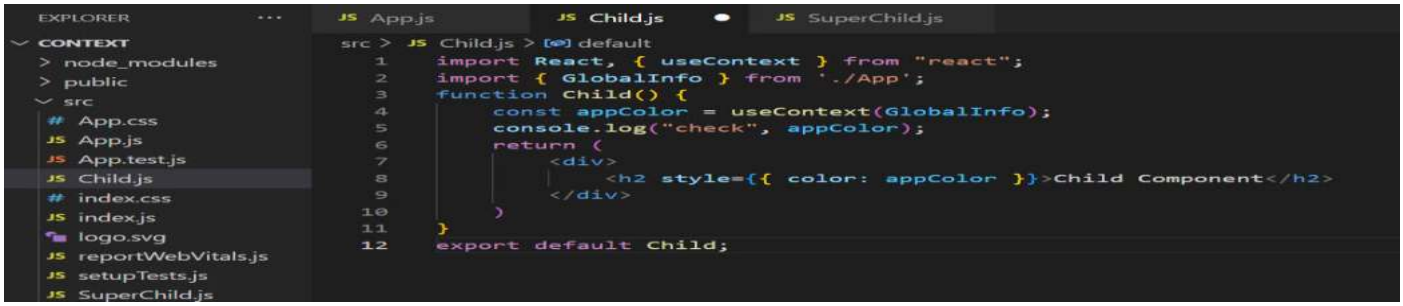
Also, we need

```
// step 1 to receive the sent data step 2 receive alppColor
```

We also need to export global and import in child

```
export const Global = createContext();   // from main component


import {Global} from "./App";      // in child component
```





Super child to parent

We will send Sunday in line No. 5 from parent component from super child

Define PWA.

It stands for Progressive Web Applications and they are websites which look and feel like mobile Apps or other native mobile apps.

Features or Benefits

It offers features like it can be used offline.
It is small in size than native apps.
It does not require to install and update as it always loads the latest updated version as soon as user interaction happens so no play store required.
we can simply make a shortcut of the website and use it.
it is very fast and responsive that can fit in every screen size automatically like in smarphone, tablet, desktop or laptop.
It is not expensive as mobile native app nee to be developed for both Android and iOS but PWA has the same feature but not that expensive.
It gives good user experience so rank in SEO and its links are also shareable. It is very secured becoz delivered over HTTPS connection
It provides push notification to interact easily with the users and provide really amazing user experience.

How to make Progressive Web Apps

In order to make progressive web app, we need service worker.

Service worker is a script that runs in the background of the browsers. Here, background means if you are doing something in the front end like submitting form or something then background script will not make any difference and work in the background. We have some functions and promises in JavaScript that we use to make script for the service worker.We need http to run the service worker. So, we will use live server to open the service worker example.
Service worker is located at navigation.serviceWorker

EXTRA QUESTIONS

Q2: How to access DOM elements in React?

```
);}
```

Q3: How to call loading function with React `useEffect` only once?

If you only want to run the function given to `useEffect` after the *initial render*, you can give it an *empty array* `[]` as the second argument.

For example:

```
function MyComponent(){

    useEffect(() => {

        loadDataOnlyOnce();

    }, []);



    return <div> { /*...*/} </div>;}
```

Q4: Provide an example of any simple *Custom* React Hook. Why do we need Custom Hooks?

A **Custom** Hook is a **stateful function** that uses other react built-in hooks (e.g. `useState`, `useCallback` etc.) that can wrap around the *stateful* logic that you wanted to gather in one place and *avoid copying and pasting the same logic* in multiple components.

Consider the increment/decriment custom hook:

```
const useCounter = () => {

  const [counter, setCounter] = useState(0);
```

```
  return {
    counter, // counter value
    increment: () => setCounter(counter + 1), // function 1
    decrement: () => setCounter(counter - 1) // function 2
  };};
```

and then in your component you can use it as follows:

```
const Component = () => {
  const { counter, increment, decrement } = useCounter();

  return (
    <div>
      <span onClick={decrement}>-</span>
      <span style={{ padding: "10px" }}>{counter}</span>
      <span onClick={increment}>+</span>
    </div>
  );}
```

Q7: Do two components using the same Hook share state?

No. Custom Hooks are a mechanism to reuse *stateful logic* (such as setting up a subscription and remembering the current value), but every time you use a custom Hook, all state and effects inside of it are *fully isolated*.

Q8: Explain the difference between `useState` and `useRef` hooks?

1. Updating a reference created by `useRef` doesn't trigger re-rendering, while updating the state (`setState`) makes the component re-render;
2. `useRef` returns an *object with a current property holding the actual value*. In contrast, `useState` returns an *array with two elements*.
3. `useRef's` current property is *mutable*, but `useState's` state variable is *not*.
4. The reference update is *synchronous* (the updated reference value is available right away), while the state update is *asynchronous* (the state variable is updated after re-rendering).

Using `useRef` - no re-renders

```
const countRef = useRef(0);
  const handle = () => {
    countRef.current++;
    console.log(`Clicked ${countRef.current} times`);};
```

Using `useState` - triggers re-render

```
const [count, setCount] = useState(0);
  const handle = () => {
    const updatedCount = count + 1;
    console.log(`Clicked ${updatedCount} times`);
    setCount(updatedCount);};
```

Q9: Define Error boundaries.

Error baounaries help to affect only that component on which we have errors rather than crashing the entire application or other components of the application. Only class components can be error boundaries.

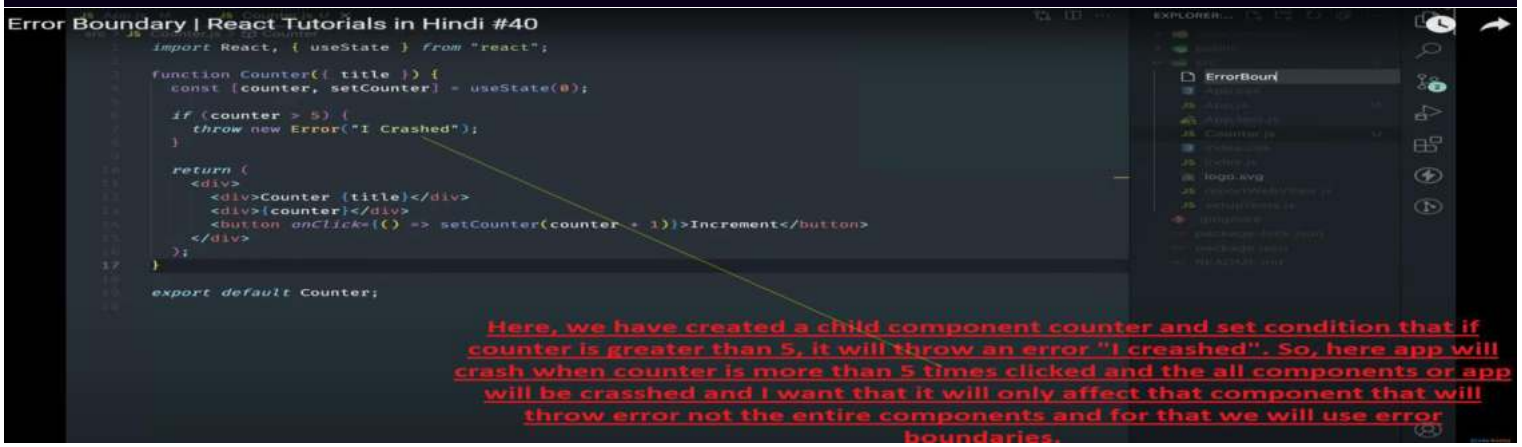
Syntax

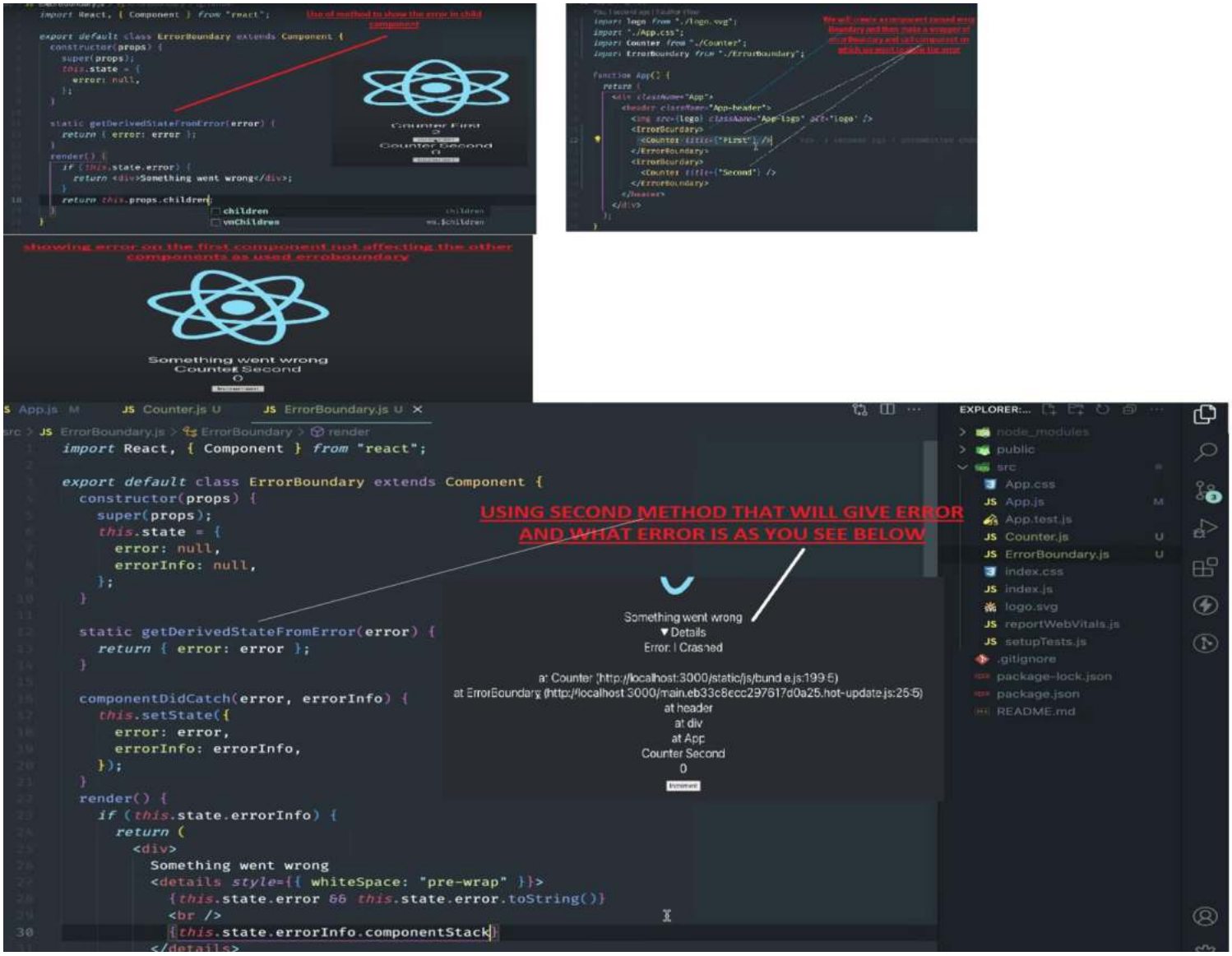
```
static getDerivedStateFromError(error) {
  // Update state so the next render will show the fallback UI.
  return { hasError: true };
}
```

We can use getDerivedStateFromError() to get/throw the error and we can change the state and update with the error message we want.

```
componentDidCatch(error, errorInfo) {
  // You can also log the error to an error reporting service
  logErrorToMyService(error, errorInfo);
}
```

We can use componetDidCatch() to throw the error and also what the error is. We can also save the error message in any log.





How can I make use of *Error Boundaries* in functional React components?

As of v16.2.0, there's no way to turn a functional component into an error boundary. The `componentDidCatch()` method works like a JavaScript `catch {}` block, but for components. **Only class components can be error boundaries. In practice, most of the time you'll want to declare an error boundary component once and use it throughout your application.**

Also bear in mind that `try/catch` blocks *won't work on all cases*. If a component deep in the hierarchy tries to update and fails, the `try/catch` block in one of the parents won't work -- because it isn't necessarily updating together with the child.

A few third party packages on npm implement error boundary hooks.

Q10: How to use `componentWillMount()` in React Hooks?

You cannot use any of the existing lifecycle methods (`componentDidMount`, `componentDidUpdate`, `componentWillUnmount` etc.) in a hook. They can only be used in class components. And with Hooks you can only use in functional components.

You can think of `useEffect` Hook as `componentDidMount`, `componentDidUpdate`, and `componentWillUnmount` combined.

Code inside `componentDidMount` run once when the component is mounted. `useEffect` hook equivalent for this behaviour is

```
useEffect(() => {

 // Your code here}, []);
```

Without the second parameter the `useEffect` hook will be called on every render (like `componentDidUpdate`) of the component which can be dangerous:

```
useEffect(() => {

 // Your code here});
```

Hook equivalent of `componentWillUnmount()` code will be as follows

```
useEffect(() => {

 window.addEventListener('mousemove', () => {});



 // returned function will be called on component unmount

 return () => {

   window.removeEventListener('mousemove', () => {})

 }}, [])
```

Q11: What are common use cases for the `useMemo`?

The primary purpose of `useMemo` hook is *"performance optimization"*.

- It returns a *memoized value*,
- It accepts *two arguments* - **create** *function* (which should return a value to be memoized) and **dependency** *array*. It will *recompute the memoized value only when one of the dependencies has changed.*

Using `useMemo` you achieve:

- referential equality of the values (to further send them to props of the components to potentially avoid re-renders)
- eliminate redoing of the computationally expensive operations for same parameters

For example:

```
function App() {

    const [data, setData] = useState([....]);



    function format() {

        console.log('formatting....'); // this will print only when data has changed

        const formattedData = [];

        data.forEach(item => {

            const newItem = //...do soemthing here,

            if (newItem) {

                formattedData.push(newItem);

            }

        })

        return formattedData;

    }



    const formattedData = useMemo(format, [data])



    return (

        <>

        {formattedData.map(item => (

            <div key={item.id}>
```

```
            {item.title}
            </div>
        ))}
        </>
    )  }
```

Q12: What are differences between `React.memo()` and `useMemo()`?

- `React.memo()` is a higher-order component (HOC) that we can use to wrap **components** that we do not want to re-render unless props within them change
- `useMemo()` is a React Hook that we can use to wrap **functions** within a component. We can use this to ensure that the values within that function are re-computed only when one of its dependencies change

Q13: What are production use cases for the `useRef`?

Answer

- `useRef` simply returns a *plain Javascript object*. Its value can be *accessed* and *modified* (mutability) as many times as you need without worrying about *rerender*.
- `useRef` value will *persist* (won't be reset to the *initialValue* unlike an ordinary object defined in your function component; it persists because `useRef` gives you the same object *instead of creating a new one* on subsequent renders) for the component lifetime and across re-renders.
- `useRef` hook is often used to store values instead of DOM references. These values can either be a state that does not need to change too often or a state that should change as frequently as possible but should *not trigger* full re-rendering of the component.

For example:

```
const refObject = useRef(initialValue);
```

Q16: When would you use `useRef`?

The main use cases:

To store a `ref` to **DOM** elements so you can later do something with them:

```
function TextInputWithFocusButton() {
   const inputEl = useRef(null);
   const onButtonClick = () => {
       inputEl.current.focus();
   };
   return (
       <>
           <input ref={inputEl} type="text"/>
           <button onClick={onButtonClick}>Focus the input</button>
       </>
   );}
```

To store values *without* triggering *re-renders:*

```
function Counter(){
   const [count, setCount] = useState(0);
```

```
  const prevCountRef = useRef();

  useEffect(() => {

      prevCountRef.current = count;

  });

  const prevCount = prevCountRef.current;



  return <h1>Now: {count}, before: {prevCount} </h1>;}
```

Q17: When writing a Custom Hook, what is the difference between it and a normal function?

Hooks use a *stateful* closure around the invocation of the function component to store the state on behalf of that component instance. That closure is maintained by React.

- Custom hook will only be "stateful" if you use state with `useState` inside (or something that implements `useState`),
- Hooks should be called from the React code only not from the regular JS functions. Hence, Hooks' scope is limited to the React code world and has more power to do a lot with React code,
- In the class-based components, the Hooks won't work but regular functions will.
- In the regular JS functions, you can't access `useState`, `useEffect`, `useContext` etc. but in react custom hooks I can.

Q18: Which *lifecycle* methods of class component is replaced by `useEffect` in functional

The lifecyce methods replaced by `useEffect` Hooks of functional component are `componentDidMount()`, `componentDidUpdate()`, *and* `componentWillUnmount()`

`componentDidMount`: is equivalent for running an *effect once*.

For example:

```
useEffect(() => {

  console.log("This is useEffect Hook equivalent of componentDidMount lifecycle method")},[]);
```

Note: empty array = `useEffect` hook *runs once on mount*

`componentDidUpdate`: is equivalent for *running effects when things change*

For example:

```
useEffect(() => {

  console.log("The name props has changed!");}, [props.name]);
```

`componentWillUnmount`: To run a *hook as the component is about to unmount*, we just have to return a function from the `useEffect` Hook

For example:

```
useEffect(() => {

  console.log('running effect');

  return () => {

      console.log('unmount');

  }})
```

Q1: Do you need to keep all component states in Redux store?

You need to keep your application state as small as possible. You don't have to push everything in there. Only do it makes a lot of sense to keep something there Or if it makes your life easier when using Dev Tools. But we shouldn't overload its importance too much.

Q

Q6: How to set initial state in Redux?

You need to pass initial state as second argument to `createStore`

```
const rootReducer = combineReducers({
  todos: todos,
  visibilityFilter: visibilityFilter});
const initialState = {
  todos: [{id:123, name:'sudheer', completed: false}]};
const store = createStore(
  rootReducer,
  initialState);
```

Q7: How to structure Redux top level directories?

Most of the applications has several top-level directories as below

1. **Components** Used for “dumb” React components unaware of Redux
2. **Containers** Used for “smart” React components connected to Redux
3. **Actions** Used for all action creators, where file name corresponds to part of the app
4. **Reducers** Used for all reducers, where file name corresponds to state key
5. **Store** Used for store initialization This structure works well for small and mid-level size apps.

Q8: What are Redux selectors and Why to use them?

Selectors are functions that take Redux state as an argument and return some data to pass to the component. For example, to get user details from the state:

```
const getUserData = state => state.user.data;
```

Q9: What are reducers in redux?

The **reducer** is a *pure function* that takes the previous state and an action, and returns the next state.

```
(previousState, action) => newState
```

It's called a reducer because it's the type of function you would pass to `Array.prototype.reduce(reducer, ?initialValue)`. It's very important that the reducer stays *pure*. Things you should never do inside a reducer:

- Mutate its arguments;
- Perform side effects like API calls and routing transitions;
- Call non-pure functions, e.g. `Date.now()` or `Math.random()`.

Q12: What is Redux Thunk?

Redux Thunk middleware allows you to write action creators that return a function instead of an action. The thunk can be used to delay the dispatch of an action, or to dispatch only if a certain condition is met. The inner function receives the store methods dispatch and `getState()` as parameters.

Q13: What is redux-saga?

redux-saga is a library that aims to make side effects (i.e. asynchronous things like data fetching and impure things like accessing the browser cache) in React/Redux applications easier and better. It is available in NPM as

```
npm install --save redux-saga
```

Q15: Are there any similarities between Redux and RxJS?

These libraries are very different for very different purposes, but there are some vague similarities.

Redux is a tool for managing state throughout the application. It is usually used as an architecture for UIs. Think of it as an alternative to (half of) Angular.

RxJS is a reactive programming library. It is usually used as a tool to accomplish asynchronous tasks in JavaScript. Think of it as an alternative to Promises.

Redux uses the Reactive paradigm little bit because the Store is reactive. The Store observes actions from a distance, and changes itself. RxJS also uses the Reactive paradigm, but instead of being an architecture, it gives you basic building blocks, Observables, to accomplish this "observing from a distance" pattern.

Q20:What are the main features of Redux Form?

Below are the major features of redux form: 1. Field values persistence via Redux store 2. Validation (sync/async) and submission 3. Formatting, parsing and normalization of field values

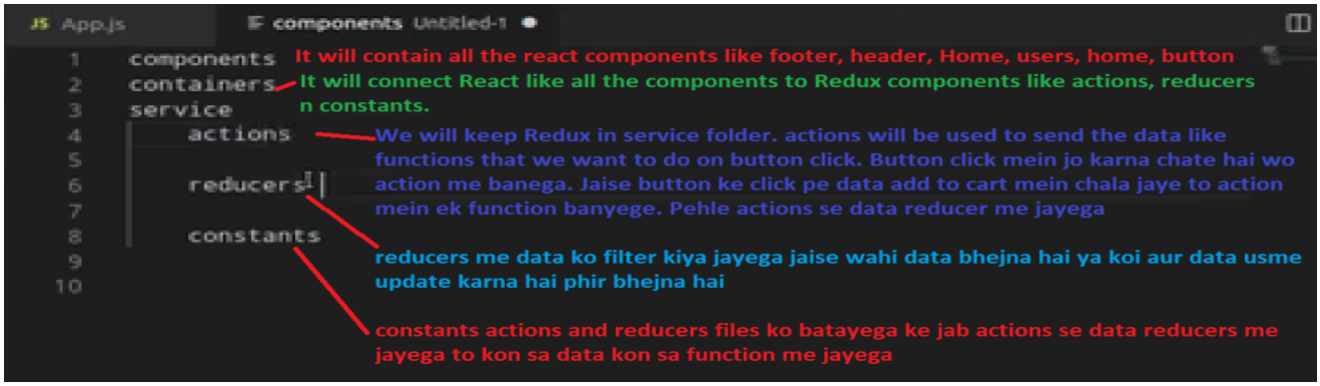
Q21: What is Redux form?

The best way to manage your form state in Redux.

Redux Form works with React and Redux to enable a form in React to use Redux to store all of its state. Redux Form can be used with raw HTML5 inputs, but it also works with very well with common UI frameworks like Material UI, React Widgets and React Bootstrap.

Q22: What is the purpose of the constants in Redux?

Constants in Redux tells actions and reducers files that what data will go to what funcitons when we send data from actions to reducers.



Constants allow you to easily find all usages of that specific functionality across the project when you use an IDE. It also prevents you from introducing silly bugs caused by typos -- in which case, you will get a ReferenceError immediately.

Q23. Why React uses `className` over `class` attribute?

`class` is a keyword in JavaScript and JSX is JavaScript XML extension to the JavaScript. That's the principal reason why React uses `className` instead of class.

PRACTICAL

```
import React, { useState } from 'react'
function State(){
    const[data, setData]=useState("sarfraz");
    var x =100;
    function hello(){
        var x = 120;
        setData("Alam");
    }
    return (
        <>
            <h1> I am state data : {data} and variable value {x}</h1>
            {/* <button onClick={hello}>Click</button> */}
            <button onClick={()=>{hello()}}>Click</button>
        </>
    )
}export default State;
```

How can you embed two or more components into one?

Note:- It is not necessary to make 2nd component to embed the another component and how to implement HOD

Remember jab kisi component ko as props receive karege to <prps.name/> name is a component lege

```
import React from "react";

function State(){
    return(
        <>
            <h1> I am main conponent</h1>
            <HODred name= {Counter}/>
            <HODblack name= {Counter}/>
        </>
    )
}
function HODred(props){
    return(
        <h3 style={{backgroundColor:'red', width: 200, height:200}}><props.name/></h3>
    )
}
function HODblack(props){
    return(
        <h3 style={{backgroundColor:'blue', width: 200, height:200}}><props.name/></h3>
    )
}
function Counter(){
    return(
```

```
        <>
            <h1> I am inside conponent</h1>
        </>
    )
}
export default State;
```

State and Props in class component

```
import React, { Component } from 'react'
class State extends Component{

    constructor(){
        super();
        this.state= {name:"sarfraz", age: 25 };
    }
    // hello() {
    //     this.setState({name:"Alam", age:30})
    // }
    render(){
        return(
            <>
                <h1>{`My name is ${this.state.name} and My age is ${this.state.age}`}</h1>
                {/* <button onClick={()=>this.hello()}>Click On Me</button> */}
                <button onClick={()=>this.setState({name:"Alam", age:30})}>Click On Me</button>
            </>
        )
    }
}
export default State;
```

```
                            PROPS IN CLASS COMPONENT(IT IS PARENT COMPONENT)


import State from './State';
import React from 'react';
class App extends React.Component {
  render() {
    return (
      <>
        <State name="Sarfraz" age="22" address={{ city: "Dhanbad", State: "Jharkhand" }} />
      </>
    );
  }
}
export default App;
```

```
                                PROPS IN CLASS COMPONENT(IT IS CHILD COMPONENT)


import React from 'react'

class State extends React.Component {
    render() {
        return (
            <>
                <h1>{` My Name is ${this.props.name} and my age is ${this.props.age}
                and my city is ${this.props.address.city}`}</h1>
            </>
        )
    }
}
export default State;     output     My Name is Sarfraz and my age is 22 and my city is Dhanbad
```

Props in functional component

Parent component

```
import State from './State';

function App() {
  return (
    <>
      <State name= "Sarfraz" age= "22" address= {{city:"Dhanbad", State:"Jharkhand"}}/>

    </>
  );
}
export default App;
```

Child component

```
import React from 'react'

function State(p){
    return(
        <>
        <h1>Hello</h1>
        <h1>{` My Name is ${p.name} and my age is ${p.age} and my city is ${p.address.city}`}</h1>
        </>
    )
}
export default State;
```

Ref in class component

```
import React,{ createRef } from "react";
class State extends React.Component{
        constructor(){
            super()
            this.Sarfraz = createRef();
        }
        hello(){
            this.Sarfraz.current.style.backgroundColor=  "red"
            this.Sarfraz.current.style.color=  "blue"
        }
        render(){
            return(
                <>
                <h1>Example of Ref in Class Component</h1>
                <input name='text' ref={this.Sarfraz} />
                <button onClick={()=>this.hello()}>Click On Me</button>
                </>
            )
        }
}
export default State;
```

useRef in function component

```
import React, { useRef } from 'react';
function State(){
               const inputRef =useRef();
               console.log("All I got", inputRef);
               function sarfraz(){
                // inputRef.current.value= "100"
                inputRef.current.style.color= "red";
                inputRef.current.style.textAlign= "center";

               }
        return(
            <>
            <h1>useRef in functional Component</h1>
            <input type= "text" ref={inputRef}/>
            <button onClick={()=>sarfraz()}>On Click</button>
            </>
```

```
        )
}
export default State;
```

Useref and forward ref

Parent component

```
import State from './State';
import React, { useRef } from 'react';

function App(){
  const inputRef = useRef();
  function hello(){
    // inputRef.current.value="100";
    inputRef.current.style.backgroundColor="red";
  }
  return(
    <>
    <State ref ={inputRef}/>
    <button onClick={()=>hello()}>Click On Me</button>
    </>
  )
}
export default App;
```

Child component to use forwardref

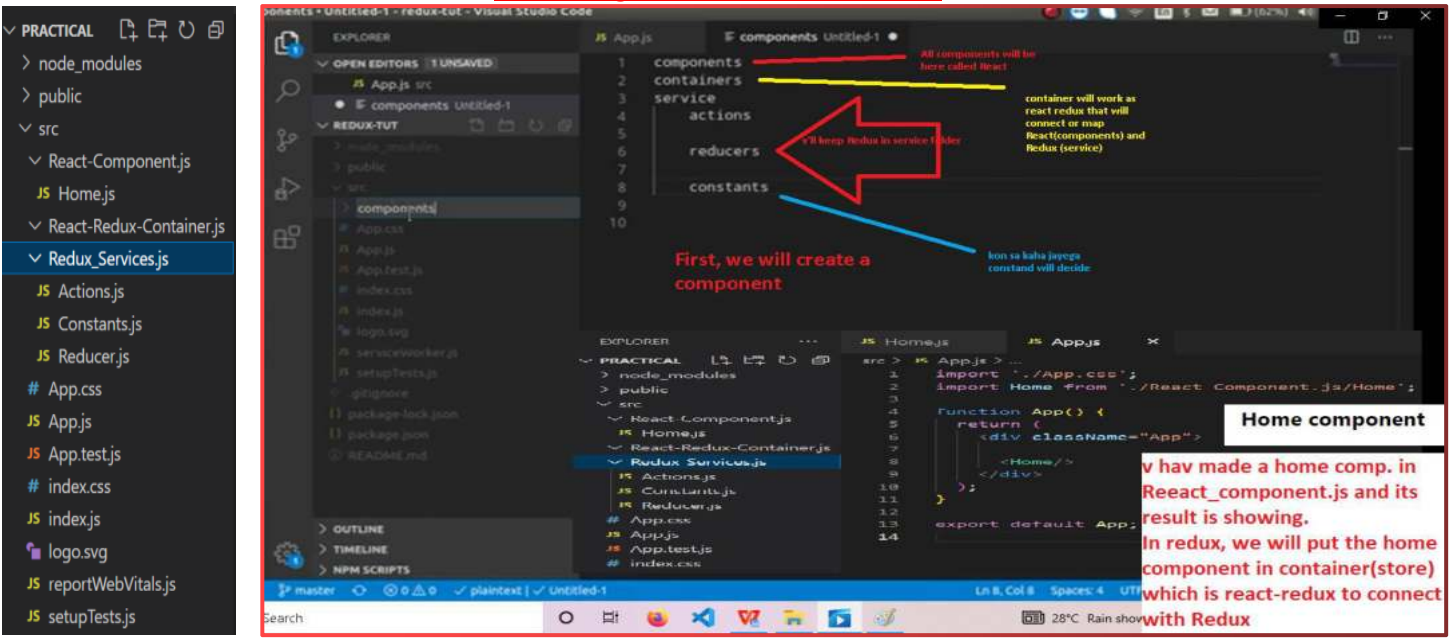
```
import React, { forwardRef } from "react";

function State(props, ref){
    const abc =forwardRef()
    return(
      <>
      <input  ref={ref}/ >
      </>
    )
  }
  export default forwardRef(State);
```

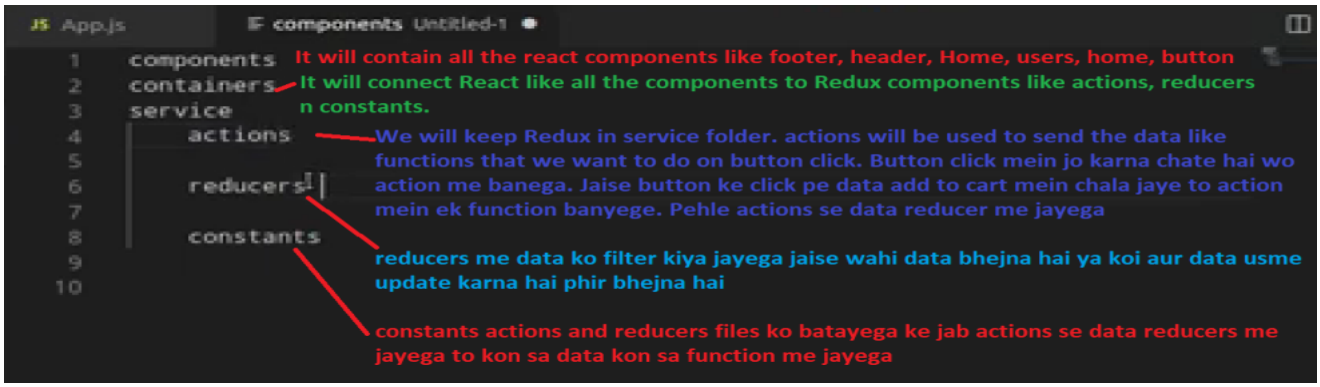
Pure component

```
import React, { Component } from 'react'
class State extends Component{

    constructor()
    {
        super();
        this.state = {
            count:10
        }
    }
    // tipu(){
    //     this.setState({count: this.state.count + 1})
    // }
        render()
        {
            console.warn("checking rerendering");
            return(
                <>
                    <h1>{this.state.count}</h1>
                    {/* <button onClick={()=>this.tipu()}>Hello</button> */}
                    <button onClick={()=>this.setState({count: this.state.count + 1})}>Hello</button>  // It will render when
number is incrasing on every button click
                    <button onClick={()=>this.setState({count: this.state.count })}>Hello</button> // It will still rerender or
render function will update on every button click even when number is not increasing but it can be controlled using
pureComponent. We need to change component to pureComponent import React, { PureComponent } from 'react'
```

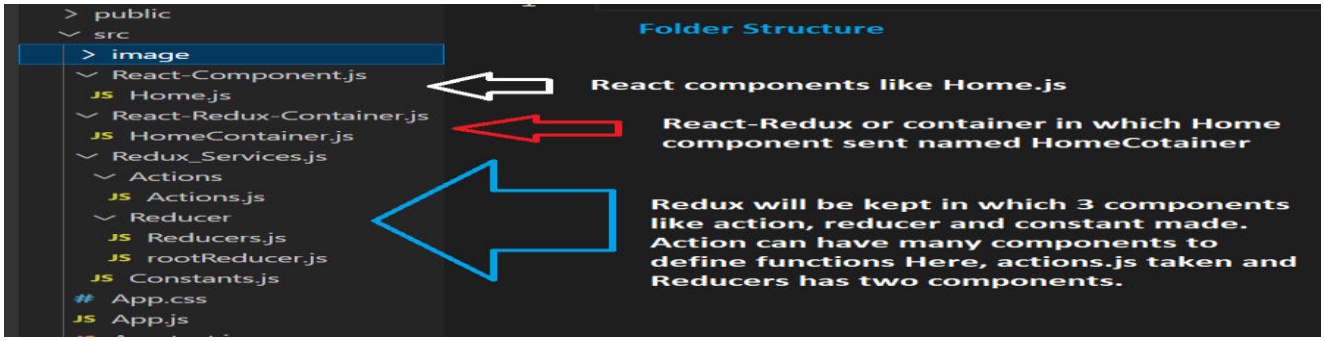
```
class State extends PureComponent{///f

                </>
            )
        }
    }
export default State;
```

File Management of Redux in Details

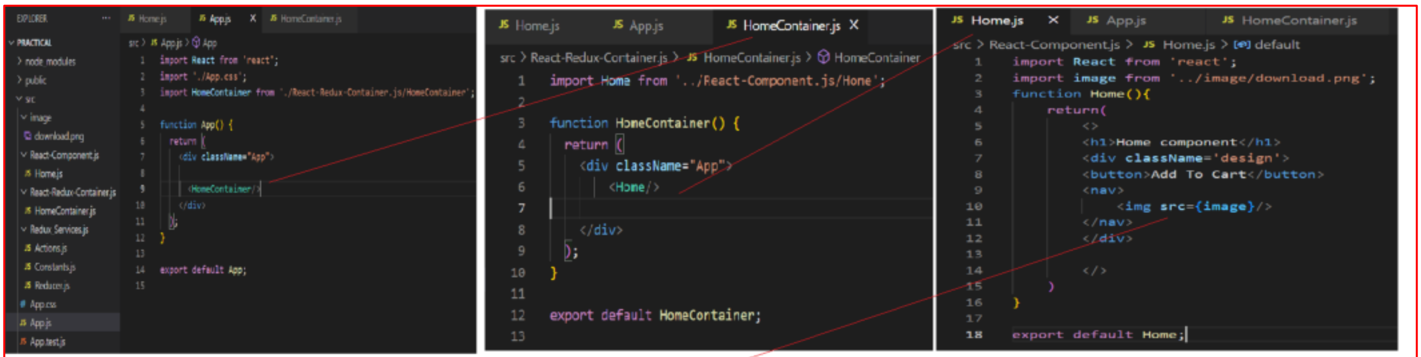




We will also make one file rootReducer.js inside reducer folder that will combine all the reducers into one file. You can give I any name index.js instead of rootReducer.js



Here, we will bring component data in the container(store or React Redux that connect React to Redux) and from container we will pass the data elsewhere, wherever required. In order to put the home data in the container, we need to call <Home/> from container component. So, we will import Home in container component not in App.js component as shown below:-



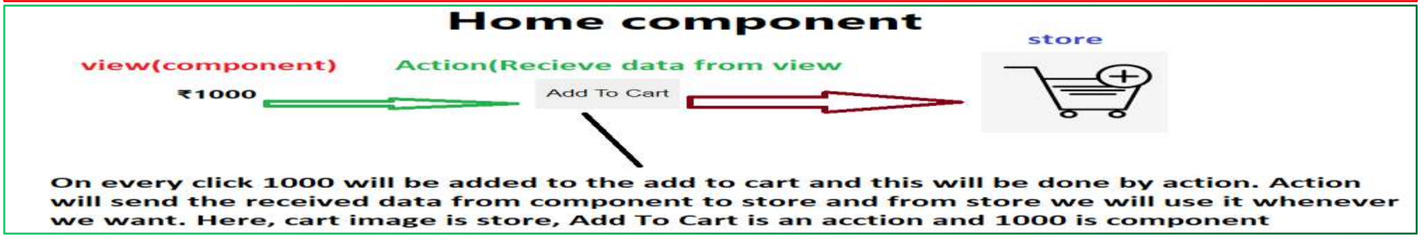
import React from 'react';
import './App.css';
import HomeContainer from './React-Redux-Container.js/HomeContainer';
function App() {
return (
<div className="App">
<HomeContainer/>
</div>
export default App;
import Home from '../React-Component.js/Home';
function HomeContainer() {
return (
<div className="App">
<Home/>
</div>
export default HomeContainer;
import React from 'react';
import image from '../image/download.png';
function Home(){
return(
<h1>Home component</h1>
<div className='design'>
<button>Add To Cart</button>
<nav>

</nav>
</div>
export default Home;

Step 2: called Home and then HomeContainer was linked to main component App.js

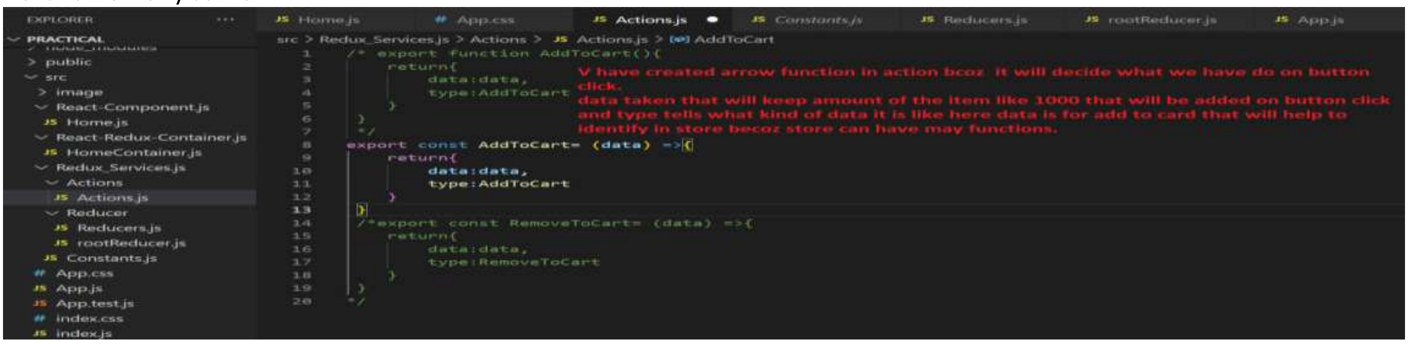
Step1 : We will create home componet and in order to put it HomeContainer we will call it in HomeContainer.js

will use src={image}, curly braces taken we src="" taken in javaScript and to use it in React we take curly braces

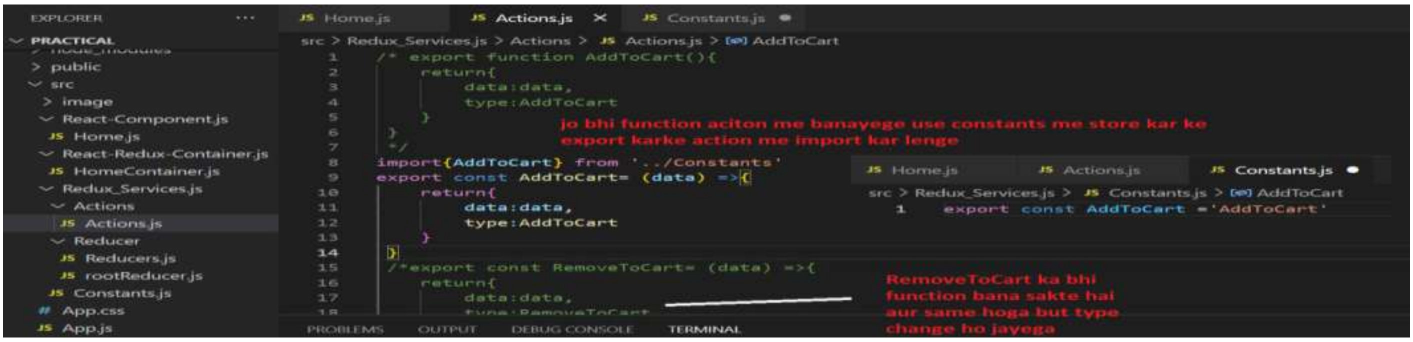


Home component
view(component)
Action(Recieve data from view
store
₹1000
Add To Cart
On every click 1000 will be added to the add to cart and this will be done by action. Action will send the received data from component to store and from store we will use it whenever we want. Here, cart image is store, Add To Cart is an acction and 1000 is component

We use mapDispatchToProps to send the data from action(button click) to store and from store to elsewhere we will use mapStateToPorps. We can give it name x or y as well

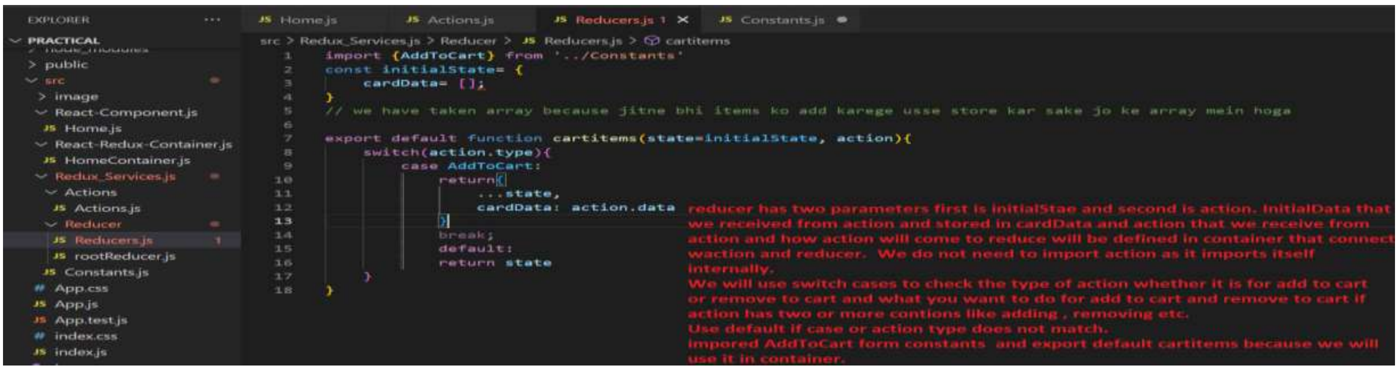


export const AddToCart= (data) =>{
return{
data:data,
type:AddToCart
V have created arrow function in action bcoz it will decide what we have do on button click.
data taken that will keep amount of the item like 1000 that will be added on button click and type tells what kind of data it is like here data is for add to card that will help to identify in store becoz store can have may functions.



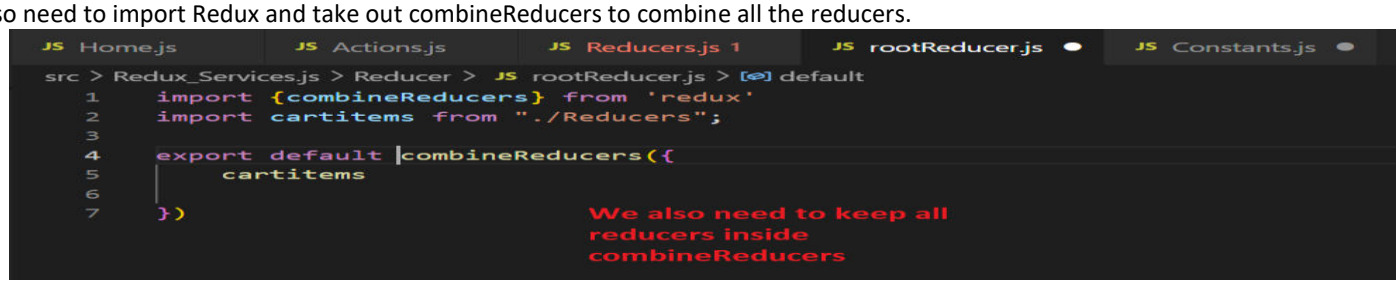
jo bhi function aciton me banayege use constants me store kar ke export karke action me import kar lenge
import{AddToCart} from '../Constants'
export const AddToCart= (data) =>{
export const AddToCart ='AddToCart'
RemoveToCart ka bhi function bana sakte hai aur same hoga but type change ho jayega

We will store the same function in constants so that this file can be used by both actions and reducers. We know that we send data like here price 1000 from actions to reducer and in reducer the sent data like 1000 will be saved in reducer initial data.



import {AddToCart} from '../Constants'
const initialState= {
cardData= [];
// we have taken array because jitne bhi items ko add karege usse store kar sake jo ke array mein hoga
export default function cartitems(state=initialState, action){
switch(action.type){
case AddToCart:
return{
...state,
cardData: action.data
break;
default:
return state
reducer has two parameters first is initialStae and second is action. InitialData that we received from action and stored in cardData and action that we receive from action and how action will come to reduce will be defined in container that connect waction and reducer. We do not need to import action as it imports itself internally.
We will use switch cases to check the type of action whether it is for add to cart or remove to cart and what you want to do for add to cart and remove to cart if action has two or more contions like adding , removing etc.
Use default if case or action type does not match.
impored AddToCart form constants and export default cartitems because we will use it in container.

We know that there can be only one store in one React.js project and for that we need one reducer. If we have more that one reducers then we need to combine those to the rootReducer to make one reducer as shown below:-
We have exported the function cartitems so that it can be used or imported in rootReducer.
We also need to import Redux and take out combineReducers to combine all the reducers.



We use mapDispatchToProps to send the data from action(button click) to store and from store to elsewhere we will use mapStateToPorps. We can give it name x or y as well

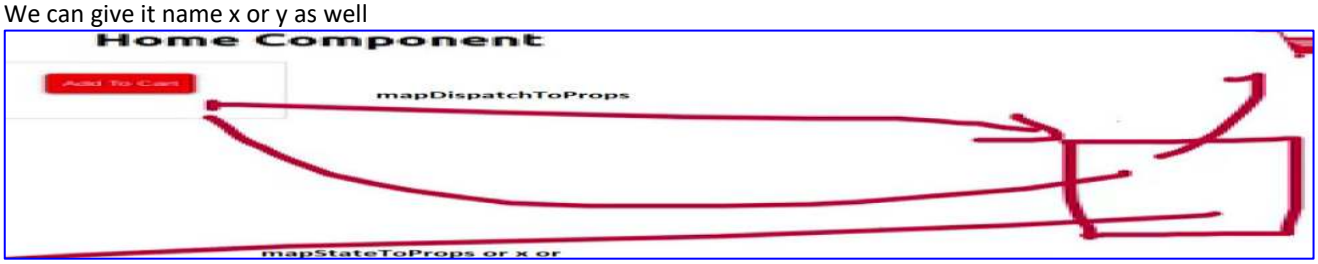
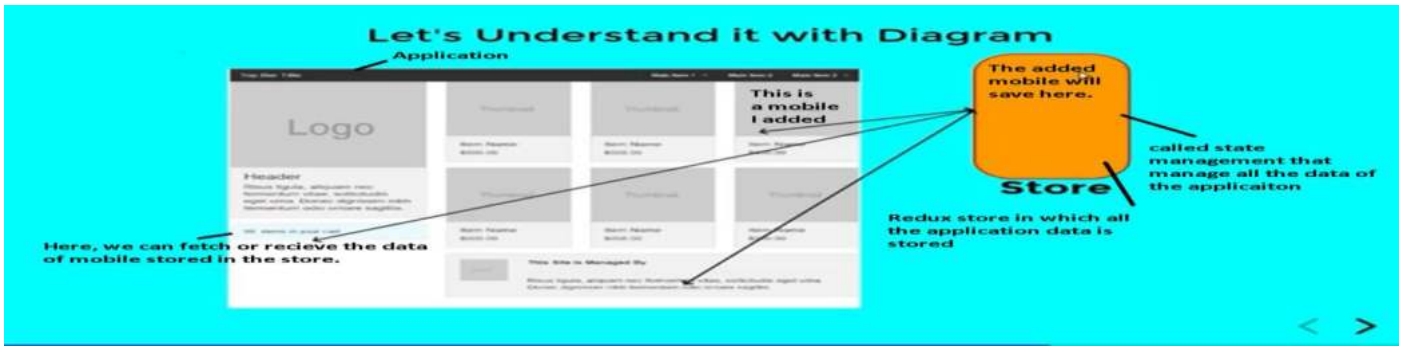


Diagram to understand..



1) What is WordPress?

WordPress is an Open Source Content Management System (CMS) and a blogging tool, based on PHP and MySQL. It is free of cost, and you can use it for any personal and commercial website.

Wordpress is the most popular blogging system and used for more than 60 million websites. It is licensed under GPL.

2) Is a website on WordPress safe and secure?

WordPress is safe and secure to operate. You need not worry about anything but, still, it is suggested to update your site with latest WordPress version to avoid hacking.

3) Is there any limitation for WordPress website?

WordPress can be used for e-commerce sites, membership sites, photo galleries and any other type of site you can think. Following are some disadvantages of WordPress:

- Using several plugins can make the website heavy to load and run.
- PHP knowledge is required to make modifications or changes in the WordPress website.
- Sometimes software needs to be updated to keep the WordPress up-to-date with the current browsers and mobile devices. Updating WordPress version leads to loss of data, so a backup copy of the website is required.

- Modifying and formatting the graphic images and tables is difficult.

4) When was the Wordpress released initially? When is it announced as open source software?

WordPress was initially released on 27th May 2003 by Matt Mullenweg and Mike Little. WordPress was announced as open source in October 2009.

More Details...

5) What is the latest version of WordPress?

The latest version of WordPress is 4.9.8 released on Aug 02, 2018.

6) What are the system requirements for installing WordPress?

Database - MySQL 5.0 +

- Web Server -
    - WAMP (Windows)
    - LAMP (Linux)
    - XAMP (Multi-platform)
    - MAMP (Macintosh)
- Operating System - Cross-platform
- Browser Support - IE (Internet Explorer 8+), Firefox, Google Chrome, Safari, Opera
- PHP Compatibility - PHP 5.2+

7) What are the steps should be followed for installing WordPress?

- Download Wordpress from www.wordpress.org
- Extract the downloaded files and upload it on your web server or localhost.
- Open your web browser and navigate to the WordPress file path. Then you see the first screen of the WordPress installer.
- In the next step, please select your preferred language.
- In the next step, you see the information needed for the database before proceeding the WordPress installation.
- enter the details of your Mysql database.
- Wordpress checks the details provided by you and gives you a confirmation screen.
- In the next step, you have to enter the administration details.
- After entering the administration details, click on "Install WordPress" button.
- After successful installation, you get a screen stating "success." You can see the username and the password on this screen.
- Enter your username and password on the next screen and click 'login.'

8) Explain the components shown on the Home screen of WordPress.

- Dashboard menu: This component of home screen provides the navigation menu options for posts, media library, pages, and comments on the left side of the screen.
- Screen options: This component of the home screen display different types of widgets which can be shown or hidden on some screen. This component also contains the checkboxes to show or hide different screen options and also allow to customize sections on the admin screen.
- Welcome: This component of the home page help us to customize WordPress theme by clicking on the customize your site button. Also, the center column provides some of the useful links such as creating a page and view the front end of your web page, creating a blog post. Moreover, the last column contains links to the menus, widgets, settings related to comments.
- Quick Draft: This component of the home screen includes a mini post editor which is used to write, save, and publish the post from the admin dashboard. This include title of the draft followed by some notes about the draft and save it as a draft.
- Wordpress news: This component of home screen displays the latest news regarding the latest software update of WordPress.
- Activity: This component of the home screen shows the latest comments, recent posts published. It also allows you to reply, edit, delete, approve, or disapprove the comments. You can also move comments to spam.
- At a Glance: This component of home screen displays an overview of your blog?s posts, number of published posts and pages, and number of comments.

9) Is it possible to SEO a WordPress site to show it on Google first page?

WordPress has an inbuilt SEO search engine benefit. You can also have an additional plug-in in WordPress to help with SEO and rank on a favorite search engine like Google.

10) What is the difference between character 23 and x23?

Here 23 specifies the octal 23 and x23 determine the hex 23.

11) What are the hooks? Define different types of hooks in WordPress.

Hooks enable users to create WordPress themes or plug-ins with shortcode without changing the original files.

There are two types of hooks:

Action hooks: Action hooks facilitate you to insert an additional code from an outside resource.

Filter hooks: Filter hooks facilitate you to add content or text at the end of the post.

12) What are the most exciting and useful features of WordPress?

These are the features which make WordPress very popular:

- Easy to install and upgrade
- In-built SEO engine
- Free and easy theme selection
- Flexibility
- Multilingual- available in more than 70 languages
- Own data- no unwanted advert on your website
- Flexibility and Easy publishing option

13) Why does WordPress use MySQL?

Following are the reasons to use MySQL with WordPress:

- Open source
- Extremely fast
- A widely available database server
- Supported by low-cost Linux hosting

14) How many tables are there in WordPress by default?

At present version, there are about 11 tables in WordPress by default. You can check the number of tables in WordPress by phpMyAdmin.

- wp_commentmeta
- wp_comments
- wp_links
- wp_options
- wp_postmeta
- wp_posts
- wp_terms
- wp_term_relationships
- wp_term_taxonomy
- wp_usermeta
- wp_users

Note: The number of tables may be changed with successive releases.

15) What is by default prefix of WordPress tables?

The wp_ is by default prefix of WordPress tables.

16) In WordPress, objects are passed by value or by reference.

In WordPress, all objects are passed by value.

More Details...

17) How can you call a constructor for a parent class?

You can call a constructor for a parent class by this way:

Parents:: constructor($value)

More details...

18) In which cases, WordPress is not suitable for a website?

These are some situations when WordPress is not recommended:

- If the client is working on the non-CMS based project.
- For sophisticated and innovative e-commerce sites.
- Sites which require custom scripting solutions.

More details...

19) Is WordPress the best CMS or any other CMS is better than WordPress?

No doubt WordPress is good CMS, but Drupal and Joomla are considered better CMS than WordPress to work.

More details...

20) Which is considered more secured wordpress.com or wordpress.org?

Wordpress.com is relatively more secure than wordpress.org because they limit the themes and does not allow plugin's installation. However, security depends on the hosting company of your website and also what steps they are taking to prevent the security problems.

21) Explain posts in WordPress.

Posts allow you to write a blog and post it on your site. They are listed in reverse chronological order on the front page of your blog.

More Details...

22) Explain pages in WordPress.

Pages are different from posts. They are static, and they do not change often. You can add pages containing information about you and your site.

More details...

23) What is the difference between WordPress posts and pages?

WordPress posts are content published on a site with an exact date and time. They can be categorized systematically by category and tags. They are listed in reverse chronological order on a website.

WordPress pages are static and do not frequently change such as contact us, about us, privacy policy. They don?t have a date and time published. However, the database stores their published date and time.

More details...

24) What is a loop in WordPress?

WordPress uses PHP codes to display posts. This PHP code is known as a loop.

More details...

25) How can you disable WordPress comment?

You can disable the WordPress comment on the dashboard. On the panel, under the options- discussion you find "Allow people to post the comment." Uncheck this to disable comment.

More details...

26) How can you edit a WordPress comment?

You can edit WordPress comment using the dashboard. From the panel, under the Comments option, select edit to edit a comment.

27) Explain moderate comment in WordPress.

Comments by visitors on a post are not published directly unless the admin provides it. It is called moderation. To change comment moderation setting, select Settings option from the dashboard and check the option "Comment must be manually approved."

More details...

28) How to allow only registered users to comment on WordPress?

If you don't want a comment from a new user on your blog, check the option "Users must be registered and logged in to comment" from Discussions under the Settings option.

More details...

29) Explain Avatar and Gravatar in WordPress.

Word Avatar is used for a user's profile image in online communications. Gravatar is a web-based service which allows its users to use the Avatar image.

More details...

30) How can you handle the situation if your WordPress site is hacked?

You should follow these steps:

- Install security plugins like WP security
- Re-install the latest version of WordPress
- Change password and user-ids for all your users
- Check your themes and plug-ins are up to date

More details...

31) Explain Categories in WordPress.

Categories allow a user to divide its content into different sections. Different topics publishing on a single website can be divided into different groups. It tells a reader what a post is about and they can easily find their content from a lot.

More details...

32) How to convert a category into the tag?

WordPress provides you an option to change category into tag and tag into a category. For this, you need to install Categories and Tags Converter from Import option under Tools section.

More details...

33) Explain Tags in WordPress.

With the help of tags, similar posts can be grouped. Hence, it makes more comfortable for the users to search for a particular post. Tags are similar to categories but still different.

More details...

34) Explain the difference between WordPress Categories and Tags.

WordPress Categories are broad-ranging. It helps a user to identify about a blog. It is possible for a post to have more than one category. A post must have at least one category.

WordPress Tags are like categories, but they are used to describe a post more specifically. Tags are not necessary for every post.

35) How many types of users WordPress have?

WordPress user role determines access permission to the users of a WordPress site.

- **Administrator**: They have full rights over a site.
- **Editor**: They deal with the content section of a website.
- **Author**: They only deal with their posts. They can delete their post even after publishing.
- **Contributor**: A contributor doesn't have the right to publish their post or page. They need to send it to the administrator for review.
- **Subscriber**: Users who are subscribed to your site can log in and update their profile.
- **Follower**: They don't have any right. They can only read and comment on your post.
- **Viewer**: They can only read and comment on a post.

More details...

36) What are WordPress Themes?

With the help of a WordPress theme, you can design the layout and appearance of your website in the front-end.

More details...

37) How will you select a WordPress theme?

WordPress theme should complement your site. It can be either free or paid. An ideal theme should have qualities like simple, responsive, supports plugins, SEO friendly.

38) What is the difference between custom themes and normal themes?

Custom themes are very straightforward to format. You don't need much technical knowledge to change in the coding part. The custom theme allows for SEO search, but with an SEO plugin available it would not make much difference to a normal theme.

39) What is the difference between free and paid WordPress theme?

Free and premium WordPress themes are different in many ways. With free themes, you have to compromise with the quality as many other sites using the same theme. Problems in premium themes solved more quickly than in free themes.

More details...

40) What are WordPress Plugins?

WordPress plugins are programs written in PHP scripting language that extends the functionality or add some new features to a WordPress site. They provide additional functions to an application.

More details...

41) When will you not be able to see the plugin menu on a WordPress site?

When a site is hosted on free wrodpress.com, plugin menu will not appear. Also if you don't have an account for the administrator level, you can't see the plugin menu.

More details...

42) What are the plugins you can use to create a contact form in WordPress?

You can use a plug-in like Gravity forms or a free plugin contact form 7 to create a contact form in WordPress.

More details...

43) Explain the difference between WordPress Themes and Plugins.

A theme is an external effect of a website's presentation and designing. Whereas, a plugin is an interior effect which adds or remove some functions to a site.

You can customize some functionality in themes instead of installing the plugin, but plugins can't replace themes.

More details...

44) How to update a WordPress site?

You should always update your site to the latest version of WordPress. Whenever there is a new WordPress version, you receive an update message on your admin screen.

There are two methods to update a WordPress site:

- One-click update
- Manually update

45) How to improve security on a WordPress site.

Security in a WordPress site is essential as these websites are favorite targets for hackers. A site can't be perfectly secured, but one can at least reduce the risk by applying different security measures.

Some of the measures to secure a WordPress site are given below:

- Choosing a host
- Strong password
- 2 step login authentication
- Limit number of login attempts
- Use SSL
- Keep site updated

46) How to reset password in WordPress?

If you forget or lost the password of your account, WordPress gives you an option to reset it through different methods.

There are four ways to reset WordPress password:

- Change password from the dashboard
- Reset password via email
- Manually reset the password through database via PhpMyAdmin
- Using emergency password reset script

47) How to optimize WordPress site performance?

Optimization increases the speed of your site and gives the best possible outcome. It improves the ranking of a website.

There are many tricks to optimize a WordPress site. Some of them are given below:

- Use CDN
- Use a caching plugin
- Use a simple theme/framework
- Keep site updated
- Split long posts into smaller pages

48) What are template tags in WordPress?

Template tags are used within themes to retrieve content from your database. It is a piece of code that tells WordPress to get something from the database.

49) What are meta-tags in WordPress?

Meta-tags are keyword and description used to display website or page information.

50) What are custom fields in WordPress?

Custom fields are also called post meta. It is a feature through which users can add arbitrary information when writing a post. These metadata can be displayed using template tags in WordPress themes.

51) How to take the backup of a WordPress site?

A backup is just the copy of your original site through which you can restore your original site if anything goes wrong. Keeping a backup on the safe side is always advisable.

A site is generally made up of two parts. One is all the installed themes/plugins, media and other is Database which stores all your blogs, posts, and comments. Without files, there is no site, and without the database, there is no data. Hence both of them is important and need a backup.

52) Why is Backup important?

Backup allows you to retrieve your site back if something goes wrong or if your site breaks. There could be many reasons for this like external attack, hacking, server down.

You can lose everything from data to all the posts on your site. The backup is essential to avoid it.

Securing your backup is also an important issue. Make sure to update your plugins and themes regularly or in a specified interval. Use strong Username and Password. Database name should also be uncommon.

53) How to Backup WordPress root Files?

There are many files in your WordPress directory. These files can be downloaded back from wordpress.org site. Some of them can be retrieved back but some can't.

The wp-content directory contains all your installed themes and plugins including all your media files like images, audio or video files which you uploaded on your site. Hence, it makes it personal and unique.

The wp-config.php file consists of your Database and other personnel options which make it also a unique file.

Both the above files can't be replaced by other default files. Hence, they need to be backed up.

54) Explain the steps involved in the backup of the database.

In the database, all your posts, media files, comments, and metadata stored. It also contains user's information and all your plugin settings. All these information are personal and unique. If you lose them, they are gone permanently. Select your WordPress database (which you created during the installation of WordPress).

- Click on Export.
- Quick is used when the database is small. Custom is used when the database is large. Suppose we are choosing Custom.
- After choosing Custom, a table appears. Select all the tables.
- Now come to the Output section.
- Choose SQL from the Format drop-down menu.
- Check "Add drop table." It is useful for overwriting an existing table.
- Check "If not exists." It prevents errors when tables are already there.
- Now click Go. Your data will be saved in your system.

55)Which licensing authority is responsible for WordPress?

WordPress is licensed under GPLv2 (GNU General Public License) which makes it free and open-source software. Each copy of WordPress has a licensed copy with it.

56) What is GNU?

The GNU General Public License is called GPL in short. It has some terms and conditions to copy, modify and distribute the software licensed under its name.

GNU makes sure that any software source-code licensed under it has to make originating source code open and freely available to all its users. Here, freely doesn't mean by cost but it says that it is freely available to users to distribute and modify the code, but they can't impose any restrictions on further distribution, the source code has to be made available.

1.Which of the following are responsible for data handling in ReactJS?
Select one or more:
A. Events and props
B. Events and states
C. States and components
D. States and props

C&d

2. In ReactJS, which of the following lifecycle methods is NOT invoked on server-side rendering?
Select one or more:
A. componentWillMount
B. componentDidMount
C. componentWillUnmount
D. render

Ans:- . componentWillMount

3. What is the purpose of the state in ReactJS?
Select one or more:
A. To store data that is passed to child components
B. To store data that cannot be changed within the component
C. To store data that can be changed within the component
D. To store data that is passed from parent components

Ans:- .C. To store data that can be changed within the component

4. What is the purpose of the props in ReactJS?
Select one or more:
A. To store data that can be changed within the component
B. To store data that is passed to child components
C. To store data that cannot be changed within the component

D. To store data that is passed from parent components

Answer: A) Virtual DOM is used in React.js to increase performance.
B) ReactJS is a user-interface framework.
C) Answer: D) props are used to pass data to components from outside.
D) Answer: B) Jordan Walke created React.js.
E) Answer: B) React.js is written in Javascript.
F) A) Babel is javaScript compiler.
Answer : D) 8080 is the default where webpack-dev-server runs.
Answer: A) A valid react component can return only 1 element.
Answer: A) A state in React.js is also known as the internal storage of the component.
Answer: C) this.values is used to access the state of a component from inside of a member function.
Yes, Props are methods into other components?
Answer: B) Arbitrary inputs of components in react also known as Props.
Yes,React.js covers only the view layer of the app.
Answer: B) FLux helps react for keeping their data uni-directional.

Answer: A) preventDefault() is used to prevent default behavior in react.
Answer: C) Components are the smallest building blocks of ReactJS.
Answer: C) ReactJS is mainly used for building user interfaces.
Answer: A) this.setState() is used to change the state of react component.
Answer : D) props act as the input of the class-based components.
Answer: B)Extends are used to create a class inheritance.
Answer: A) The correct data flow sequence of flux concept is :
Action -> Dispatcher -> Store -> View
Answer: C) There are three ways of defining variables in ES6: let, var and const.
Answer : A)renderComponent is a must-have for every ReactJS component.
Answer: B) State and props is used to handle data in react.
Answer: B) super() refers to the parent class in ReactJS.
Answer : B)render() is called to render HTML to a web page in react.
Answer: B) setState invoke code after the setState operation is done.
Answer: C) ReactDOM.destroy() is not a part of ReactDOM.
Answer: C) Source of truth is DOM is correct in the context of uncontrolled components in ReactJS.
34. In which directory is react component saved?
js/components in
Answer: A) Webpack is a model blunder.
Yes, React merges the objects you provide into the current state using setState().
Answer: A)The key should be unique among his siblings only.
Answer : D) shouldComponentUpdate should be override to stop the component from updating.
41. Which of the following is used to access a function fetch() from h1 element in JSX?
Answer : D) The correct command is <h1>{fetch()}</h1>.
Answer : D) Stackoverflow error occurs when setState() is called inside render() method.
Answer: A) It will be rendered as enabled if you render an input element with disabled = {false}.
Answer: A)When we want to replace redux, useReducer is used over useState.
Answer: C)Jest is most often associated with react.
Answer: A) React.lazy is used to handle code splitting.
Answer : D) When you need the browser to paint before the effect runs, useLayoutEffect used.
Answer: C) Declarative is commonly used to describe react applications.
Answer: C) Ref is used to directly access the DOM node.

TM
Tanmesh Mishra
Started on Wednesday, 15 February 2023, 1:18 PM
State Finished
Completed on Wednesday, 15 February 2023, 1:37 PM
Time taken 18 mins 52 secs
Marks 26.00/40.00
Grade 6.50 out of 10.00 (**65**%)2/15/23, 2:15 PM
REACT JS QUESTIONNAIRE: Attempt review
https://msassesments.com/mod/quiz/review.php?attempt=188&cmid=45
2/27
Question 1
Incorrect
Mark 0.00 out of 1.00
Response history
Step
Time
Action
State
Marks
1
15/02/23, 13:18:20
Started

Not yet answered
2
15/02/23, 13:18:33
Saved: States and props
Answer saved
3
15/02/23, 13:37:12
Attempt finished
Incorrect
0.00

Which of the following are responsible for data handling in ReactJS?
Select one or more:
A. Events and props
B. Events and states
C. States and components
· States and components
· States and components
States and components
D. States and props

Your answer is incorrect.
The correct answer is:
Events and props

Question 2

Incorrect
Mark 0.00 out of 1.00

Response history

Step
Time
Action
State
Marks

1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:31:24
Saved: componentDidMount
Answer saved
3
15/02/23, 13:37:12
Attempt finished
Incorrect
0.00

In ReactJS, which of the following lifecycle methods is NOT invoked on server-side rendering?
Select one or more:
A. componentWillMount
B. componentDidMount
C. componentWillUnmount
D. render

Your answer is incorrect.
The correct answer is:
componentWillMount

Question 3

Incorrect
Mark 0.00 out of 1.00

Response history

Step
Time
Action
State

Marks
1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:19:00
Saved: ReactJS is a programming language
Answer saved
3
15/02/23, 13:37:12
Attempt finished
Incorrect
0.00

Which of the following statements about ReactJS is true?
Select one or more:
A. ReactJS is a back-end framework
B. ReactJS is a database management system
C. ReactJS is a programming language
D. ReactJS is a front-end library
Your answer is incorrect.
The correct answer is:
ReactJS is a front-end library

Question 4

Correct
Mark 1.00 out of 1.00

Response history

Step
Time
Action
State
Marks
1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:19:12
Saved: To prevent memory leaks in the application ; To improve the performance of the application
Answer saved
3
15/02/23, 13:37:12
Attempt finished
Correct
1.00

What is the purpose of the virtual DOM in ReactJS?
Select one or more:
A. To reduce the amount of memory used by the application
B. To improve the readability of the code
C. To prevent memory leaks in the application
D. To improve the performance of the application
Your answer is correct.
The correct answer is:
To improve the performance of the application

Question 5

Correct
Mark 1.00 out of 1.00

Response history

Step

Time

Action

State

Marks

1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:27:09
Saved: Two-way data binding ; One-way data binding
Answer saved
3

15/02/23, 13:37:12

Attempt finished

Correct

1.00

Which of the following is NOT a feature of ReactJS?
Select one or more:

A. Two-way data binding

B. One-way data binding

C. Component-based architecture
D. Virtual DOM
Your answer is correct.
The correct answer is:
Two-way data binding

Question 6

Correct
Mark 1.00 out of 1.00

Response history

Step

Time

Action

State

Marks

1
15/02/23,
13:18:20
Started
Not yet
answered
2
15/02/23,
13:19:47
Saved: To store data that is passed to child components ; To store data that can be changed within the component ; To store data that is passed from parent components
Answer
saved
3

15/02/23,

13:37:12

Attempt finished

Correct

1.00

What is the purpose of the state in ReactJS?
Select one or more:

A. To store data that is passed to child components

B. To store data that cannot be changed within the component

C. To store data that can be changed within the component

D. To store data that is passed from parent components

Your answer is correct.

The correct answer is:
To store data that can be changed within the component

Question 7

Correct
Mark 1.00 out of 1.00

Response history

Step	Time	Action	State	Marks
1	15/02/23, 13:18:20	Started	Not yet answered	
2	15/02/23, 13:20:06	Saved: To store data that is passed to child components ; To store data that cannot be changed within the component ; To store data that is passed from parent components	Answer saved	
3	15/02/23, 13:37:12	Attempt finished	Correct	1.00

What is the purpose of the props in ReactJS?
Select one or more:
A. To store data that can be changed within the component
B. To store data that is passed to child components
C. To store data that cannot be changed within the component
D. To store data that is passed from parent components

Your answer is correct.
The correct answer is:
To store data that is passed from parent components

Question 8

Incorrect
Mark 0.00 out of 1.00

Response history

Step	Time	Action	State	Marks
1	15/02/23, 13:18:20	Started	Not yet answered	
2	15/02/23, 13:20:28	Saved: To update the props of the component ; To update the state of the component ; To render the component to the virtual DOM	Answer saved	
3				

15/02/23,

13:37:12

Attempt finished

Incorrect

0.00

What is the purpose of the render() method in ReactJS?
Select one or more:
A. To render the component to the DOM
B. To update the props of the component
C. To update the state of the component
D. To render the component to the virtual DOM
Your answer is incorrect.
The correct answer is:
To render the component to the DOM

Question 9

Correct
Mark 1.00 out of 1.00

Response history

Step

Time

Action

State

Marks

1
15/02/23,
13:18:20
Started
Not yet
answered
2
15/02/23,
13:20:45
Saved: To initialize the state of the component ; To update the state of the component ; To render the component to the virtual DOM
Answer
saved
3

15/02/23,

13:37:12

Attempt finished

Correct

1.00

What is the purpose of the constructor() method in ReactJS?
Select one or more:
A. To initialize the state of the component
B. To update the state of the component
C. To render the component to the DOM
D. To render the component to the virtual DOM
Your answer is correct.
The correct answer is:
To initialize the state of the component

Question 10

Correct
Mark 1.00 out of 1.00

Response history

Step

Time

Action

State

Marks
1
15/02/23,
13:18:20
Started
Not yet
answered
2
15/02/23,
13:21:12
Saved: To specify the class name of the component ; To specify the type of the component ;
To specify a unique identifier for the component
Answer
saved
3
15/02/23,
13:37:12
Attempt finished
Correct
1.00
What is the purpose of the key prop in ReactJS?
Select one or more:
A. To specify the class name of the component
B. To specify the type of the component
C. To specify a unique identifier for the component
D. To specify the style of the component
Your answer is correct.
The correct answer is:
To specify a unique identifier for the component
Question 11
Correct
Mark 1.00 out of 1.00
Response history
Step
Time
Action
State
Marks
1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:21:54
Saved: All of the above
Answer saved
3
15/02/23, 13:37:12
Attempt finished
Correct
1.00
Which of the following statements is true about React hooks?
Select one or more:
A. All of the above
B. Hooks allow you to use state and other React features without writing a class
C. useState is a hook that can be used to add state to a functional component
D. Hooks can only be used in functional components
Your answer is correct.
The correct answer is:
All of the above
Question 12
Correct

Mark 1.00 out of 1.00

Response history

Step

Time

Action

State

Marks

1
15/02/23,
13:18:20
Started
Not yet
answered
2
15/02/23,
13:22:27
Saved: A header with the text "My Component" and a paragraph with the text "This is
my first React component!"
Answer
saved
3

15/02/23,

13:37:12

Attempt finished

Correct

1.00

```
import React from 'react';
function MyComponent() {
return (
<div>
<h1>My Component</h1>
<p>This is my first React component!</p>
</div>
);
}
export default MyComponent;
```

Select one:

A. A header with the text "My Component" and a paragraph with no text

B. A header with the text "My Component" and a paragraph with the text "This is my first React component!"

C. This code will not compile due to a missing import statement.

D. A paragraph with the text "My Component This is my first React component!"

Your answer is correct.

The correct answer is: A header with the text "My Component" and a paragraph with the text "This is my first React component!"

Question 13

Correct
Mark 2.00 out of 2.00

Response history

Step

Time

Action

State

Marks

1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:22:49
Saved: "Hello, John!"
Answer saved
3

15/02/23, 13:37:12

Attempt finished

Correct

2.00

```
import React from 'react';
function MyApp() {
const name = 'John';
return <h1>Hello, {name}!</h1>;
}
export default MyApp;
```

Select one:

A. "Hello, John!"

B. "Hello, {name}!"

C. This code will not compile due to a missing import statement.

D. This code will not compile due to a syntax error.

Your answer is correct.

The correct answer is:

"Hello, John!"

Question 14

Incorrect

Mark 0.00 out of 2.00

Response history

Step	Time	Action	State	Marks
1	15/02/23, 13:18:20	Started	Not yet answered	
2	15/02/23, 13:23:16	Saved: A div containing "Item 1", "Item 2", and "Item 3"	Answer saved	
3	15/02/23, 13:37:12	Attempt finished	Incorrect	0.00

```
import React from 'react';
function MyList() {
const items = ['Item 1', 'Item 2', 'Item 3'];
return (
<ul>
{items.map((item) => (
<li>{item}</li>
))}
</ul>
);
}
export default MyList;
```

Select one:

A. A div containing "Item 1", "Item 2", and "Item 3"

B. A bulleted list containing "Item 1", "Item 2", and "Item 3"

C. This code will not compile due to a missing import statement.

D. A numbered list containing "Item 1", "Item 2", and "Item 3"

Your answer is incorrect.

The correct answer is:

A bulleted list containing "Item 1", "Item 2", and "Item 3"

Question 15

Correct

Mark 1.00 out of 1.00

Response history

Step	Time

Action

State

Marks

1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:23:47
Saved: The console will log the message "Button clicked!".
Answer saved
3

15/02/23, 13:37:12

Attempt finished

Correct

1.00

```
import React from 'react';
function MyButton() {
function handleClick() {
console.log('Button clicked!');
}
return <button onClick={handleClick}>Click me</button>;
}
export default MyButton;
```

Select one:
A. The button text will change to "Button clicked!".
B. The console will log the message "Button clicked!".
C. The browser will navigate to a new page.
D. Nothing will happen.
Your answer is correct.
The correct answer is:
The console will log the message "Button clicked!".

Question 16

Correct
Mark 2.00 out of 2.00

Response history

Step

Time

Action

State

Marks

1
15/02/23, 13:18:20
Started
Not yet answered
2
15/02/23, 13:24:14
Saved: An image with the specified path and alt text.
Answer saved
3

15/02/23, 13:37:12

Attempt finished

Correct

2.00

```
import React from 'react';
function MyImage() {
return <img src="path/to/my/image.png" alt="My image" />;
}
export default MyImage;
```

Select one:
A. A broken image icon.
B. This code will not compile due to a missing import statement.
C. An image with the specified path and alt text.
D. This code will not compile due to a syntax error.
Your answer is correct.
The correct answer is:

Question 17

Incorrect
Mark 0.00 out of 3.00

Response history

Step	Time	Action	State	Marks
1	15/02/23, 13:18:20	Started	Not yet answered	
2	15/02/23, 13:24:58	Saved: A div containing a heading that says "Count: 0" and a button that says "Click me!" that, when clicked, updates the count to 1. Answer saved		

```
import React, { useState } from 'react';
function MyCounter() {
const [count, setCount] = useState(0);
function handleClick() {
setCount(count + 1);
}
return (
<div>
<h1>Count: {count}</h1>
<button onClick={handleClick}>Click me!</button>
</div>
);
}
export default MyCounter;
```

Select one:

A. A div containing a heading that says "Count: 1" and a button that says "Click me!".

B. A div containing a heading that says "Count: 0" and a button that says "Click me!".

C. A div containing a heading that says "Count: 0" and a button that says "Click me!" that, when clicked, updates the count to 1.

D. A div containing a heading that says "Count: 1" and a button that says "Click me!" that, when clicked, updates the count to 2.

Your answer is incorrect.
The correct answer is:
A div containing a heading that says "Count: 0" and a button that says "Click me!".

Response history

Step	Time	Action	State	Marks
3	15/02/23, 13:37:12	Attempt finished	Incorrect	0.00

Question 18

Correct
Mark 4.00 out of 4.00

```
import React from 'react';
class MyForm extends React.Component {
```

```
constructor(props) {
super(props);
this.state = { name: '', email: '' };
}
handleNameChange = (event) => {
this.setState({ name: event.target.value });
}
handleEmailChange = (event) => {
this.setState({ email: event.target.value });
}
handleSubmit = (event) => {
event.preventDefault();
alert(`Name: ${this.state.name}, Email: ${this.state.email}`);
}
render() {
return (
<form onSubmit={this.handleSubmit}>
<label>
Name:
<input type="text" value={this.state.name} onChange={this.handleNameChange} />
</label>
<br />
<label>
Email:
<input type="email" value={this.state.email} onChange={this.handleEmailChange} />
</label>
<br />
<input type="submit" value="Submit" />
</form>
);
}
```


Response history

Step	Time	Action	State	Marks
1	15/02/23, 13:18:20	Started	Not yet answered	
2	15/02/23, 13:33:31	Saved: A form with two input fields (Name and Email) and a Submit button. When the form is submitted, an alert dialog will appear displaying the name and email entered by the user. Answer saved		
3	15/02/23, 13:37:12	Attempt finished	Correct	4.00

export default MyForm;

Select one:

A. A form with two input fields (Name and Email) and a Submit button. When the form is submitted, an alert dialog will appear displaying the name and email entered by the user.

B. A form with two input fields (Name and Email) and a Submit button. When the Submit button is clicked, an alert dialog will appear displaying the name and email entered by the user.

C. A form with two input fields (Name and Email) and a Submit button. When either input field is changed, the state of the component will update to reflect the new value.

D. A form with two input fields (Name and Email) and a Submit button. When either input field is changed, an error message will be displayed indicating that the field is required.

Your answer is correct.

The correct answer is:
A form with two input fields (Name and Email) and a Submit button. When the form is submitted, an alert dialog will appear displaying the name and email entered by the user.2/15/23, 2:15 PM
REACT JS QUESTIONNAIRE: Attempt review
https://msassesments.com/mod/quiz/review.php?attempt=188&cmid=45
22/27

Question 19

Correct
Mark 4.00 out of 4.00

Response history

Step

Time

Action

State

Marks

```
import React, { useState, useEffect } from 'react';
function MyComponent() {
const [count, setCount] = useState(0);
useEffect(() => {
console.log(`The count is now ${count}.`);
}, [count]);
const handleClick = () => {
setCount(count + 1);
}
return (
<div>
<p>You have clicked the button {count} times.</p>
<button onClick={handleClick}>Click me</button>
</div>
);
}
export default MyComponent;
```

Select one:
A. The console will log "The count is now 0.", "The count is now 1.", "The count is now 2.", "The count is now 3.", "The count is now 4.", and "The count is now 5.", and the page will display "You have clicked the button 5 times."

B. The console will log "The count is now 5." five times, and the page will display "You have clicked the button 5 times."
C. The console will log "The count is now 5.", and the page will display "You have clicked the button 5 times."
D. The console will log "The count is now 0.", and the page will display "You have clicked the button 5 times."
Your answer is correct.
The correct answer is:
The console will log "The count is now 0.", "The count is now 1.", "The count is now 2.", "The count is now 3.", "The count is now 4.", and "The count is now 5.", and the page will display "You have clicked the button 5 times."2/15/23, 2:15 PM
REACT JS QUESTIONNAIRE: Attempt review
https://msassesments.com/mod/quiz/review.php?attempt=188&cmid=45
23/27

Step

Time

Action

State

Marks

1
15/02/23,
13:18:20
Started
Not yet
answered

2
15/02/23,
13:34:46
Saved: The console will log "The count is now 0.", "The count is now 1.", "The count is now 2.", "The count is now 3.", "The count is now 4.", and "The count is now 5.", and the page will display "You have clicked the button 5 times."
Answer
saved

3

15/02/23,

13:37:12

Attempt finished

Correct

Question 20

Correct

Mark 5.00 out of 5.00

```
import React, { useState } from 'react';
function MyComponent() {
const [text, setText] = useState('');
const [history, setHistory] = useState([]);
const handleChange = (event) => {
setText(event.target.value);
}
const handleClick = () => {
setHistory([...history, text]);
setText('');
}
return (
<div>
<input type="text" value={text} onChange={handleChange} />
<button onClick={handleClick}>Add to history</button>
<ul>
{history.map((item, index) => (
<li key={index}>{item}</li>
))}
</ul>
</div>
);
}
export default MyComponent;
```

Select one:

A. The page will display an unordered list with one item, "apple".

B. The page will display an unordered list with two items, "apple" and "undefined".

C. The page will display an unordered list with two items, "apple" and "" (empty string).

D. The page will display an unordered list with one item, "" (empty string).

Your answer is correct.

The correct answers are:

The page will display an unordered list with one item, "apple".,

Response history

Step	Time	Action	State	Marks
1	15/02/23, 13:18:20	Started	Not yet answered	
2	15/02/23, 13:36:46	Saved: The page will display an unordered list with one item, "apple".	Answer saved	
3	15/02/23, 13:37:12	Attempt finished	Correct	5.00

The page will display an unordered list with one item, "" (empty string).

Question 21

Not answered

Marked out of 5.00

Response history

Step

Time

Action

State

Marks

1

15/02/23, 13:18:20

Started

Not yet answered

Create a ReactJS component that displays a list of items from an array. The list should have a heading, and each item in the list should have a unique key and a checkbox that can be clicked to mark the item as complete. When an item is clicked, the component should update its state to reflect the change.

27/27

Step

Time

Action

State

Marks

2

15/02/23, 13:37:12

Attempt finished

Not answered