Introduction to HTML

- ✓ HTML Stands for Hyper Text Markup Language
- ✓ HTML used to develop static web pages
- ✓ Current version of HTML is HTML5.X
- ✓ HTML Released by "Tim Berners Lee" in 1991
- ✓ we will execute HTML with the help of "Browsers"

  Ex.

  Google Chrome
  Mozilla
  Opera
  IE
  Safari
  Netscape Navigator
- ✓ IDE is used to develop the software applications

  Ex.

  Notepad
  Edit Plus
  Notepad++
  Visual Studio Code
  Eclipse
- ✓ Visual Studio Code is Recommended IDE to develop web applications
- ✓ Visual Studio Code provided by "Microsoft" and "open-source IDE"
- ✓ CSS Stands for Cascading Style Sheet
- ✓ CSS used to apply styles to web pages

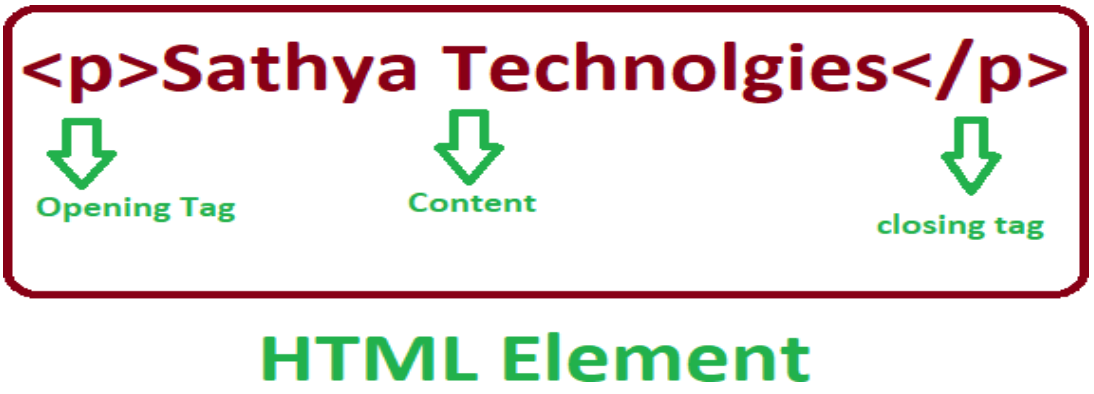
  Ex.

  color
  background-color
  margin
  padding
  border
  border-radius
- ✓ Current version of CSS is CSS3.X
- ✓ Extension of CSS files is ".CSS"
- ✓ JavaScript is the Scripting Language
- ✓ JavaScript used to develop Dynamic Web Pages
- ✓ JavaScript also used to implement the Forms Validations
- ✓ Current version of JavaScript is ES13
- ✓ ES Stands for ECMA Script
- ✓ Extension for JavaScript files is ".js"

- ✓ HTML is TAG Based Markup Language
  Ex.
  <table></table>
  <h1></h1>
  <a>

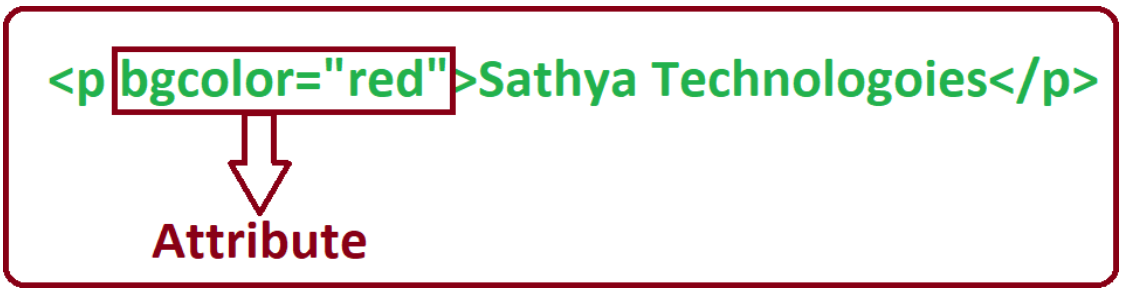
  <img>

  <title></title>
  <head></head>
- ✓ TAGS Divided into two types
  1) container/paired tags
  2) non container tags/non paired tags
- ✓ container/paired tags contains both opening and closing tag
- ✓ non container tags/non paired tags contain only opening tag
- ✓ closing tag should contain "/"
- ✓ combination of opening tag, content and closing tag called as HTML Element



HTML Element

- ✓ Attributes enhances html element functionality
- ✓ Attributes are key and value pairs
- ✓ key and value separated by using "="

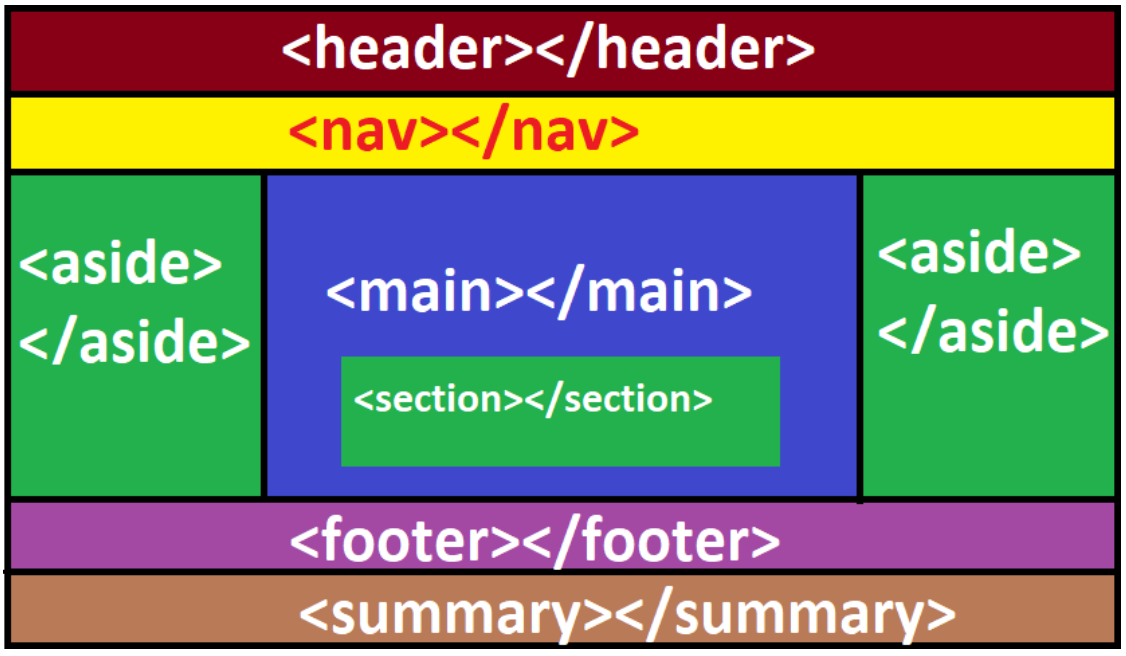


Features of HTML5.X

- ✓ WebSocket
  WebSocket used to implement the chat applications
  WebSocket available in HTML5.X

- ✓ Web Worker
  - Web Worker used to run JavaScript in Background
  - Web Worker also available in HTML5.X
- ✓ Storages API
  - HTML5.X Supports two types of Storages
    1) Local Storage
    2) Session Storage
    - ✓ whenever we close the browser/opens the new tab we won't lost the data from Local Storage
    - ✓ whenever we close the browser/opens the new tab we will lose the data from Session Storage
- ✓ Geolocation API
  - it is used to find the current location of user/device
  - Geolocation API also available in HTML5.X
- ✓ Drag & Drop API
  - This API helps to Drag and Drop HTML Elements
  - This API also available in HTML5.X
- ✓ Semantic Elements
  - HTML5.X Released New Elements/Semantic Elements
  - Semantic Elements increases application readability

Ex.



<header></header> <nav></nav>
<main></main> <section></section> <footer></footer>
<summary></summary>
<audio></audio>

<video></video>

Structure of Webpage

DOCTYPE
ROOT SECTION
    HEAD SECTION
        METADATA
    BODY SECTION
        MAIN CONTENT

- ✓ DOCTYPE Representing version of HTML
- ✓ Browsers only understands the DOCTYPE
- ✓ Below DOCTYPE Representing HTML5.X

```
<!DOCTYPE html>
```

- ✓ <html></html> tag used to create the ROOT SECTION
- ✓ ROOT SECTION Divided into two sections
  1) HEAD SECTION
  2) BODY SECTION
- ✓ <head></head> tag used to create the HEAD SECTION
- ✓ <body></body> tag used to create the BODY SECTION
- ✓ Data about webpage called as METADATA

  Ex.
  - Author
  - Description
  - Title
- ✓ we will define METADATA under HEAD SECTION
- ✓ we will define Main Content under BODY SECTION

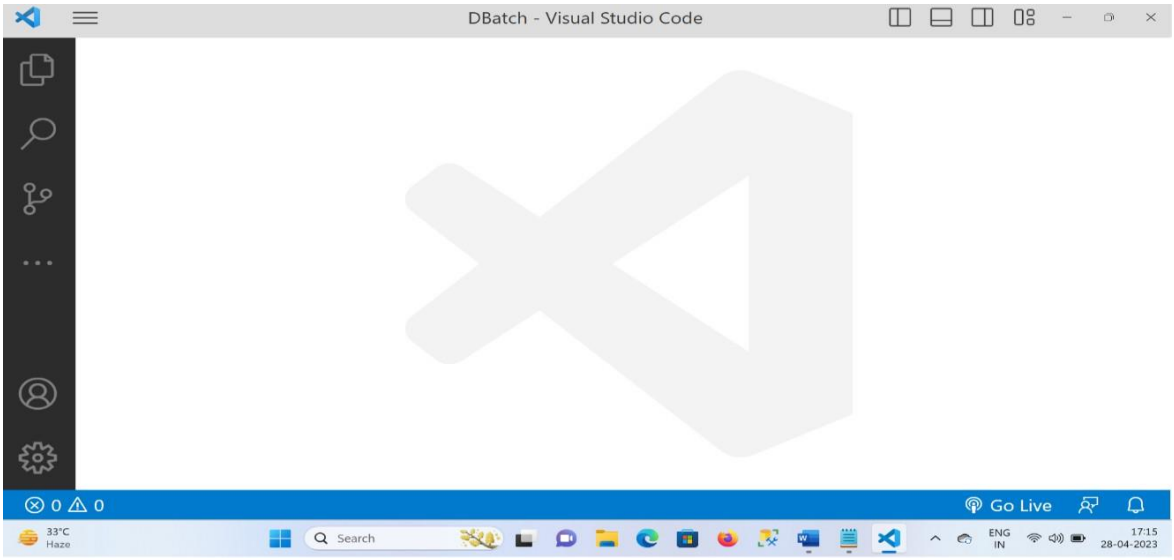
  Ex.
  - Tables
  - Forms
  - Headings
  - Paragraphs
  - Images

```
<!DOCTYPE html>
<html>
    <head>
        //METADATA
    </head>
    <body>
        //MAIN CONTENT
    </body>
```

</html>

Software Installation

1) download and install Visual Studio Code
   - ✓ Visual Studio Code is the IDE.
   - ✓ Visual Studio Code provided by "Microsoft" and "open source".
   - ✓ Visual Studio Code is "recommended" to develop "web applications"
     website: https://code.visualstudio.com/docs/?dv=win
     file: VSCodeUserSetup-x64-1.77.3.exe
2) install Live Server plugin
   - ✓ Live Server plugin watches application changes
   - ✓ Live Server plugin gives "changes notification" to browser
   - ✓ browser will "reload/refresh" automatically
     open the "VSCode" ==> Click "Extensions" ==> Search "Live Server" ==>Select "Live Server" ==> Click "install"
3) install "Dummy Text Generator" Plugin
   - ✓ "Dummy Text Generator" Plugin generates content randomly
     open the "VSCode" ==> click "Extensions" ==> Search "Dummy Text Generator" ==> select "Dummy Text Generator" ==> click "install"
4) Enable "AutoSave" Settings
   - ✓ Application will save Automatically without "ctrl+s"
     open the "VSCode" ==> click the "Manage" ==> click "settings" ==> choose "AutoSave" ==> select "afterDelay"
5) install browser
   - ✓ Google Chrome
   - ✓ Firefox



Practice Test Paper

1) HTML Stands for _____
2) What is current version of HTML _____
3) IDE Stands for_____
4) Which IDE Is Suitable IDE to develop web applications _____
5) CSS Stans for_____
6) Why CSS? Write minimum 3 points

Ans:

7) What is the Current version of CSS _____
8) What is Bootstrap?

9)What is Angular?

10)What is React?

11)Explain MEAN Stack?

12) Explain MERN Stack?

13) Explain MEVN Stack?

14) Write Features of HTML5.X?

15) Explain Semantic Elements in HTML5 with Diagram?

16) How to play audio in HTML5 _____
17)How to play Video in HTML5 _____
18)What is attribute write few points?
Ans:

19) Write basic Structure of Web Pages
Ans:

20) What DOCTYPE?
Ans:

21) Write DOCTYPE for HTML5.X _____

22) What is METADATA?
Ans:

23) Write the Differences Between Container TAGS and Non-Container TAGS?
Ans:

24) How to develop static web pages_____
25)What is the Extension for HTML Pages_____
26) What are the types of tags?

Ans:
27) What are container tags?
Ans:

28) What are non-container Tags>
Ans:

29) What is HTML Element?
Ans:

30) What is the extension of CSS files_____
31) What is the extension of JavaScript files_____
32) How to implement Forms Validations_____
33) How to execute HTML, CSS and JavaScript _____
34) Write the Examples for Browsers?
Ans:

35) Write the Examples for IDE'S?
Ans:

36) Who given VSCode? _____
37) Which IDE Recommended to develop web applications _____
38) How to give line break_____
39)How to draw Horizontal line in webpages _____
40)Are comments executed by browsers? (yes/no) _____
41)Write the Syntax for HTML Comments
Ans:

HEADINGS

- ✓ Headings in HTML
- ✓ External CSS
- ✓ Padding in CSS
- ✓ Block Level Elements

Headings in HTML

- ✓ HTML Supports 6 types of Headings

<h1></h1> main heading
<h2></h2> sub heading
<h3></h3> ""
<h4></h4> ""
<h5></h5> ""
<h6></h6> ""

External CSS

- ✓ <link> tag, used to include the external CSS file
- ✓ <link> tag is non paired tag
- ✓ <link> tag supports two attributes

1) href
2) rel

Padding

- ✓ space around the content called as padding

Ex.

padding:20px;

Block Level Element

- ✓ Block Level Elements Starts with new line
- ✓ Headings are Example for Block Level Elements

headings.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Headings</title>
    <link rel="stylesheet" href="headings.css">
  </head>
  <body>
    <h1>ExcelR</h1>
    <h2>ExcelR</h2>
    <h3>ExcelR</h3>
    <h4>ExcelR</h4>
```

```
    <h5>ExcelR</h5>
    <h6>ExcelR</h6>
  </body>
</html>
```

headings.css

```
h1,h2,h3,h4,h5,h6{
  color: red;
  background-color: yellow;
  text-align: center;
  padding: 20px;
  border: 1px solid red;
  width: 50%;
  border-radius: 0px 60px;
  box-shadow: 0px 0px 20px red;
}
```

Output

PRACTICE PAPER

1) Write the Tags to create Headings in HTML

   Ans:

2) How to apply external CSS?

   Ans:

3) What is padding in CSS?

4) What are Block Level Elements?

   Ans:

5) Are Headings Block Level Elements? (yes/no) _____

6) write the html code to develop below application with headings

Text Formatting Tags

1) ----------bold text
2) ----------strong text
3) ----------italic text
4) ----------emphasized text
5) <u></u> ----------underlined text
6) <del></del>----------deleted text
7) <strike></strike>----------strike text
8) <mark></mark>----------marked text
9) ----------superscript
10) ----------subscript
11) <big></big>----------big text
12) <small></small> ----------small text
13) <code></code>----------represent the computer code
14) <var></var> ----------mathematical formulas
15)
----------line break
16) <hr>----------horizontal line
17) <address></address>------- display address
18) <abbr></abbr>----------abirritations
    title attribute used to define abbreviation
19) <bdo dir=""></bdo> --------bi directional override
20) <blockquote></blockquote> --- represent the quotations

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Text Formatting Tags</title>
  </head>
  <body>
    This is <b>Bold</b> Text <br>
    This is <strong>strong</strong> text <br>
    This is <i>Italic</i> Text <br>
    This is <em>Emphasized</em> Text <br>
    This is <u>underlined</u> Text <br>
    This is <del>Deleted</del> Text <br>
    This is <strike>Striked</strike> Text <br>
```

```
    This is <mark>Marked</mark> Text <br>
    a<sup>2</sup>+b<sup>2</sup> <br>
    H<sub>2</sub>O <br>
    This is <big>Big</big> Text <br>
    This is <small>small</small> Text <br>
    <code>for (int i=0; i<10; i++) {} </code> <br>
    <var>a2+b2</var> <br>
    <hr>
    <address>
      ExcelR <br>Ameerpet
      <br> Hyderabad
    </address>
    <br>
    <abbr title="Hyper Text Markup Language">
      HTML
    </abbr>
    <br>
    <bdo dir="rtl">Welcome to UI Technologies</bdo>
    <br>
    <blockquote>
      Stay Home Stay Safe
    </blockquote>
  </body>
</html>
```

Output

This is **Bold** Text
This is **strong** text
This is *Italic* Text
This is *Emphasized* Text
This is underlined Text
This is ~~Deleted~~ Text
This is ~~Striked~~ Text
This is Marked Text
a^2+b^2
H_2O
This is Big Text
This is small Text
`for(int i=0;i<10;i++){}`
a2+b2

Sathya Technologies
Ameerpet
Hyderabad

HTML
seigolonhceT IU ot emocleW

Stay Home Stay Safe

PARAGRAPHS

- ✓ Paragraphs in HTML
- ✓ Inline Elements
- ✓ Converting Block Level Elements to Inline Elements
- ✓ Converting Inline Elements to Block Level Elements
- ✓ Margin in CSS
- ✓ Box Model in CSS

Paragraphs

- ✓ tag, used to display paragraphs
- ✓ tag also block level element

Inline Elements

- ✓ each element never starts with the new line
  (Inline elements displays horizontally)

Converting Block Level Elements to Inline Elements in CSS

- ✓ Below snippet used to convert block level elements to inline elements
  Ex.
  display: inline-block;

Converting Inline Elements to Block Level Elements in CSS

- ✓ Below snippet used to convert inline elements to block level elements
  Ex.
  display: block;

Margin in CSS

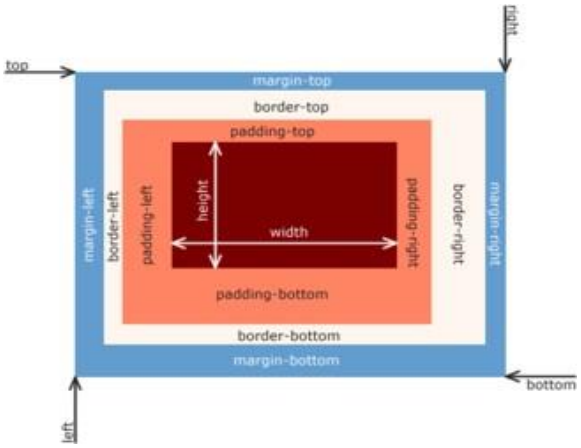
- ✓ space around the element called as margin

Ex.

margin: 20px

box-model in CSS

- ✓ combination of padding, border and margin called as box-model in CSS



Example:

paragrahs.html

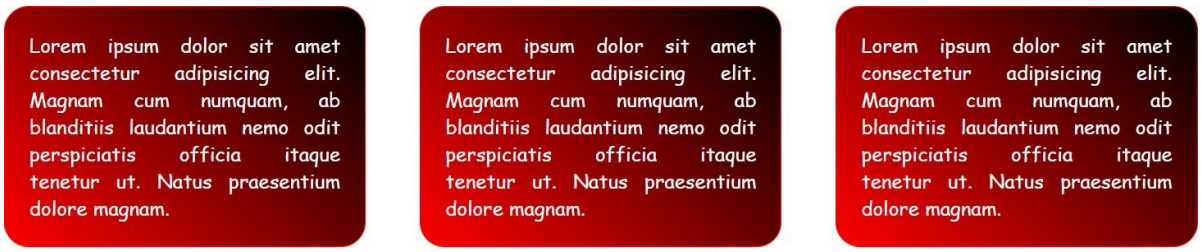
```
<!DOCTYPE html>
<html>
  <head>
    <title>Paragraphs</title>
    <link rel="stylesheet" href="paragraphs.css">
  </head>
  <body>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam cum numquam, ab blanditiis laudantium nemo odit perspiciatis officia itaque tenetur ut. Natus praesentium dolore magnam.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam cum numquam, ab blanditiis laudantium nemo odit perspiciatis officia itaque tenetur ut. Natus praesentium dolore magnam.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam cum numquam, ab blanditiis laudantium nemo odit perspiciatis officia itaque tenetur ut. Natus praesentium dolore magnam.
    </p>
  </body>
```

```
</html>
```

paragraphs.css

```
p{
   border: 1px solid red;
   width: 20%;
   display: inline-block;
   padding: 20px;
   margin: 20px;
   border-radius: 20px;
   text-align: justify;
   font-family: Comic Sans MS;
   color: white;
   background: linear-gradient(45deg,red,black);
}
```

Output



PRACTICE PAPER

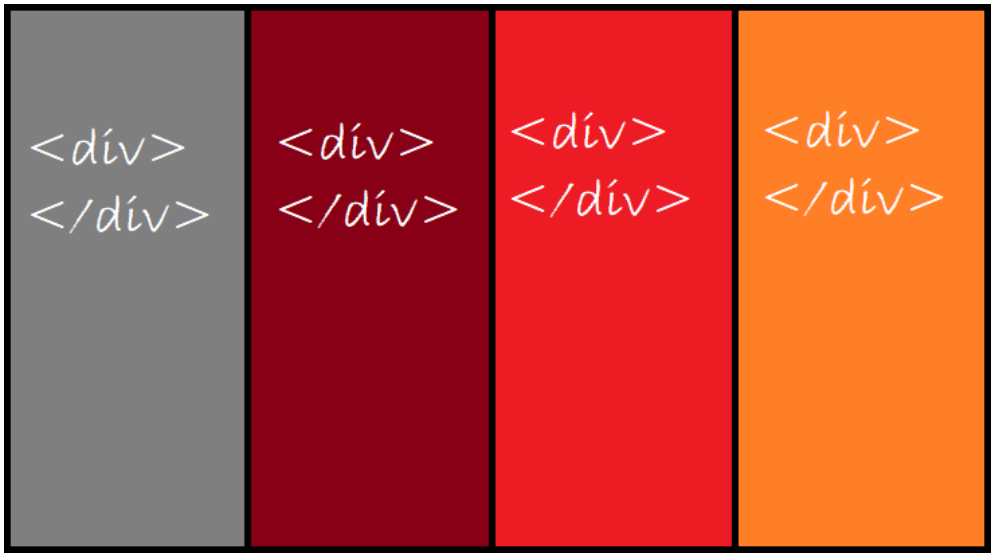
1) which tag is used to create the Paragraphs_____
2) Are Paragraphs Block Level Elements? (yes/no) _____
3) how to convert block to inline _____
4) how to convert inline to block_____
5) complete the below application with paragraphs

DIVISIONS

- ✓ divisions in html
- ✓ internal CSS
- ✓ differences between internal CSS and external CSS
- ✓ selectors
    1. class selector
    2. id selector
    3. element selector
    4. universal selector
- ✓ differences between class selector and id selector

divisions in html

- ✓ Divisions are used to divide webpage into multiple sections
- ✓ tag used to create divisions in webpages
- ✓ tag also block level element



internal CSS

- ✓ <style></style> tag used to write internal CSS

differences between internal CSS and external CSS

Internal CSS	External CSS
<style></style> tag used to write the internal CSS	<link> tag, used to include the external CSS
<style></style> tag is paired tag	<link> tag is non paired tag
we can achieve CSS reusability through external CSS	we can't reuse CSS through internal CSS

selectors

- ✓ Selector "Selects" Particular "HTML Elements" to apply CSS

Types of Selectors

1. class selector
2. id selector
3. element selector / tag selector
4. universal selector

class selector

- ✓ class selector should start with "." (dot)

Syntax

HTML	CSS
*****	***
	.c1{
	}

id selector

- ✓ id selector should start with "#"

Syntax

HTML	CSS
*****	***
	#id1{
	}

element selector

- ✓ element selector starts with "tag" name

Syntax

HTML	CSS
*****	***
	div {
	}

universal selector

- ✓ we will represent "universal selector" with "*"
- ✓ CSS applied to "all elements" through "universal selector"

differences between class selector and id selector

Class selector	**Id selector**
class selector starts with . (dot)	id selector starts with "#"
we may duplicate class	Id should be unique
we can apply more than one class Ex. 	we can apply only one id Ex.
class selector has less priority compared to id selector (specificity)	id selector has more priority compared to class selector (specificity)
if a greater number of teams working on same project, then class selector is suggested	to override 3rd party CSS then we will use id selector

Example
divisions.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Divisions</title>
    <style>
      div{
        border: 1px solid gray;
        width: 25%;
        display: inline-block;
        padding: 10px;
        margin: 10px;
        border-radius: 20px;
      }
      h1{
        color: red;
        text-align: center;
      }
      p{
        text-align: justify;
        font-family: Comic Sans MS;
        color: white;
      }
      .c1{
        background: linear-gradient(90deg,green,yellow);
      }
      #id1{
        background: linear-gradient(90deg,pink,yellow);
      }
      #id2{
        background: linear-gradient(90deg,black,blue);
      }
    </style>
  </head>
  <body>
    <div class="c1">
      <h1>HTML</h1>
      <p>
```

```
        Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aperiam alias esse ducimus, atque quidem exercitationem saepe. Maiores fugit praesentium mollitia iste magnam sunt earum facilis?
      </p>
    </div>

    <div id="id1">
      <h1>CSS</h1>
      <p>
        Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aperiam alias esse ducimus, atque quidem exercitationem saepe. Maiores fugit praesentium mollitia iste magnam sunt earum facilis?
      </p>
    </div>
    <div id="id2">
      <h1>JAVASCRIPT</h1>
      <p>
        Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aperiam alias esse ducimus, atque quidem exercitationem saepe. Maiores fugit praesentium mollitia iste magnam sunt earum facilis?
      </p>
    </div>
  </body>
</html>
```

Output

PRACTICE PAPER

1) How to apply internal styles to the web pages _____
2) write the Differences between external CSS and Internal CSS?

3) how to create Divisions in HTML _____
4) Explain Selectors in CSS?

5) Class selector should start with _____
6) ID Selector should start with _____
7) Differences Between Class selector and ID Selector?

8) How to create universal selector _____

9) Write the code for below application

Buttons, span, images and Inline CSS

- ✓ **Button**
- ✓ **applying css to webpages**
- ✓ **CSS Specificity (Priority)**
- ✓ **Span**
- ✓ **Images**
- ✓ **Hover in CSS**

Button

- ✓ <button></button> tag used to display Buttons in webpages
- ✓ <button></button> tag is inline element

Ex.

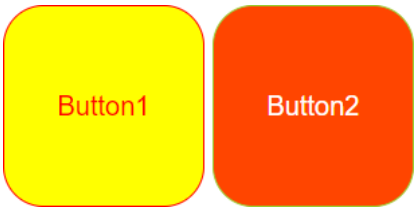
```
<button>
        Login
</button>
```

Example

buttons.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Buttons</title>
  </head>
  <body>
    <button style="width: 100px;
            height: 100px;
            background-color: yellow;
            color: red;
            border: 1px solid red;
            border-radius: 20px;">
      Button1
    </button>
    <button style="width: 100px;
            height: 100px;
            background-color: orangered;
            color: white;
            border: 1px solid yellowgreen;
            border-radius: 20px;">
      Button2
    </button>
  </body>
</html>
```

Output:



applying css to webpages

- ✓ we can apply css to webpages in three ways
  1) external css
  2) internal css
  3) inline css

external css

<link> tag used to include the external css
through external css we can achieve css reusability

internal css

<style></style> tag used to write the internal css
through internal css we can't reuse the css

inline css

style attribute used to apply the inline css
through "inline css" also we can't reuse the css

CSS Specificity (Priority)

CSS priorities are defined below like this

Ex.

inline > id > class > element

span

- ✓ wrap particular "content" from "text", we will use span
- ✓ tag is inline element
- ✓ tag used to display validation messages

Example

span.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>SPAN</title>
  </head>
  <body>
    <p>
      welcome to <span style="color: red;">ui</span> technologies
    </p>
  </body>
```

```
</html>
```

Output

welcome to ui technologies

Images

- ✓ <img> tag used to display the images
- ✓ <img> tag is non paired tag
- ✓ <img> tag is inline element

Attributes

1) width
   used to define image width
2) height
   used to define image height
3) src
   refers image page
4) alt
   whenever image displays fail automatically alternative text will display

Example

Images.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Images</title>
    <style>
      img{
        margin: 20px;
        border: 2px solid blue;
      }
      .c1:hover{
        width: 200px;
        height: 150px;
        border: 2px solid red;
      }
      .c2:hover{
        height: 150px;
        border-radius: 100%;
        border-color: green;
```

```
        }
        .c3:hover{
            display: none;
        }
    </style>
  </head>
  <body>
    <img width="150px" height="100px" src="flipkart.png" alt="Error"
class="c1">
    <img width="150px" height="100px" src="amazon.png" alt="Error"
class="c2">
    <img width="150px" height="100px" src="myntra.png" alt="Error"
class="c3">
  </body>
</html>
```

Output:

Hover

- ✓ Hover is a pseudo class in css
- ✓ Whenever mouse over a particular element automatically hover effect will execute

Syntax

```
.c1: hover {
    //css code
}

#id1: hover {
     //css code
}

Img: hover {
    //css code
}
```

PRACTICE PAPER

1) How to Create Button and Write Basic Example
   Ans:

2) Explain tag in HTML?
   Ans:

3) Are Button Span and img tags are inline (yes/no) _____
4) How to write inline CSS _____
5) In How Many ways we can apply CSS?
   Ans:

6) Explain CSS Specificity?
   Ans:

7) How to display images in HTML _____
8) Is <img> tag paired tag?_____
9) Write <img> tag attributes
   Ans:

10) Explain hover in CSS?
Ans:

11)write the code for below application

- ✓ **Hyperlinks**
- ✓ **Fieldset**
- ✓ **Marquee**

Hyperlinks

- ✓ <a> tag used to create the hyper links
- ✓ <a> tag used to navigate from one web page to another Webpage
- ✓ <a> tag also used to send emails
- ✓ <a> tag also used to open the external websites
- ✓ <a> tag is inline element

Attributes

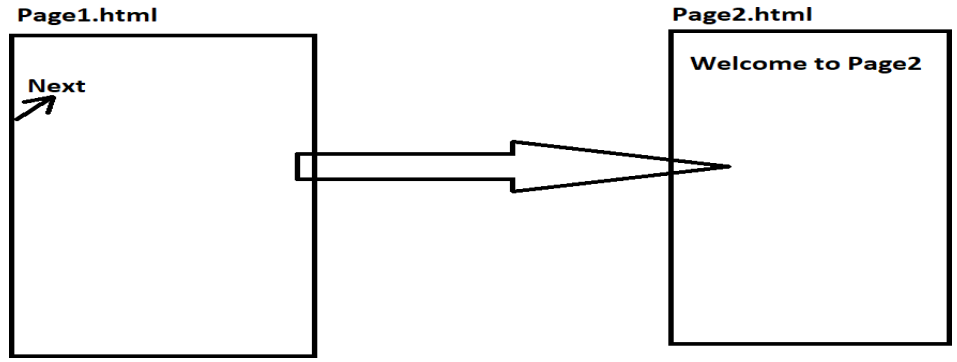
1) href

used to link the path of "target resource"

2) target="_blank"

used to open the "new tab" in browser

Example1:



Code

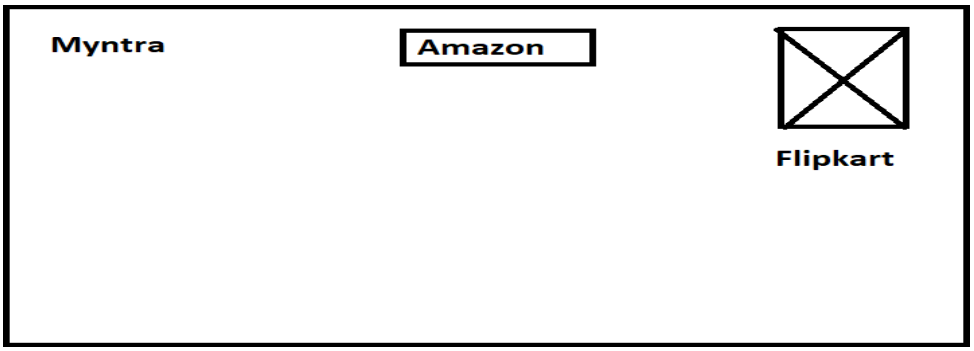
page1.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Anchor</title>
  </head>
  <body>
    <a href="page2.html" target="_blank">
      Next
    </a>
  </body>
</html>
```

page2.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Page2</title>
   </head>
   <body>
      <h1>Welcome to Page2... !</h1>
   </body>
</html>
```

Example2:



Example

```
<!DOCTYPE html>
<html>
   <head>
      <title>Websites Example</title>
   </head>
   <body>
      <a href="https://www.myntra.com/" target="_blank"
         style="margin: 40px;">
         Myntra
      </a>
      <a href="https://www.amazon.in/" target="_blank"
         style="margin: 20px;">
         <button>Amazon</button>
      </a>
      <a href="https://www.flipkart.com/" target="_blank">
         <img width="50px" height="50px" src="flipkart.png" alt="Error">
      </a>
      </body>
</html>
```

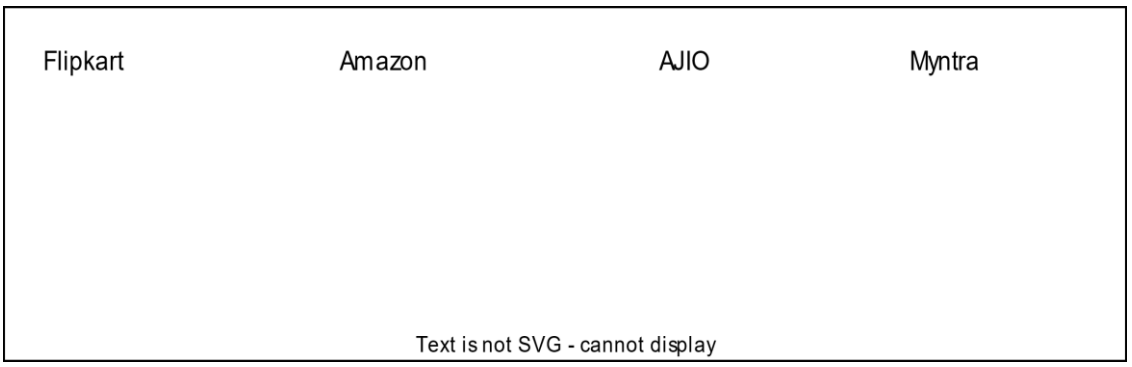
PRACTICE PAPER

1) How to create Hyper Links in HTML?
Ans:

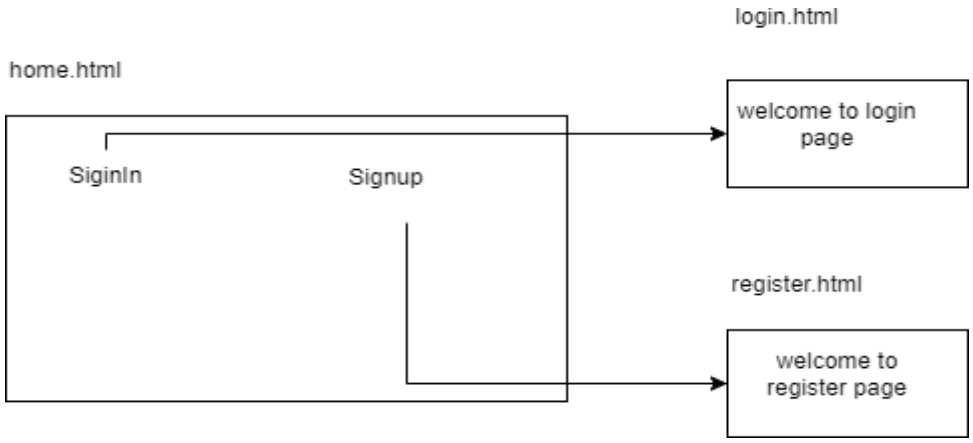
2) Write Attributes of Anchor Tag?
Ans:

3) implement the below application with anchor tag

Flipkart -> https://www.flipkart.com/
Amazon -> https://www.amazon.in/
AJIO -> https://www.ajio.com/
Myntra -> https://www.myntra.com/
Load above web sites in new tab



4) implement the below application with anchor tag



Fieldset

- ✓ this tag used to create rectangular structures

Ex.

```
<fieldset>
        <legend>LOGIN FORM</legend>
            ---
            ---
```

```
    </fieldset>
```

Example:

fieldset.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Fieldset</title>
    <style>
      fieldset{
        width: 25%;
        border-color: red;
        border-radius: 20px;
      }
      legend{
        background-color: yellow;
        color: red;
        padding:  5px;
        border: 2px solid red;
        border-radius: 20px;
      }
      p{
        text-align: justify;
        font-family: Comic Sans MS;
        color: red;
      }
    </style>
  </head>
  <body>
    <fieldset>
      <legend>LOGIN FORM</legend>
      <p>
        Lorem, ipsum dolor sit amet consectetur adipisicing elit. Possimus, quaerat tenetur debitis libero et sequi doloremque ratione aperiam fugiat cum. Praesentium, fugiat. Voluptates, deleniti. Omnis!
      </p>
    </fieldset>
  </body>
</html>
```

Output:

PRACTICE PAPER

1) Design the Below Application with fieldset and legend

Marquee

- ✓ it is used to scroll the content
- ✓ <marquee></marquee> used to create the marquee elements

Attributes

1) direction="left"/"right"/"up"/"down"
2) scollamount

Example

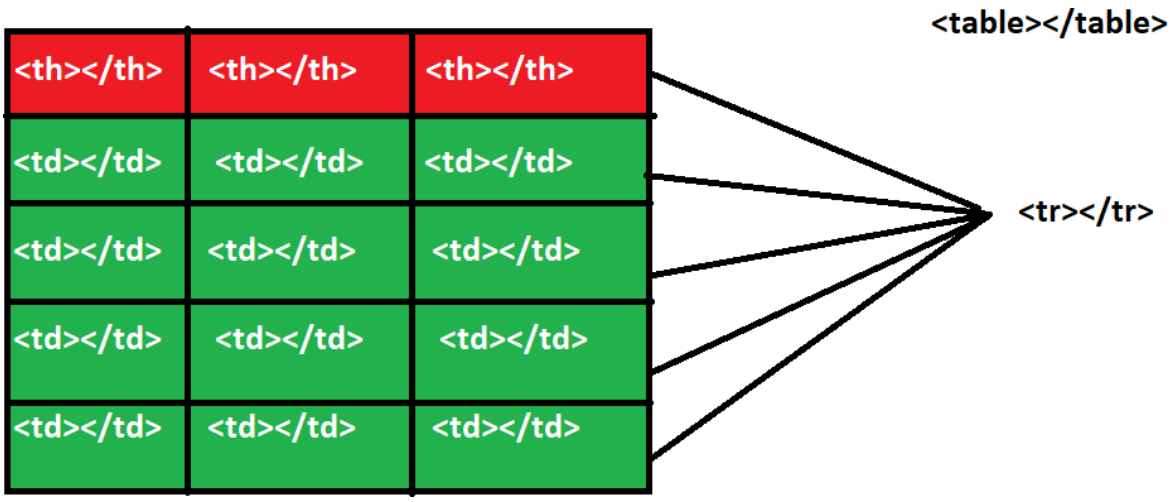
marquee.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Marquee</title>
  </head>
  <body>
    <marquee scrollamount="100">
      <h1>Bye Bye !!!</h1>
    </marquee>

    <marquee>
```

```
    <img src="flipkart.png">
  </marquee>

  <marquee>
    <h1>HTML</h1>
  </marquee>

  <marquee direction="right">
    <h1>CSS</h1>
  </marquee>

  <marquee direction="up">
    <h1>JAVASCRIPT</h1>
  </marquee>

  <marquee direction="down">
    <h1>REACTJS</h1>
  </marquee>

 </body>
</html>
```

TABLES

- ✓ collection of rows and columns called as table



TAGS

- ✓ <table></table> tag used to create the tables
- ✓ <tr></tr> tag used to create table row
- ✓ <th></th> tag used to display table heading
- ✓ <td></td> tag used to display cell data

Attributes

1) border

it is used to draw the border to the table

Ex.

border="1"

2) align="left"/"center"/"right"

it is used to align the table

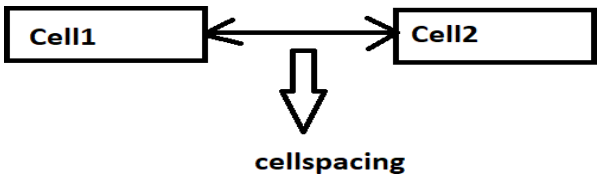
default value is "left"

3) cellspacing

it is used to maintain space between cells

Ex.

cellspacing="10px"

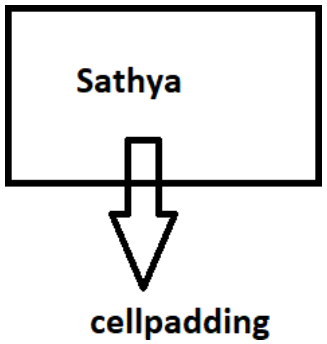


4) cellpadding

space around cell content called as cellpadding

Ex.

cellpadding="10"



5) colspan

it is used to merge the cells horizontally

Ex.

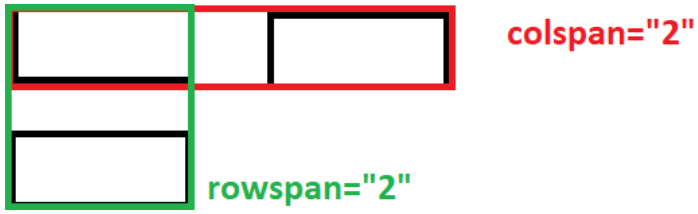
colspan="2"

6) rowspan

it is used to merge the cells vertically

Ex.

rowspan="2"



Examples

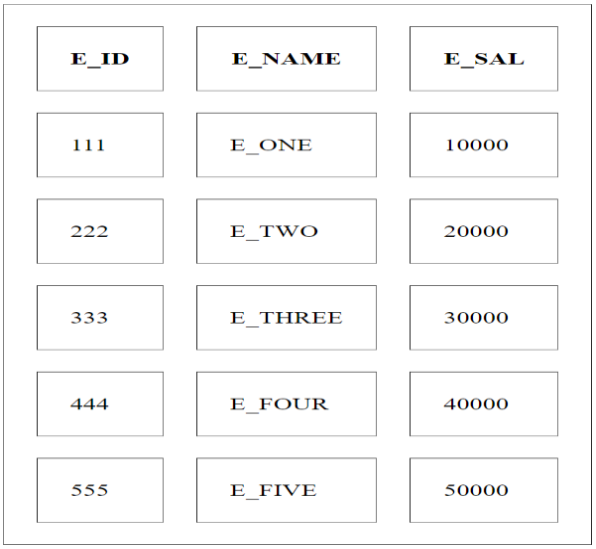
****

table1.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tables</title>
    </head>
    <body>
        <table border="1"
               align="center"
               cellspacing="20px"
               cellpadding="20px">
```

```
            <tr>
                <th>E_ID</th>
                <th>E_NAME</th>
                <th>E_SAL</th>
            </tr>

            <tr>
                <td>111</td>
                <td>E_ONE</td>
                <td>10000</td>
            </tr>

            <tr>
                <td>222</td>
                <td>E_TWO</td>
                <td>20000</td>
            </tr>

            <tr>
                <td>333</td>
                <td>E_THREE</td>
                <td>30000</td>
            </tr>

            <tr>
                <td>444</td>
                <td>E_FOUR</td>
                <td>40000</td>
            </tr>

            <tr>
                <td>555</td>
                <td>E_FIVE</td>
                <td>50000</td>
            </tr>
        </table>
    </body>
</html>
```

Output:



E_ID	E_NAME	E_SAL
111	E_ONE	10000
222	E_TWO	20000
333	E_THREE	30000
444	E_FOUR	40000
555	E_FIVE	50000

table2.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>TABLES</title>
    </head>
    <body>
        <table border="1"
               align="center"
               cellspacing="10px"
               cellpadding="10px">
            <tr>
                <th>P_ID</th>
                <th>P_NAME</th>
                <th>P_COST</th>
                <th>P_IMAGE</th>
                <th>UPDATE</th>
                <th>DELETE</th>
            </tr>

            <tr>
                <td>111</td>
                <td>P_ONE</td>
                <td>10000</td>
                <td>
                    <img src="flipkart.png" width="100px"
height="100px" alt="Error">
                </td>
```

```
                <td>
                    <button>UPDATE</button>
                </td>
                <td>
                    <button>DELETE</button>
                </td>
            </tr>

            <tr>
                <td>222</td>
                <td>P_TWO</td>
                <td>20000</td>
                <td>
                    <img width="100px" height="100px"
src="amazon.png" alt="Error">
                </td>
                <td>
                    <button>UPDATE</button>
                </td>
                <td>
                    <button>DELETE</button>
                </td>
            </tr>

            <tr>
                <td>333</td>
                <td>P_THREE</td>
                <td>30000</td>
                <td>
                    <img width="100px" height="100px"
src="myntra.png" alt="Error">
                </td>
                <td>
                    <button>UPDATE</button>
                </td>
                <td>
                    <button>DELETE</button>
                </td>
            </tr>
        </table>
    </body>
```

```
</html>
```

Output:

P_ID	P_NAME	P_COST	P_IMAGE	UPDATE	DELETE
111	P_ONE	10000		UPDATE	DELETE
222	P_TWO	20000		UPDATE	DELETE
333	P_THREE	30000		UPDATE	DELETE

tables3.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>TABLES</title>
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    </head>
    <body>
        <table border="1"
               align="center"
               cellspacing="10px"
               cellpadding="10px">
            <tr>
                <th>NAME</th>
                <th>DELETE</th>
                <th>UPDATE</th>
```

```
            </tr>
            <tr>
                <td>LAPTOP</td>
                <td>
                    <i class="fa fa-trash"></i>
                </td>
                <td>
                    <i class="fa fa-edit"></i>
                </td>
            </tr>
            <tr>
                <td>WATCH</td>
                <td>
                    <i class="fa fa-trash"></i>
                </td>
                <td>
                    <i class="fa fa-edit"></i>
                </td>
            </tr>
            <tr>
                <td>
                    MOBILE
                </td>
                <td>
                    <i class="fa fa-trash"></i>
                </td>
                <td>
                    <i class="fa fa-edit"></i>
                </td>
            </tr>
        </table>
    </body>
</html>
```

Output

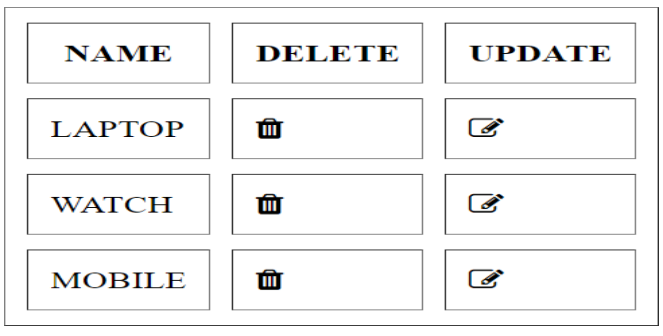
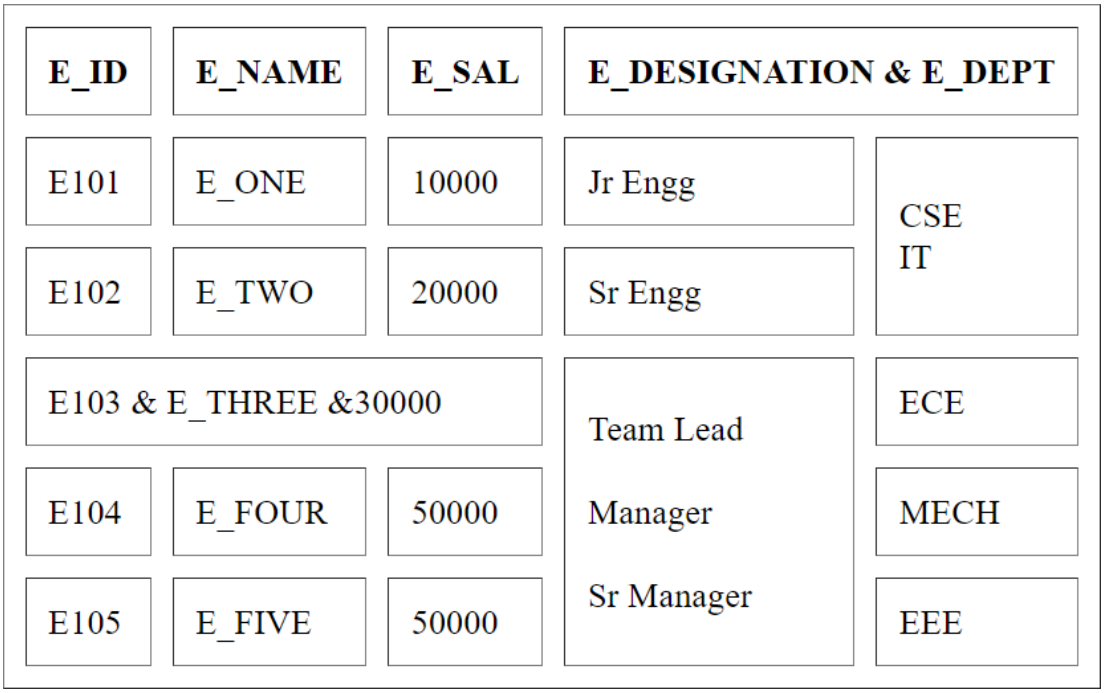


table4.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>TABLES</title>
    </head>
    <body>
        <table border="1"
               align="center"
               cellspacing="10px"
               cellpadding="10px">
            <tr>
                <th>E_ID</th>
                <th>E_NAME</th>
                <th>E_SAL</th>
                <th colspan="2">E_DESIGNATION &
E_DEPT</th>
            </tr>
            <tr>
                <td>E101</td>
                <td>E_ONE</td>
                <td>10000</td>
                <td>Jr Engg</td>
                <td rowspan="2">CSE <br> IT</td>
            </tr>
            <tr>
                <td>E102</td>
                <td>E_TWO</td>
                <td>20000</td>
                <td>Sr Engg</td>
            </tr>
            <tr>
                <td colspan="3">E103 & E_THREE &30000</td>
                <td rowspan="3">Team Lead <br><br> Manager
<br><br> Sr Manager</td>
                <td>ECE</td>
            </tr>
            <tr>
                <td>E104</td>
                <td>E_FOUR</td>
                <td>50000</td>
```

```
                <td>MECH</td>
            </tr>
            <tr>
                <td>E105</td>
                <td>E_FIVE</td>
                <td>50000</td>
                <td>EEE</td>
            </tr>
        </table>
    </body>
</html>
```

Output



E_ID	E_NAME	E_SAL	E_DESIGNATION & E_DEPT	
E101	E_ONE	10000	Jr Engg	CSE IT
E102	E_TWO	20000	Sr Engg	
E103 & E_THREE &30000			Team Lead	ECE
E104	E_FOUR	50000	Manager	MECH
E105	E_FIVE	50000	Sr Manager	EEE

tables5.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>TABLES WITH CSS</title>
        <style>
            table{
                width: 100%;
                text-align: center;
                border: 1px solid gray;
                border-collapse: collapse;
            }
```

```
            th,td{
                border: 1px solid gray;
            }
            th{
                height: 50px;
            }
            td{
                height: 30px;
            }
            tr:nth-child(even){
                background-color: yellow;
                color: red;
            }
            tr:nth-child(odd){
                background-color: orange;
                color: white;
            }
            tr:hover{
                background-color: burlywood;
            }
        </style>
    </head>
    <body>
        <table>
            <tr>
                <th>SNO</th>
                <th>SNAME</th>
                <th>CLASS</th>
            </tr>
            <tr>
                <td>1</td>
                <td>SONE</td>
                <td>DBatch</td>
            </tr>
            <tr>
                <td>2</td>
                <td>STWO</td>
                <td>CBatch</td>
            </tr>
            <tr>
                <td>3</td>
```

```
                <td>STHREE</td>
                <td>BBatch</td>
            </tr>
            <tr>
                <td>4</td>
                <td>SFOUR</td>
                <td>ABatch</td>
            </tr>
        </table>
    </body>
</html>
```

Output

SNO	SNAME	CLASS
1	SONE	DBatch
2	STWO	CBatch
3	STHREE	BBatch
4	SFOUR	ABatch

PRACTICE PAPER

1) which tag used to create tables _____
2) write the tags to create tables with diagram
   Ans

3) write table attributes with explanation
   Ans

4) write the code for below diagram in practice paper and execute in laptop

SNO	P_ID	P_NAME	P_COST
1	P101	P_ONE	10000
2	P102	P_TWO	20000
3	P103	P_THREE	30000
4	P104	P_FOUR	40000
5	P105	P_FIVE	50000

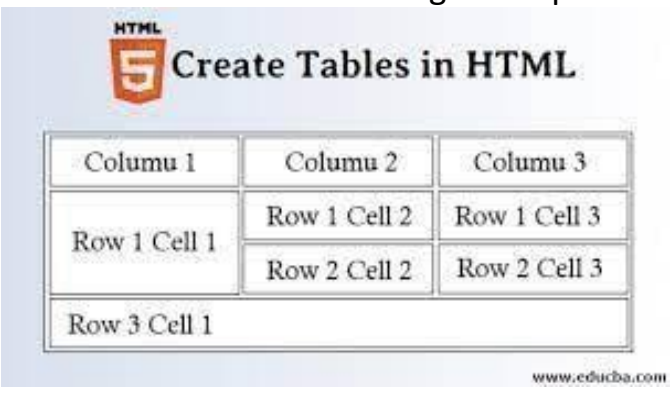
5) write the code for below diagram in practice paper and execute in laptop

COLLEGE TIME TABLE

	8:30-9:30	9:30-10:30	10:30-11:30	11:30-12:30	12:30-2:00	2:00-3:00	3:00-4:00	4:00-5:00
MONDAY	---	SUB1	SUB2	SUB3		SUB4	SUB5	COUNSELLING CLASS
TUESDAY	SUB1	SUB2	SUB3	---		SUB2	SUB2	LIBRARY
WEDNESDAY	SUB1	SUB2	SWA	---		LAB		
THURSDAY	SUB1	SUB2	SUB3	---		SUB2	SUB2	LIBRARY
FRIDAY	SUB1	SUB2	SUB3	---		SUB4	SUB5	LIBRARY
SATURDAY	SUB1	SEMINAR				SUB4	SUB5	LIBRARY

LUNCH

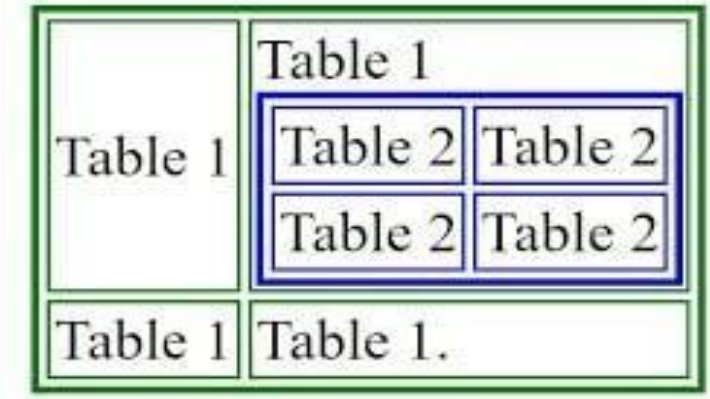
6) write the code for below diagram in practice paper and execute in laptop



7) write the code for below diagram in practice paper and execute in laptop

Day	Seminar		
	Schedule		Topic
	Begin	End	
Monday	8:00 a.m.	5:00 p.m.	Introduction to XML
			Validity: DTD and Relax NG
Tuesday	8:00 a.m.	11:00 a.m.	XPath
	11:00 a.m.	2:00 p.m.	
	2:00 p.m.	5:00 p.m.	XSL Transformations
Wednesday	8:00 a.m.	12:00 p.m.	XSL Formatting Objects

8) write the code for below diagram in practice paper and execute in laptop



9) write the code for below diagram in practice paper and execute in laptop

10)write the code for below diagram in practice paper and execute in laptop

SNO	NAME	COST	UPDATE	DELETE
1	LAPTOP	₹50000.00		
2	Mobile	$50		
3	Shoes	€10		

11)write few points related to font-awesome

12) cart symbol _____
13) trash symbol _____
14) edit symbol _____
15) rupee symbol_____
16) dollar symbol _____
17) user profile _____
18) how to use font-awesome_____
19) CDN Stands for _____
20) is Font Awesome CSS library? _____

LISTS

- ✓ **Lists**
  1. **Ordered List**
  2. **Unordered List**
  3. **Definition List**
- ✓ **Nested Lists**
- ✓ **Practice Paper**

Lists

- ✓ HTML Supports three types of lists

  1) ordered list
  2) unordered list
  3) definition list

ordered list

- ✓ is the tag used to create "ordered list"
- ✓ is the tag used to create "list item"

Syntax

```
<ol>
    <li></li>
      ---
      ---
      ---
</ol>
```

Attributes

1) type="A"/"a"/"I"/"i"/1
   default is the number
2) start="5"
   list starts with specified number
3) reversed
   it is used to reverse the list item numbers

Example

lists1.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ordered List</title>
    <link rel="stylesheet" href="lists.css">
  </head>
  <body>
    <ol>
```

```
    <li>HTML</li>
    <li>CSS</li>
    <li>JAVASCRIPT</li>
    <li>REACTJS</li>
    <li>ANGULAR</li>
</ol>

<ol type="A">
    <li>HTML</li>
    <li>CSS</li>
    <li>JAVASCRIPT</li>
    <li>REACTJS</li>
    <li>ANGULAR</li>
</ol>

<ol type="a">
    <li>HTML</li>
    <li>CSS</li>
    <li>JAVASCRIPT</li>
    <li>REACTJS</li>
    <li>ANGULAR</li>
</ol>

<ol type="I">
    <li>HTML</li>
    <li>CSS</li>
    <li>JAVASCRIPT</li>
    <li>REACTJS</li>
    <li>ANGULAR</li>
</ol>

<ol type="i">
    <li>HTML</li>
    <li>CSS</li>
    <li>JAVASCRIPT</li>
    <li>REACTJS</li>
    <li>ANGULAR</li>
</ol>

<ol start="5" reversed>
```

```
        <li>HTML</li>
        <li>CSS</li>
        <li>JAVASCRIPT</li>
        <li>REACTJS</li>
        <li>ANGULAR</li>
      </ol>
   </body>
</html>
```

lists.css

```
ol, ul {
   background-color: red;
   width: 15%;
   display: inline-block;
   margin: 20px;
   border: 2px solid black;
   border-radius: 20px;
}
li {
   color: white;
   background-color: black;
   margin: 10px;
   padding: 10px;
   border: 2px solid white;
   border-radius: 20px;
}
```

Output

unordered list

- ✓ is the tag used to create **unordered** list

Syntax

```
<ul>
    <li></li>
        ---
        ---
        ---
</ul>
```

Attributes

type="disc"/"square"/"circle"

default symbol is "**disc**"

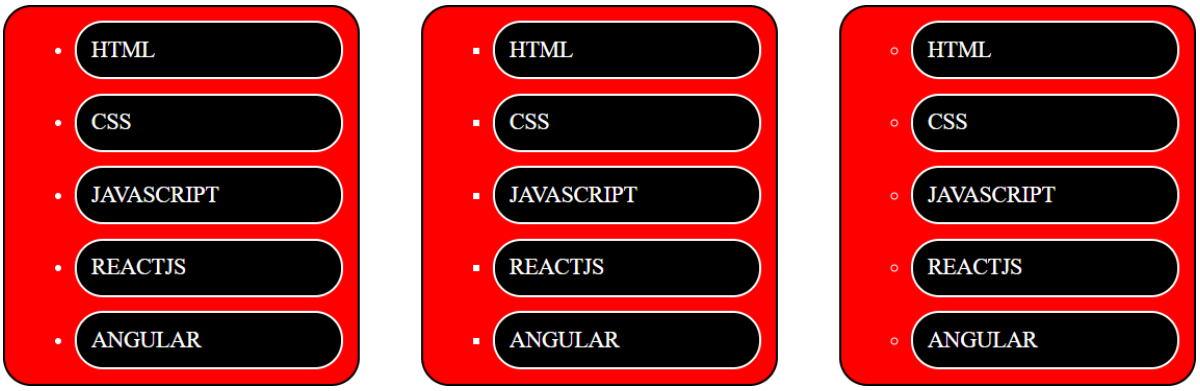
Example

lists2.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Unordered List</title>
    <link rel="stylesheet" href="lists.css">
  </head>
  <body>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JAVASCRIPT</li>
      <li>REACTJS</li>
      <li>ANGULAR</li>
    </ul>

    <ul type="square">
      <li>HTML</li>
      <li>CSS</li>
      <li>JAVASCRIPT</li>
      <li>REACTJS</li>
      <li>ANGULAR</li>
    </ul>

    <ul type="circle">
      <li>HTML</li>
      <li>CSS</li>
```

```
        <li>JAVASCRIPT</li>
        <li>REACTJS</li>
        <li>ANGULAR</li>
      </ul>
   </body>
</html>
```

Output



definition list

- ✓ <dl></dl> used to create the definition list
- ✓ <dt></dt> stands for definition term
- ✓ <dd></dd> stands for definition description

Syntax

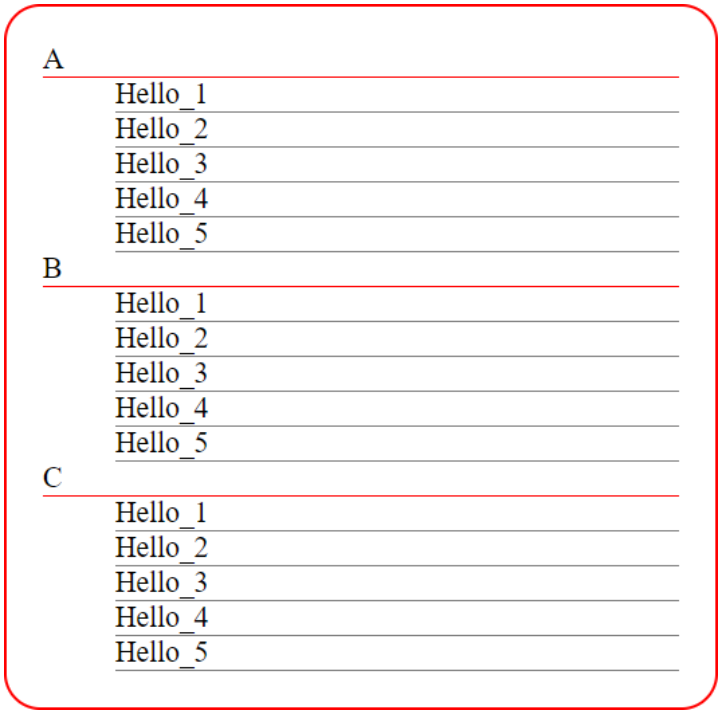
```
        <dl>
                <dt></dt>
                <dd></dd>
                <dd></dd>
                   ---
                   ---

                <dt></dt>
                <dd></dd>
                <dd></dd>
                   ---
                   ---
        </dl>
```

lists3.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Definition List</title>
      <style>
```

```
        dl {
          border: 2px solid red;
          width: 25%;
          padding: 20px;
          border-radius: 20px;
        }
        dt {
          border-bottom: 1px solid red;
        }
        dd {
          border-bottom: 1px solid gray;
        }
    </style>
  </head>
  <body>
    <dl>
      <dt>A</dt>
      <dd>Hello_1</dd>
      <dd>Hello_2</dd>
      <dd>Hello_3</dd>
      <dd>Hello_4</dd>
      <dd>Hello_5</dd>

      <dt>B</dt>
      <dd>Hello_1</dd>
      <dd>Hello_2</dd>
      <dd>Hello_3</dd>
      <dd>Hello_4</dd>
      <dd>Hello_5</dd>

      <dt>C</dt>
      <dd>Hello_1</dd>
      <dd>Hello_2</dd>
      <dd>Hello_3</dd>
      <dd>Hello_4</dd>
      <dd>Hello_5</dd>
    </dl>
  </body>
</html>
```

Output



Nested Lists

- ✓ One list in another list called as **Nested List**

Example

lists4.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>NESTED LISTS</title>
  </head>
  <body>
    <ol>
      <li>
        what is HTML?
        <ul type="circle">
          <li>Hyper Text Markup Language</li>
          <li>Cascading Style Sheet</li>
          <li>Scripting Language</li>
          <li>All of the Above</li>
        </ul>
      </li>

      <li>
        what is CSS?
```

```
            <ul type="circle">
              <li>Hyper Text Markup Language</li>
              <li>Cascading Style Sheet</li>
              <li>Scripting Language</li>
              <li>All of the Above</li>
            </ul>
          </li>

          <li>
            what is JavaScript?
            <ul type="circle">
              <li>Hyper Text Markup Language</li>
              <li>Cascading Style Sheet</li>
              <li>Scripting Language</li>
              <li>All of the Above</li>
            </ul>
          </li>
        </ol>
    </body>
</html>
```

Output

1. what is HTML?
    - Hyper Text Markup Language
    - Cascading Style Sheet
    - Scripting Language
    - All of the Above
2. what is CSS?
    - Hyper Text Markup Language
    - Cascading Style Sheet
    - Scripting Language
    - All of the Above
3. what is JavaScript?
    - Hyper Text Markup Language
    - Cascading Style Sheet
    - Scripting Language
    - All of the Above

Practice Paper

1) Types of Lists in HTML?
   Ans:

2) Write the Syntax for Ordered List?
   Ans:

3) Write the Attributes of Ordered List?
   Ans:

4) Write the Syntax for Unordered List?
   Ans:

5) Write the Unordered List Attributes?
   Ans:

6) Write the Syntax for Definition List
   Ans:

7) <dt></dt> stands for_____
8) <dd></dd> stands for_____
9) stands for_____
10) <dl></dl> stands for_____
11) Write the code for below application and execute in laptop

12) Write the code for below application and execute in laptop

Quiz

1. HTML is an ——— ?

i. Markup language
ii. Programming language
iii. none of these

2. CSS is used for

i. Styling
ii. scripting
iii. none of these

3.Which of the following is dynamic from of HTML ?

i. XML
ii. DHTML
iii. none of these

4. Which of the following can be linked with HTML and CSS ?

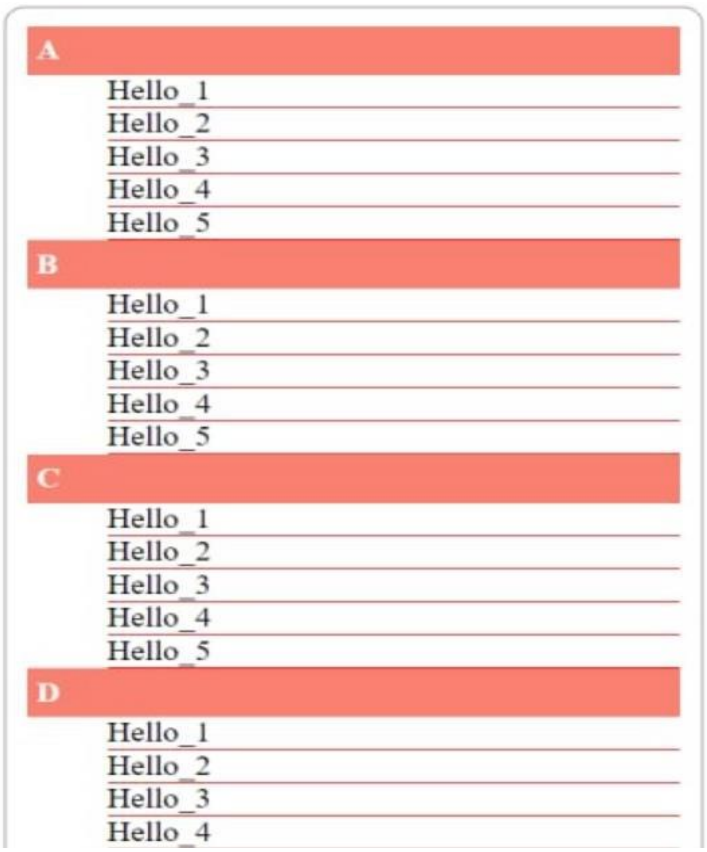
i. javascript
ii. C++
iii. none of these

13) Write the code for below application and execute in laptop

Preceding Text

I. List Item 1
   a. Nested Item 1.1
   b. Nested Item 1.2
II. List Item 2
   1. Nested Item 2.1
   2. Nested Item 2.2
      - Nested Item 2.2.1
      - Nested Item 2.2.2
        - Nested Item 2.2.2.1
        - Nested Item 2.2.2.2
      - Nested Item 2.2.3
   3. Nested Item 2.3
III. List Item 3
   - Nested Item 3.1
   - Nested Item 3.1
   - Nested Item 3.1

14) Write the code for below application and execute in laptop



FORMS

- ✓ **Introduction**
- ✓ **Form Controls**
- ✓ **Radio Buttons**
- ✓ **Check Boxes**
- ✓ **Submit & Reset Button**
- ✓ **Address**
- ✓ **Dropdown**
- ✓ **Practice Paper**

Introduction

- ✓ collection of related data called as Form
  Ex
  LoginForm
  RegistrationForm
  EnrollmentForm

- ✓ <form action="" method="" name="" onsubmit="m1()"></form> tag used to create the Form

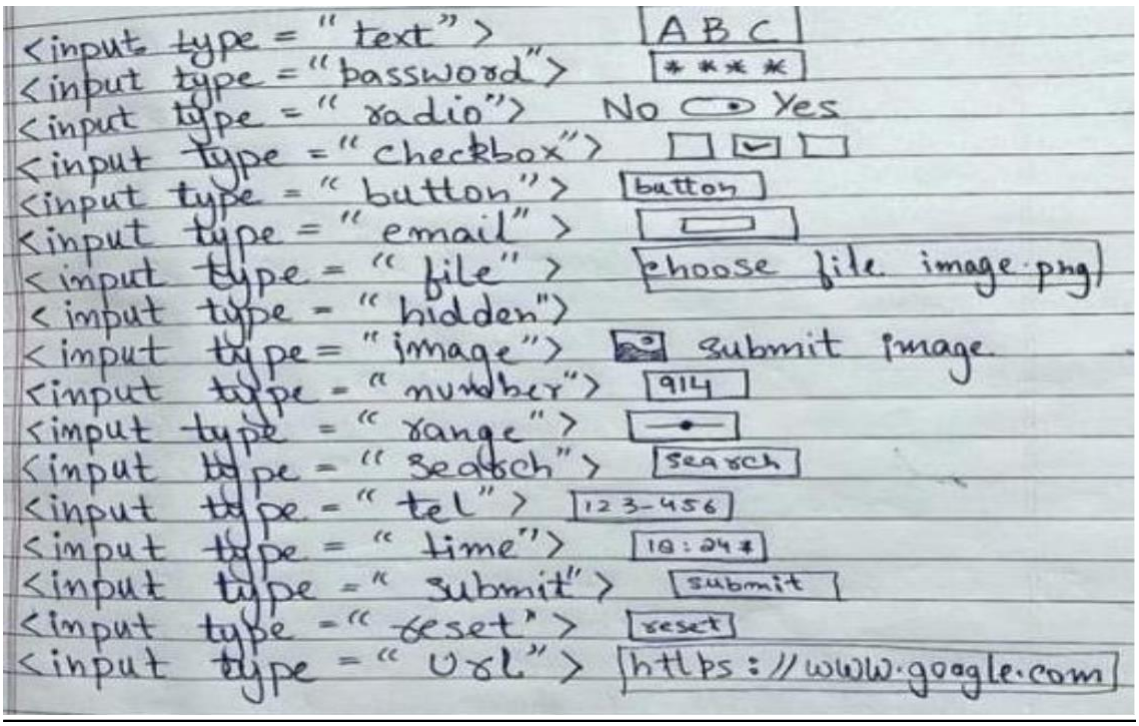
- ✓ action attribute used to connect to backend
- ✓ reading data from backend called as GET Request
- ✓ adding data to backend called as POST Request
- ✓ updating backend data called as PUT Request
- ✓ deleting backend data called as DELETE Request

- ✓ name attribute used to assign logical name to the form

- ✓ onsubmit attribute used to handle the submit event

- ✓ whenever we click submit button automatically m1() function will execute in javascript environment

Form Controls



Radio Buttons

- ✓ Radio Buttons also called as single selection control

male female

```
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="female">Female
```

should be same

Check Boxes

- ✓ multi selection control called as checkboxes

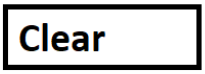
C C++ Java dotnet

```
<input type="checkbox" name="skills" value="c">C
<input type="checkbox" name="skills" value="c++">C++
<input type="checkbox" name="skills" value="java">Java
<input type="checkbox" name="skills" value="dotnet">DotNet
```

should be same

submit & reset buttons


```
<input type="submit" value="Register">
```




```
<input type="reset" value="clear">
```

Address

- ✓ <textarea cols="" rows="" name=""></textarea> tag used to create the address in forms
- ✓ cols attribute used to increse/decrease width of address field in form
- ✓ rows attribute used to increse/decrease height of address field in form

```
<textarea cols="10"   rows="10" name="useraddress"></textarea>
```

dropdown

- ✓ <select name=""></select> used to create the dropdowns
- ✓ <option></option> tag used to populate options to dropdown


```
<select name="countries">
   <option value="">select country</option>
   <option value="ind">India</option>
   <option value="pak">Pakistan</option>
   <option value="chn">China</option>
   <option value="ukr">Ukraine</option>
   <option value="sri">Srilanka</option>
</select>
```

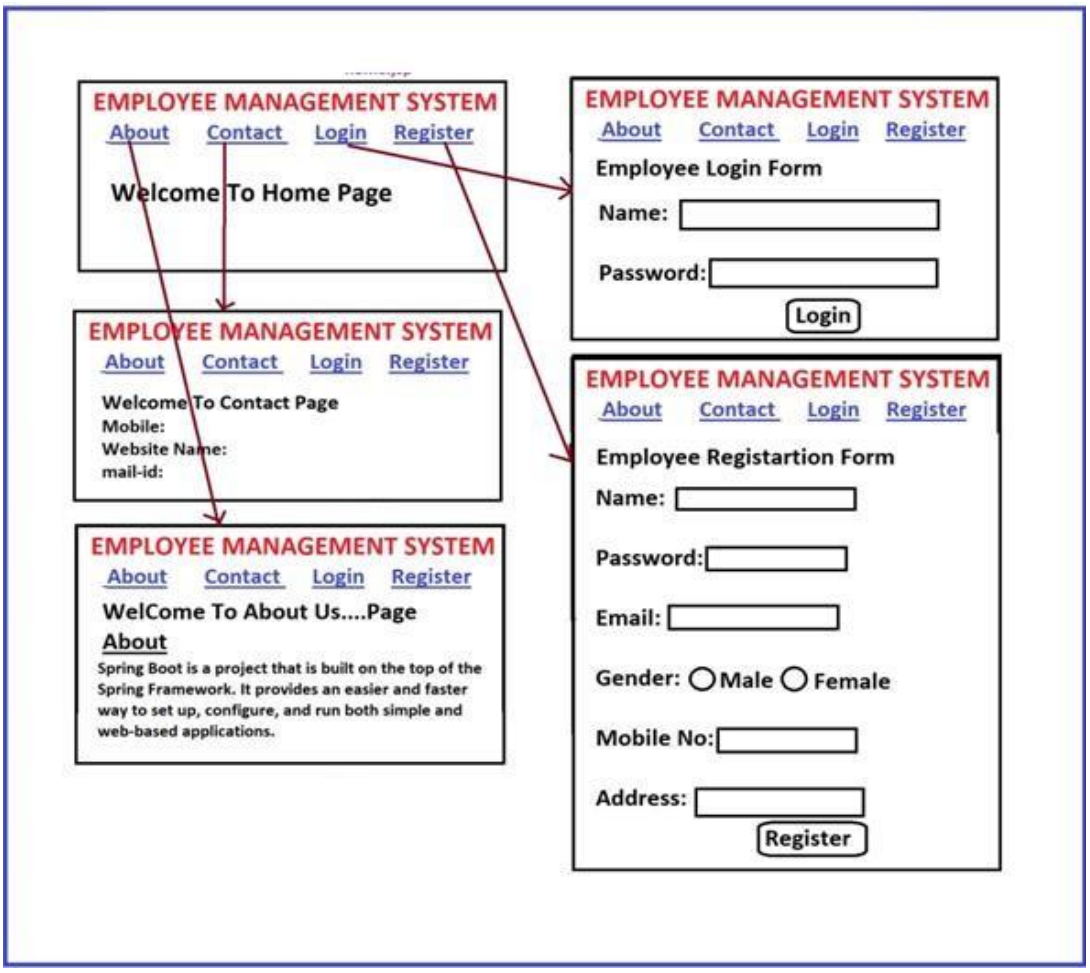
PRACTICE PAPER

1) How to create Forms in HTML _____
2) Write form attributes

3) Why placeholder attribute in <input></input> tag_____
4) How to create dropdown in Forms write basic example

5) How to create address field in HTML Forms

6) Single selection controls called as _____
7) Multi selection controls called as _____
8) implement the below form with text field



9) implement the below form

Fill below form to register

Enter your Id:
Enter your name:
Enter your Password:
Register

10) implement the below form

Customer Login Page

Enter your name:	
Enter your Password:	
Login	

11) Write HTML code to get following text box?

User Name User Name

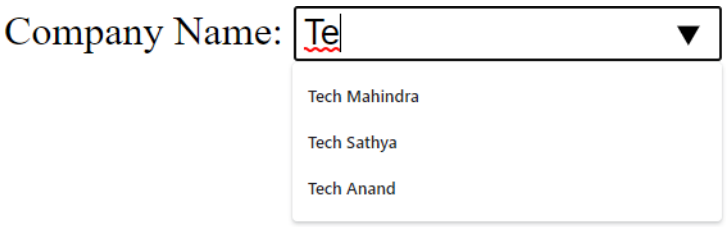
12) Write HTML Code for following output

User Name

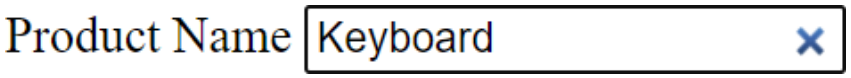
13) Write code for following output

Password

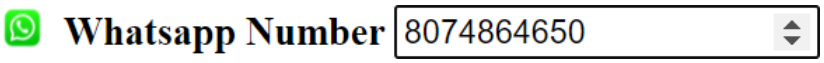
14) Write Code for following output



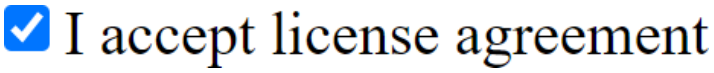
15) Write code for following output



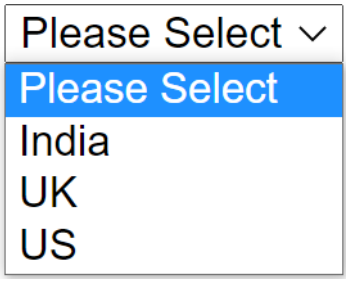
16) Write HTML code to for following Output



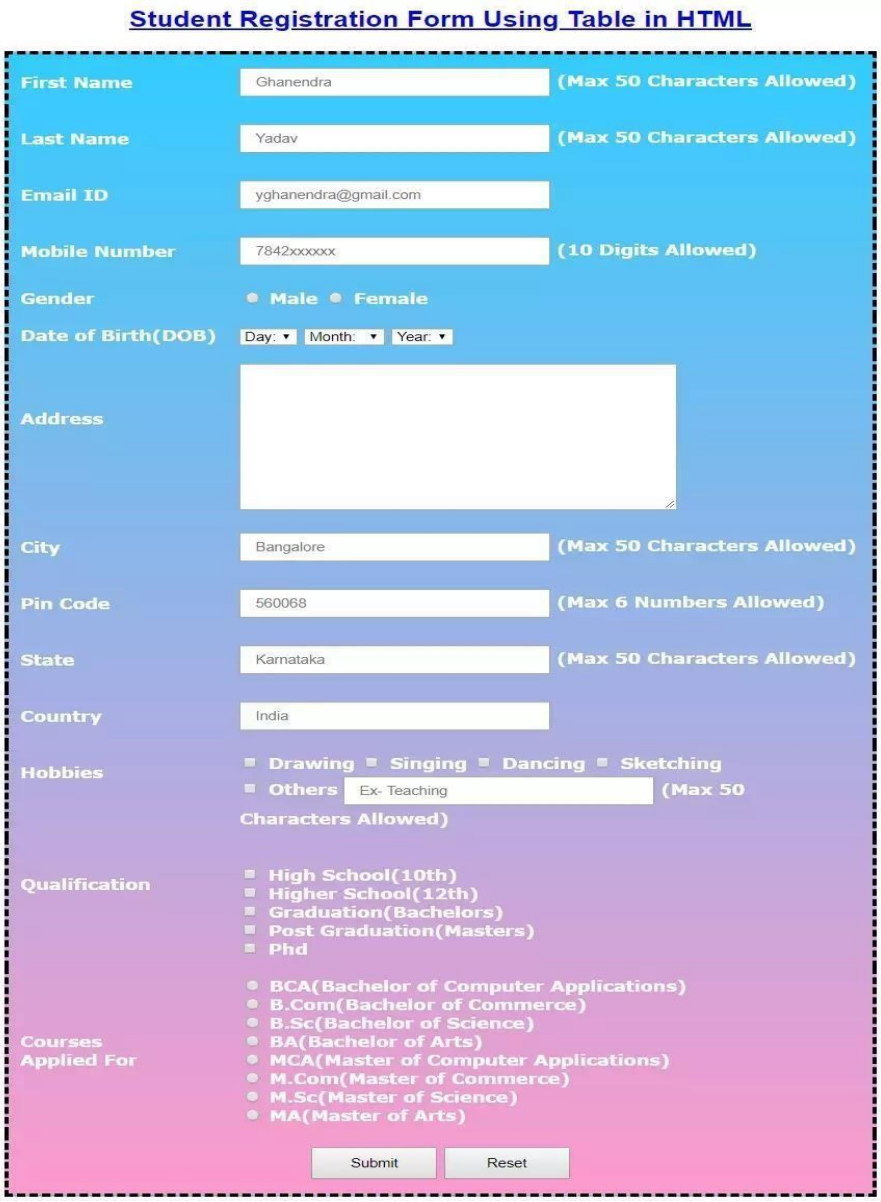
17) Write HTML Code for following output



18) Write HTML code for following output



19) implement the below form



CSS POSITIONS

- ✓ CSS Positions
  - Relative
  - Absolute
  - Fixed
  - Sticky
  - Static
- ✓ Box Sizing
- ✓ z-index

Relative

- ✓ moving element from its original position (actual position)

Syntax:

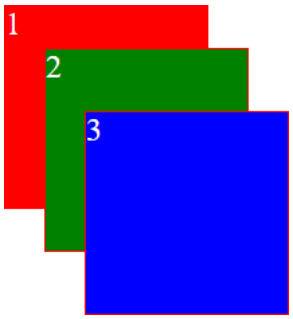
position: relative;
left:
right:
top:
bottom:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Relative Position</title>
    <style>
      div{
        width: 100px;
        height: 100px;
        border: 1px solid red;
        margin: 10px;
        color: white;
      }
      .c1{
        background-color: red;
      }
      .c2{
        background-color: green;
        position: relative;
        left: 20px;
```

```
        }
        .c3{
            background-color: blue;
            position: relative;
            left: 40px;
            bottom: 170px;
        }
    </style>
  </head>
  <body>
    <div class="c1">1</div>
    <div class="c2">2</div>
    <div class="c3">3</div>
  </body>
</html>
```

Output



Absolute

- ✓ move element with respect to ancestor(parent)

  Syntax

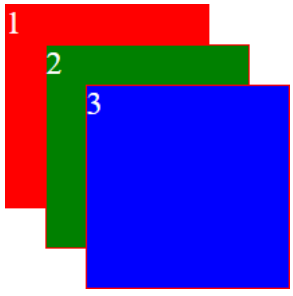
  position: absolute;
  left:
  right:
  top:
  bottom:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Absolute Position</title>
    <style>
      div{
```

```
            width: 100px;
            height: 100px;
            border: 1px solid red;
            margin: 5px;
            color: white;
        }
        #id1{
            background-color: red;
            position:  absolute;
            left: 20px;
            top: 20px;
        }
        #id2{
            background-color: green;
            position: absolute;
            left: 40px;
            top: 40px;
        }
        #id3{
            background-color: blue;
            position: absolute;
            left: 60px;
            top: 60px;
        }
    </style>
  </head>
  <body>
    <div id="id1">1</div>
    <div id="id2">2</div>
    <div id="id3">3</div>
  </body>
</html>
```

Output



Fixed Positions

- ✓ it is used to fix element permanently on webpage

Syntax

position: fixed;
left:
right:
top:
bottom:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Fixed Position</title>
    <style>
      h1{
        color: red;
        position: sticky;
        top: 0;
      }
    </style>
  </head>
  <body>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <h1>Adv Here</h1>
    <p>Paragraph</p>
    <p>Paragraph</p>
    <p>Paragraph</p>
  </body>
</html>
```

Output:

Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph

Adv Here

Sticky

- ✓ element **positioned** based on **users scroll**

Syntax

position: sticky;
top:
left:
bottom:
right:

```
<style>
        h1{
          color: red;
          position: sticky;
          top: 0;
        }
 </style>
```

Output

Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph
Paragraph

Adv Here

Paragraph
Paragraph
Paragraph

Static

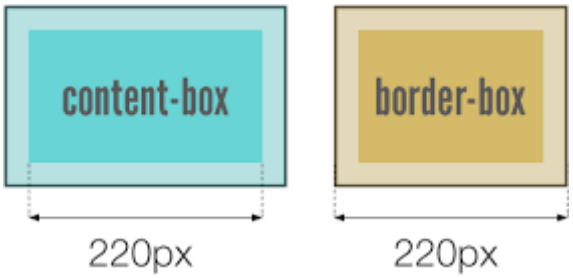
- ✓ **default** position of an element is **static**

Box Sizing

- ✓ adjusting padding, border and margin with in the actual width and height of an element called as **border-box**
- ✓ this feature introduced in **CSS3.X**

Syntax:

box-sizing: border-box;



z-index

- ✓ z-index property specifies the stack order of an element
- ✓ an element with highest stack order will display on top

Syntax

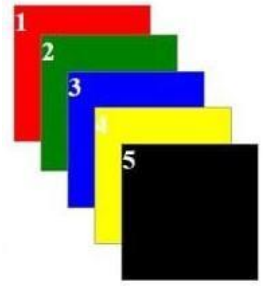
z-index:10;

PRACTICE PAPER

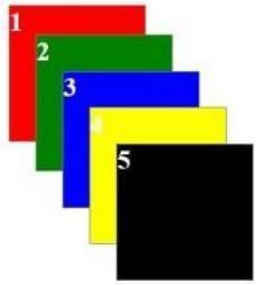
1) Explain Positions in CSS?

2) What are the Differences Between Absolute and Relative Positions?

3) Implement The Below Application Absolute Position



4) Implement the Below Application with CSS Relative Position



5) What is Z-Index in CSS?

6) Which one is the default position in CSS_____

7) How to display Element in center of webpage

Ans:

8) What is border-box in CSS?

JavaScript Basics

- ✓ JavaScript is a **Scripting Language**
- ✓ JavaScript used to develop **Dynamic Web pages**
- ✓ JavaScript used to implement the **Form Validations**
- ✓ JavaScript also used to develop
  - **1) Angular**
  - **2) React**
  - **3) VueJS**
  - **4) NodeJS**
- ✓ Current version of JavaScript is **ES13**
- ✓ ES Stands for **ECMA Script**
- ✓ ECMA Stands for **European Computers Manufacturing Association**
- ✓ JavaScript Released by **Netscape**
- ✓ IDE Stands for **Integrated Development Environment**
- ✓ IDE is used to develop **Applications**
- ✓ Ex**. Notepad**
  - **Edit plus**
  - **Notepad++**
  - **Wordpad**
  - **VSCode**
- ✓ VSCode Given by **Microsoft**
- ✓ VSCode also called as **Visual Studio Code**
- ✓ VSCode is **Open-Source IDE**
- ✓ VSCode Recommended to develop **UI Applications**
- ✓ we will save JavaScript files with **".js"** extension
- ✓ we will include JavaScript with the help of **<script></script> tag**
- ✓ we will execute JavaScript with the help of
  - **1) NodeJS --- node**
  - **2) Browser --- Interpreter**
- ✓ Below **Command** used to execute the JavaScript with node
  - **node demo.js**
- ✓ JavaScript Execution is **Synchronous Execution**
  (Code will execute line by line)
- ✓ JavaScript is **Object Based Scripting Language**
- ✓ vendor/browser provided so many predefined objects
  1) console
  2) document
  3) window
  4) Date
  5) Local Storage

6) SessionStorage
7) Location

✓ console object used to **debug** JavaScript application

Ex.

console.log("debugging soon....");
console.table(["Angular","React","NodeJS","VueJS","MongoDB"]);

(index)	Value
0	'Angular'
1	'React'
2	'NodeJS'
3	'VueJS'
4	'MongoDB'

✓ document object used to manipulate the HTML Elements/DOM Elements

Ex.

document.write("welcome to JavaScript");

✓ window object is the **super object**

✓ Date object used to work with the **Calendar**

```
x=new Date();
document.write(x);
```

✓ Browser Supports two types of **Storages**

1) Local Storage
2) Session Storage

✓ **localStorage** object used to work with **Browsers Local Storage**

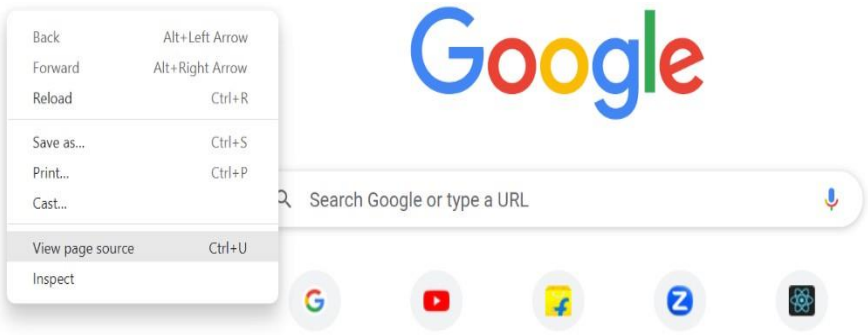
✓ **sessionStorage** object used to work with the **Browsers Session Storage**

✓ **Location** Object used to navigate from **one dynamic webpage to another dynamic webpage**

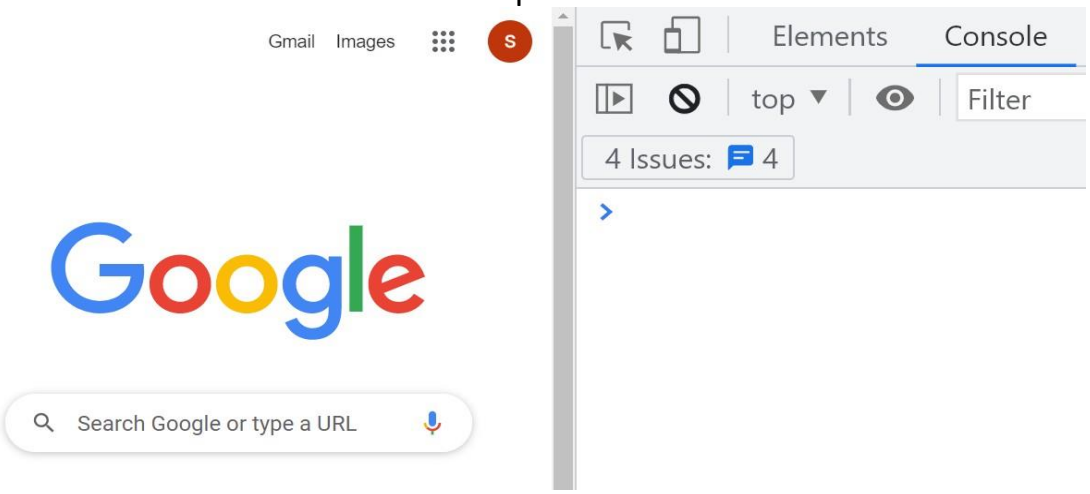
✓ JavaScript is **not secured** scripting language

✓ shortcut to open the **source code**

right click on browser ==> view **page source**



- ✓ shortcut to **open browser console**
  right click on browser window ==> inspect ==> **console**



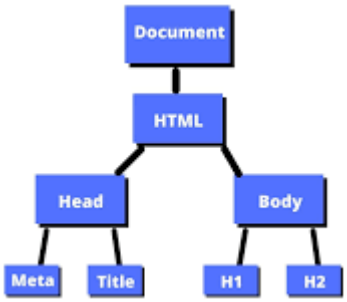
- ✓ comments **wont** executed by browser
- ✓ we have two types of comments
  1) multiline comment
  2) single line comment
- ✓ below **syntax** representing multiline comment

```
/*
        comments
    */
```

- ✓ below **syntax** representing single line comment

```
//
```

- ✓ comments increases **"application readability"**
- ✓ whenever HTML Loaded into Browser Engine **Successfully**, automatically **document object will ready**
- ✓ document object also called as **DOM Object / Real DOM Object**
- ✓ tree structure called as **DOM Tree Structure**
- ✓ **Traversing** in DOM Object called as **DOM Tree Traversing**
- ✓ **Manipulations** in DOM Object called as **DOM Manipulations**



PRACTICE PAPER-1
(BASICS)

1) What is JavaScript _____
2) Is JavaScript used to implement Forms Validations? _____
3) What is the Extension for JavaScript files _____
4) How to Execute JavaScript files _____
5) Which tag used to include JavaScript to HTML _____
6) Is JavaScript used to develop Dynamic web pages?

_____
7) Explain types of web applications?

8) What are the differences between static web application and dynamic web application?

9) Write the Differences Between Programming Languages and Scripting Languages?

10) Is JavaScript execution Synchronous? _____
11) Is JavaScript Object Based Scripting Language?_____
12) Write few Predefined objects provided by JavaScript?

13) How to manipulate DOM Elements _____
14) How to debug JavaScript application _____
15) How to get the Current Data _____

16) How to display Data on webpage?

17) How to display data on browser console?

18) How to display array in the form of table in browser console?

19) Is JavaScript Secured? _____
20) How to View the Page Source?

21) How to open the Browser console?

22) Are semi colons (;) mandatory in JavaScript?
_____

23) Why Comments in JavaScript?

24) JavaScript supports how many types of Comments?

25) Write the Syntax for Single line Comments in JavaScript?

26) Write the Syntax for Multi line Comments in JavaScript?

27) Is JavaScript Compiler Based Scripting Language? _____

28) Is JavaScript Interpreter Based Scripting Language?
_____

FAQ's

1) **What is Scripting language?**
   - ✓ A **scripting language** is a type of **programming language** designed for a runtime system to **automate the execution of tasks**
     Ex. JavaScript
         PERL

2) **What is JavaScript?**
   - ✓ JavaScript is a **Scripting Language**
   - ✓ JavaScript used to develop **dynamic web pages**
   - ✓ JavaScript used to implement **Forms Validations**

- ✓ JavaScript is an **Object Based** Scripting Language
- ✓ Interpreter will convert **JavaScript** to **Browser understandable** format

3) Differences Between **Scripting Language** and **Programming Language?**

Programming Language	**Scripting Language**
A programming language is a **computer language** that is used to communicate with **computers using a set of instructions**.	A scripting language is a type of **programming language** designed for a runtime system to **automate the execution of tasks**
Compiler Based Language	**Interpreted** based Language
It generates a **.exe** file	It does not create a **.exe** file
high **maintenance** cost	less **maintenance** cost
Takes **less time** for **compilation**	Takes **more time** for **Execution**
Ex. C, C++, Java, Scala, COBOL	Ex. Perl, Python, JavaScript
programs consume **more memory**.	Scripting Languages takes **less memory**

4) How to include JavaScript file?

Ans:

```
<script type="text/javascript" src="message.js"></script>
```

5) What is DOM? What is the use of document object?

- ✓ DOM stands for Document Object Model.
- ✓ A document object represents the HTML document
- ✓ It can be used to access and change the content of HTML

6) How to write a comment in JavaScript?

- ✓ There are **two types** of comments in JavaScript.
- ✓ Single Line Comment: It is represented by // (double forward slash)
- ✓ Multi-Line Comment: Slash represents it with asterisk symbol as /* write comment here */

Variables

- ✓ Introduction
- ✓ Rules to declare variables
- ✓ Syntax
- ✓ Datatypes
  - ➢ Primitive datatypes
  - ➢ Non primitive datatypes
- ✓ Ternary Operator
- ✓ differences between var and let
- ✓ const
- ✓ Parctice Paper (Basics)
- ✓ Practice Paper (Variables)
- ✓ Faqs

Introduction

- ✓ variables are used to store the data

  Ex.

  string
  number
  boolean
  arrays
  objects

- ✓ we can declare variables in 4 ways

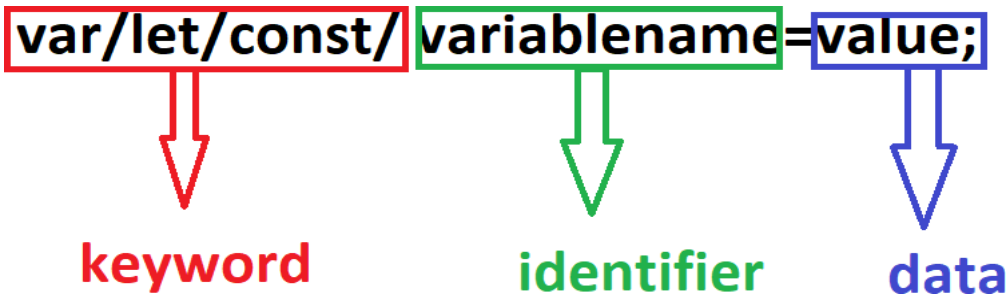
  1) var 2) let 3) const 4) no keyword

- ✓ let and const introduced in ES6 version

rules to declare variables

1) variables declaration should contain a-z,A-Z,0-9,$ and _
2) variables declaration should not start with digits(numbers)

Syntax



datatype

- ✓ datatype representing type of data
- ✓ JavaScript supports two types of datatypes
  1) primitive datatype
  2) non primitive datatype
- ✓ primitive datatypes are not objects, so we can't call methods and properties
- ✓ primitive datatypes divided into 7 types
  1) string
  2) number
  3) boolean
  4) undefined
  5) null
  6) bigint
  7) symbol
- ✓ bigint and symbol introduced in ES6 version
- ✓ non primitive datatypes are objects, so we can call methods and properties
- ✓ non primitive datatypes are classified into
  1) arrays
  2) objects
  3) classes
  4) interfaces

---

string

- ✓ collection of characters called as string
- ✓ we can declare string in three ways
  1) "" (double quotes)
  2) '' (single quotes)
  3) `` (backtick)
- ✓ `` (backtick) introduced in ES6 version
- ✓ `` (backtick) technically called as "template literal"
- ✓ we can include one variable to another variable easily with template literal
- ✓ we can create multiline string with the help of template literal

Example

```
var sub=`ReactJS`;
var wish=`Welcome to ${sub}`;
```

```
        document.write(wish,`<br>`);
```

Output:

```
    Welcome to ReactJS
```

number

- ✓ five types of numbers are possible
  1) decimal
  2) double
  3) hexadecimal
  4) octal
  5) binary
- ✓ hexadecimal prefix with "0x"
- ✓ octal prefix with "0o"
- ✓ binary prefix with "0b"

Example

```
var decimal=100;
document.write(decimal, `<br>`);

var double=100.12345;
document.write(double,`<br>`);

var hexadecimal=0x123ABC;
document.write(hexadecimal,`<br>`);   //1194684
                //JavaScript internally converting
                  hexadecimal to decimal

var octal=0o123;
document.write(octal,`<br>`);
               //JavaScript internally converting
octal to
                  decimal

var binary=0b1010;
document.write(binary,`<br>`);
              //JavaScript internally converting binary
to
               Decimal
```

boolean

- ✓ two boolean values are possible
  1) true
     default value of true is 1
  2) false

default value of false is 0

Example

```
var flag=true;
document.write(flag,`<br>`);

var flag1=false;
document.write(flag1,`<br>`);
```

Ternary Operator

Syntax

condition? True statements: false statements;

Example

```
flag?document.write(`ReactJS`,`<br>`):document.write(`Angular`,`<br>`);
                                    //ReactJS
flag1?document.write(`ReactJS`,`<br>`):document.write(`Angular`,`<br>`);
                                  //Angular
3>2>1?document.write(`ReactJS`,`<br>`):document.write(`Angular`,`<br>`)
;
                                  // Angular
1<2<3?document.write(`ReactJS`,`<br>`):document.write(`Angular`,`<br>`)
;
                                  //ReactJS
```

Differences between undefined and null

Undefined	Null
undefined is primitive datatype	null also a primitive datatype
variable declared but value not assigned called as undefined	null representing blank/empty value
for undefined external initialization not required Ex. var x;	for null external initialization required Ex. var x=null;

undefined with arithmetic operation then result will be NaN
NaN Stands for **Not an Number**
Ex
var a;
var b=10;
a+b; //NaN
	null value converted as 0 in arithmetic operations
	Ex.
	var a=null;
	var b=10;
	a+b; //10

typeof undefined; //undefined	typeof null; // object

bigint

- ✓ bigint is the primitive datatype
- ✓ bigint datatype introduced in ES6 version
- ✓ bigint datatype used to represent the large number
- ✓ range of bigint is "2^53 - 1" to "-2^53 - 1"
- ✓ bigint numbers suffix with "n"

Example

```
var
large=1234567891234567891234567891234567891234567891234567891234
567891234567891234567891234567891234567891234567891234567891234
56789123456789123456789123456789n;
document.write(large,`<br>`);
```

Symbol

- ✓ Symbol is primitive datatype
- ✓ Symbol datatype introduced in ES6 version
- ✓ Symbol datatype used to hide identifier to meet security criteria

Example

```
var security=Symbol (100);
document.write(security.description,`<br>`);
        //100
```

Differences Between var and let

<table>
<tr><th>Var</th><th>Let</th></tr>
<tr><td>var introduced in ES1 version</td><td>let introduced in ES6 version</td></tr>
<tr><td>var keyword allows duplicate variables
Ex.
var x=10;
var x=20;
document.write(x); //20</td><td>let wont allows duplicate variables
Ex.
let x=10;
let x=20;
document.write(x); //Error</td></tr>
<tr><td>var contains functional scope</td><td>let contains block scope</td></tr>
<tr><td colspan="2">Ex.
function func_one () {
{
var x=100;
let y=200;
}
document.write(x); //100
document.write(y); //Error
}
func_one();</td></tr>
</table>

variable hoisting raised because of var keyword
Ex.
document.write(x); //undefined
var x=100;
	We can overcome variable hoisting With let keyword
	Ex.
	document.write(x); //Error
	let x=100;

global polluting issue raised with var keyword	we can overcome global polluting issue with let keyword
Ex. ***var/let x=100;*** ***{*** ***var/let x=200;*** ***}*** ***document.write(x);*** ***o/p: var : 100*** ***let : 200***	

const

- ✓ const is the keyword
- ✓ const keyword introduced in ES6 version
- ✓ reinitialization not possible with const keyword

Example1

```
const x=100;
x=200;
document.write(x);                    //Error
```

Example2

```
const arr= [10,20,30,40,50];
arr= [];                              //Error
arr= [100,200,300,400,500];           //Error
```

Example3

```
const arr= [10,20,30,40,50];
    arr [0] =100;
    arr [1] =200;
    arr [4] =500;
    arr [5] =600;
    document. write(arr);             //[100,200,30,40,500,600]
```

Example4

```
const obj= {
        key1:'Hello_1',
        key2:'Hello_2',
        key3:'Hello_3'
    };
    obj= {};                          //Error
```

```
obj= {key1: `Welcome_1`,
    key2: `Welcome_2`,
    key3: `Welcome_3`};          //Error
```

Example5

```
const obj= {
        key1:'Hello_1',
        key2:'Hello_2',
        key3:'Hello_3'
      };
      obj.key1="welcome_1";
      obj.key4="welcome_4";
```

Output:

{key1:" welcome_1", key2:'Hello_2', key3:" Hello_3", key4:" welcome_4"}

PRACTICE PAPER
(variables)

1) Why variables?

2) How to declare variables?

3) Write the Rules to declare variables?

4) What is datatype?

5) How many types of Datatypes?

6) What are the Differences Between Primitive datatype and non-primitive datatype?

7) Write the Primitive datatypes?

8) Write the non-primitive datatypes?

9) Write the JavaScript program
Store value **100** to **x** variable
Store value **200** to **y** variable
Find the addition of **x and y** variables and store to **z variable**
Print the **z variable**

10)Write the JavaScript Program
Store value **10** to **x** variable
Find the **square** of **x** variable and store result to **y variable**
Print **y variable**

11) What is string?

12) How to declare String in JavaScript?

13) Find the Result

```
var firstname=" ExcelR";
var lastname=" technologies";
var fullname=firstname+" "+lastname;
document. Write(fullname);
```

o/p:

14) Find the Result

```
var sub=`fullstack`;
var msg=`welcome to ${sub}`;
document. write(msg);
```

o/p:

15) Write the few points related to backtick operator?

16) What are the Boolean values?

17) Write the Syntax for ternary operator?

18) 9>8>7? document. write("java"): document. write("dotnet");

_____

19)1<2<3? document. write("java"): document. write("dotnet");

_____

20)NaN Stands for _____

21)Write the Differences Between undefined and null?

22) What are the differences between == and === with examples?

23) let and const introduced in_____version

24) write the range of bigint datatype _____

25) bigint datatype introduced in_____version

26) bigint datatype suffix _____

27) how to hide identifiers _____

28) write the differences between var and let?

29) write few points related to const keyword with examples?

30) What is variable hoisting?

31) How to declare block in JavaScript _____

32) Find the Result
var x=100;
var x= 200;
document. Write (x);
o/p:

33) Find the Result
let x=100;
let x=200;
document. Write (x);
o/p:

34) Find the Result
const x=100;
const x=200;
document. write(x);
o/p:

Faqs

1) Explain variables?

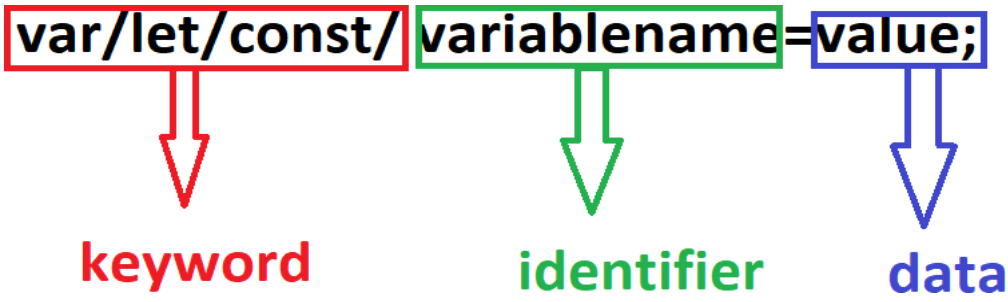
✓ variables are used to store the data
Ex.
string
number
boolean
arrays
objects

- ✓ we can declare variables in 4 ways
  1) var 2) let 3) const 4) no keyword
- ✓ let and const introduced in ES6 version

rules to declare variables

1) variables declaration should contain a-z,A-Z,0-9,$ and _
2) variables declaration should not start with digits(numbers)

Syntax



2) Explain Datatypes in JavaScript?

- ✓ datatype representing type of data
- ✓ JavaScript supports two types of datatypes
  3) primitive datatype
  4) non primitive datatype
- ✓ primitive datatypes are not objects, so we can't call methods and properties
- ✓ primitive datatypes divided into 7 types
  1) string
  2) number
  3) boolean
  4) undefined
  5) null
  6) bigint
  7) symbol
- ✓ bigint and symbol introduced in ES6 version
- ✓ non primitive datatypes are objects, so we can call methods and properties
- ✓ non primitive datatypes are classified into
  1) arrays
  2) objects
  3) classes
  4) interfaces

---

3) Explain Template Literal in JavaScript?

- ✓ `` (backtick) introduced in ES6 version
- ✓ `` (backtick) technically called as "template literal"
- ✓ we can include one variable to another variable easily with template literal
- ✓ we can create multiline string with the help of template literal

Example

```
var sub=`ReactJS`;
var wish=`Welcome to ${sub}`;
document.write(wish,`<br>`);
```

```
Output:
   Welcome to ReactJS
```

4) write the Differences Between undefined and null?

Undefined	Null
undefined is primitive datatype	null also a primitive datatype
variable declared but value not assigned called as undefined	null representing blank/empty value
external initialization not required Ex. var x;	for null external initialization required Ex. var x=null;

undefined with arithmetic operation then result will be NaN
NaN Stands for **Not an Number**
Ex
var a;
var b=10;
a+b; //NaN
	null value converted as 0 in arithmetic operations
	Ex.
	var a=null;
	var b=10;
	a+b; //10

typeof undefined; //undefined	typeof null; // object

5) differences between var and let?

Var	**Let**
var introduced in ES1 version	let introduced in ES6 version
var keyword allows duplicate variables Ex. ***var x=10;*** ***var x=20;***	let wont allows duplicate variables Ex. ***let x=10;*** ***let x=20;*** ***document.write(x); //Error***

document.write(x); //20	
var contains functional scope	let contains block scope

Ex.
function func_one () {
{
var x=100;
let y=200;
}
document.write(x); //100
document.write(y); //Error
}
func_one();	

variable hoisting raised because of var keyword
Ex.
document.write(x); //undefined
var x=100;
	We can overcome variable hoisting With let keyword
	Ex.
	document.write(x); //Error
	let x=100;

global polluting issue raised with var keyword	we can overcome global polluting issue with let keyword
Ex. ***var/let x=100;*** ***{*** ***var/let x=200;*** ***o/p: var : 100*** ***}*** ***let : 200*** ***document.write(x);***	

6) Explain const keyword?

- ✓ const is the keyword
- ✓ const keyword introduced in ES6 version
- ✓ reinitialization not possible with const keyword

7) what is variable hoisting?

- ✓ Accessing the variables before its declaration and initialization with var keyword called as **variable hoisting**
- ✓ Variables are **partially hoisted**
- ✓ We can overcome variable hoisting with let and const keyword

8) what is temporal dead zone?

- ✓ A let or const variable is said to be in a "temporal dead zone" (TDZ)
- ✓ let and const variables will execute if interpreter reaches the line (contain declaration & initialization)

- ✓ Temporal Dead Zone is the period of time during which the let and const declarations cannot be accessed.

9) what is global polluting issue?

- ✓ define too many variables that are globally accessible
- ✓ global members are affected because of block members
- ✓ global polluting issue raised because of var keyword
- ✓ we can overcome global polluting issue with let and const keyword

LOOPS

- ✓ **Array**
- ✓ **for loop**
- ✓ **forEach() loop**
- ✓ **for of() loop**
- ✓ **if else condition**
- ✓ **switch cases**

Array

- ✓ collection of indexed elements called as **array**
- ✓ index starts from **0**
- ✓ we will represent arrays with **[ ]**
- ✓ we will access array elements with **indexes**
- ✓ we will iterate array elements with loops

Ex.

for() loop
forEach() loop
for of() loop
while()
do while()

---

for loop

- ✓ loops through a block of code a number of times

Syntax

```
for (expression 1; expression 2; expression 3) {
        // code block to be executed
}
```

Expression1 --- initialization
Expression2 --- condition
Expression3 --- increment/decrement

Example

```
<script>
   let arr1=[10,20,30,40,50];
   for(let i=0;i<arr1.length;i++){
      document.write(arr1[i],`<br>`);
   }
</script>
```

Output

10
20
30
40
50

forEach() loop

- ✓ The forEach() method calls a function for each element in an array.
- ✓ The forEach() method is not executed for empty elements.

Syntax

array.forEach(function(currentValue, index, arr), thisValue)

Example

```
<script>
        let arr1=[10,20,30,40,50];
        arr1.forEach((element,index)=>{
           document.write(element,`<br>`);
        });
</script>
```

Output

10
20
30
40
50

for of() loop

- ✓ The JavaScript for of statement loops through the values of an iterable object.
- ✓ It lets you loop over iterable data structures such as Arrays, Strings, Maps, NodeLists

Syntax

```
for (variable of iterable) {
  // code block to be executed
}
```

Example

```
<script>
        let arr1=[10,20,30,40,50];
        for(let x of arr1){
           document.write(x,`<br>`);
        }
</script>
```

Output

10
20
30
40
50

if else condition

- ✓ Conditional statements are used to perform different actions based on different conditions.
- ✓ In JavaScript we have the following conditional statements
  1) If
  2) Else
  3) Else if
  4) switch

if Syntax

```
if (condition) {
        //  block of code to be executed if the condition is true
}
```

If Else Syntax

```
if (condition) {
        //  block of code to be executed if the condition is true
} else {
        //  block of code to be executed if the condition is false
}
```

Else if Syntax

```
if (condition1) {
  //  block of code to be executed if condition1 is true
} else if (condition2) {
   //block of code to be executed if the condition1 is false and condition2 is
     true
} else {
    // block of code to be executed if the condition1 is false and condition2 is
       false
}
```

Example

```
<script>
        if(9>8>7){
```

```
            document.write("DotNet");
        }else{
            document.write("Java");
        }
        //Java
    </script>
```

Example

```
  <script>
        if(true){
            document.write("DotNet");
        }else{
            document.write("Java");
        }
        //DotNet
    </script>
```

Switch

- ✓ The switch statement is used to perform different actions based on different conditions.
- ✓ Use the `switch` statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
  break;
  default:
    // code block
}
```

Example

```
<script>
        switch(new Date().getDay()){
            case 0:
                document.write("Sunday",`<br>`);
                break;
            case 1:
```

```
            document.write("Monday",`<br>`);
            break;
        case 2:
            document.write("Tuesday",`<br>`);
            break;
        default:
            document.write("No Matches");
            break;
    }
</script>
```

PRACTICE PAPER

1)How to represent arrays_____

2)Index starts from_____

3)How to access array elements _____

4)How to iterate array elements?

5)Write the for () loop Syntax?

6)Iterate following array with for loop

let arr1= [`Java`, `dotnet`, `ui`, `react`, `angular`];

7)Write the forEach () loop Syntax?

8)Iterate following array with forEach () loop

let arr1= [`Java`, `dotnet`, `ui`, `react`, `angular`];

9)Write the for of () loop syntax?

10) Iterate following array with for of () loop
let arr1= [`Java`, `dotnet`, `ui`, `react`, `angular`];

11) Write the if else condition syntax?

12) Find the Result
```
If(true) {
        document. write("java");
} else {
        Document. Write("dotnet");
}
```
o/p:

13) Write the switch cases syntax in javascript?

14) Write one basic example for switch case in javascript?

15) Find the Result
let arr1 = [10,20,30,40,50];

arr1[0]_____	arr1.at (0) _____
arr1[4]_____	arr1.at (-1) _____
arr1[10]_____	arr1.at (-5) _____
arr1[-1] _____	arr1.at (-10) _____

Functions in JavaScript

- ✓ **Function**
- ✓ **Types of Functions**
  - ➢ **Function Declaration / Named Functions**
  - ➢ **Function Expression / Anonymous Functions**
  - ➢ **Arrow Functions**
- ✓ **Handling click event**
- ✓ **Reading/Writing Operations on input control**
- ✓ **Adding the content to HTML Element**
- ✓ **Rest Parameters**
- ✓ **Default Parameters**
- ✓ **Examples**
- ✓ **Practice Paper**
- ✓ **FAQ's**

Function

- ✓ Particular **business logic** called as Function
  (or)
- ✓ Set of statements called as **Function**
- ✓ Functions are used to reuse **Business Logic**

Types of Functions

- ✓ **Function Declaration / Named Functions**
- ✓ **Function Expression / Anonymous Functions**
- ✓ **Arrow Functions**

Function Declaration / Named Functions

- ✓ The function with user defined name called as **Function Declaration / Named Functions**

Syntax

Function Declaration

```
function functionname (param1, param2, param3,……param n){
        //business logic
}
```

Function Calling

```
functionname (arg1, arg2, arg3 ..... arg n);
```

Function Expression / Anonymous Functions

- ✓ The function without name called as **Function Expression / Anonymous Functions**
- ✓ We can store function expression to **variables**

Syntax

Function Expression

```
let variablename = function (param1, param2, param3,……param n){
        //business logic
}
```

Function Calling

```
variablename (arg1, arg2, arg3..... arg n);
```

Arrow Functions

- ✓ Arrow Functions are introduced in **ES6** version
- ✓ We will represent Arrow functions with **"=>"**
- ✓ We can store arrow functions to **variables**
- ✓ Arrow functions follows **"Lexical Scope"**

Syntax

Function Expression

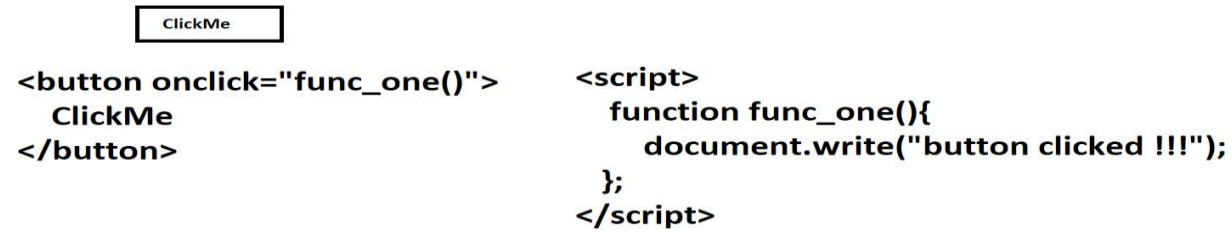
```
let variablename = (param1, param2, param3,…param n)=>{
        //business logic
}
```

Function Calling

```
variablename (arg1, arg2, arg3..... arg n);
```

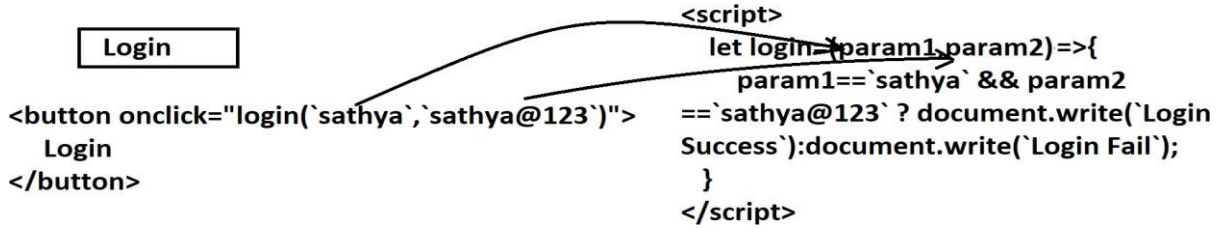
Handling click event

Ex1:



Whenever we click button automatically **"func_one ()"** will execute

Ex2:

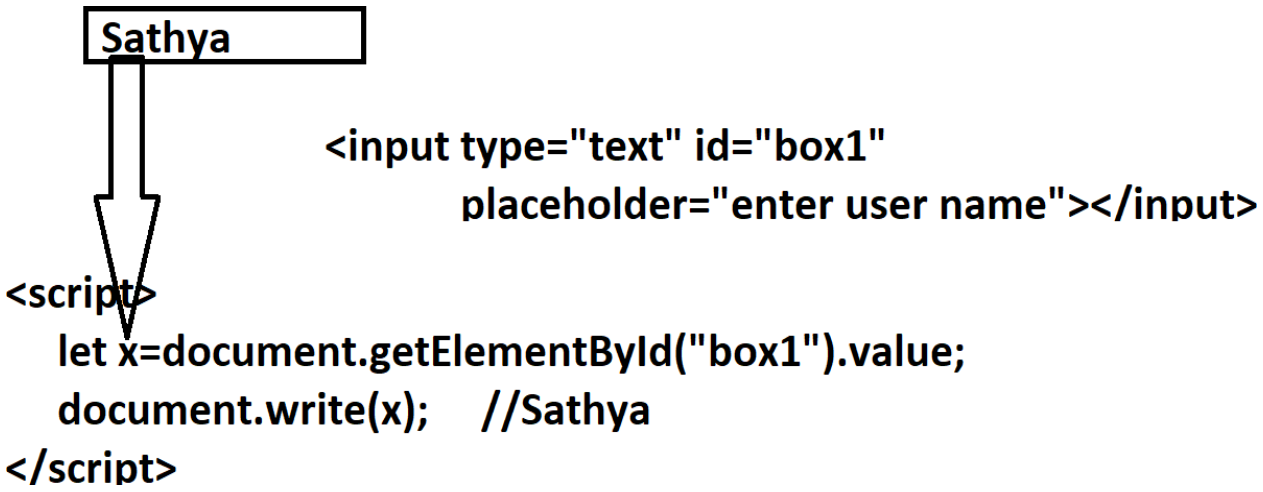


Whenever we click button automatically **"login ()"** function will execute with arguments

Reading/Writing Operations on input control

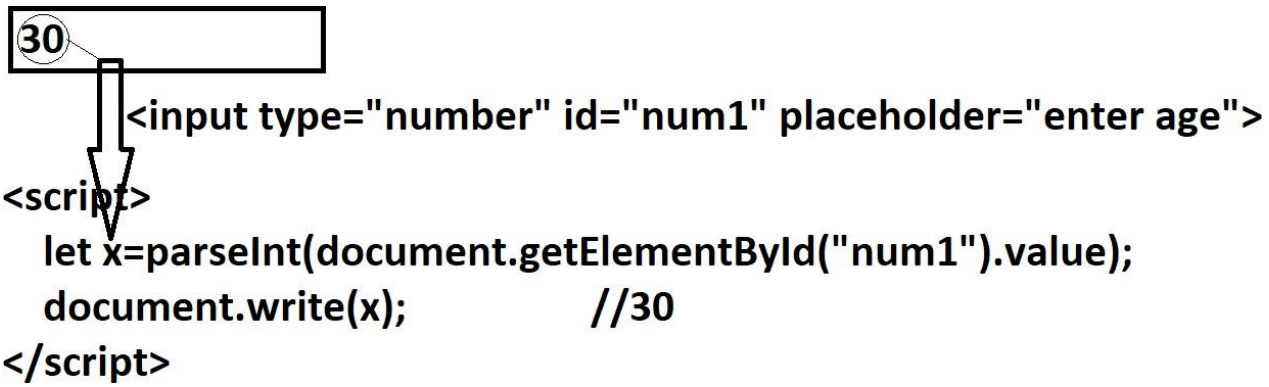
Reading Operation

- ✓ **getElementById** () is the method used to get the reference of HTML Element

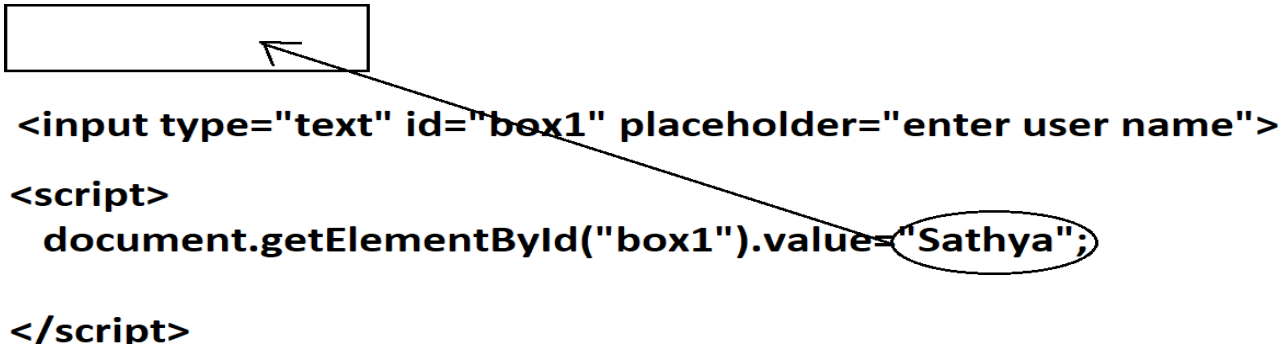


Reading Operation

- ✓ **parseInt** () is the predefined method used to convert string datatype to number datatype



Writing Operation



Adding the content to HTML Element

- ✓ **innerHTML** is the property used to add content to HTML Element

```
<h1 id="res">

</h1>

<script>
    document.getElementById("res").innerHTML="Sathya";
</script>
```

Rest Parameters

- ✓ … called as spread operator
- ✓ Spread operator introduced in ES6 version
- ✓ The rest parameter syntax allows a function to accept an indefinite number of arguments as an array

  Syntax

```
function func_one(...rest){


}
```

Rules for Spread Operator

1. We can pass only ***one*** Spread Operator per function
2. Spread Operator always ***last*** in Parameters Occurrences

Default Parameters

- ✓ While defining the functions we will initialize the parameters
- ✓ This concept also introduced in ES6 version

  Syntax

```
function func_one(param1="Sathya",param2="Technologies",param3="ReactJS"){

}
```

Default parameters

Examples

Ex1.

Function Declaration

```
<script>
    function func_one (){
      document.write("welcome to functions","<br>");
    }
    func_one();
    func_one();
    func_one();
    func_one();
    func_one();
</script>
```

Function Expression

```
<script>
    let func_one=function(){
      document.write("welcome to functions","<br>");
    }
    func_one();
    func_one();
    func_one();
    func_one();
    func_one();
</script>
```

Arrow Functions

```
<script>
    let func_one= () => {
      document.write("welcome to functions","<br>");
    }
    func_one();
    func_one();
    func_one();
    func_one();
    func_one();
</script>
```

Output:

welcome to functions
welcome to functions
welcome to functions
welcome to functions
welcome to functions

Example2:
Arrow Function

```
<script>
        let func_one= (param1, param2, param3) => {
           document. write (param1, param2, param3,"<br>");
        }
        func_one (100,200,300);     //100 200 300
        func_one(`ReactJS`,`NodeJS`,`MongoDB`); //ReactJS NodeJS MongoDB
        func_one (10,20,30,40);    //10 20 30
        func_one ();               //undefined undefined undefined
        func_one(undefined,`Hello`); //undefined Hello undefined
        func_one(null,undefined,null); //null undefined null
        func_one(undefined,undefined,undefined);
                                   //undefined undefined undefined
        func_one(null,null,null);   //null null null
    </script>
```

Output:

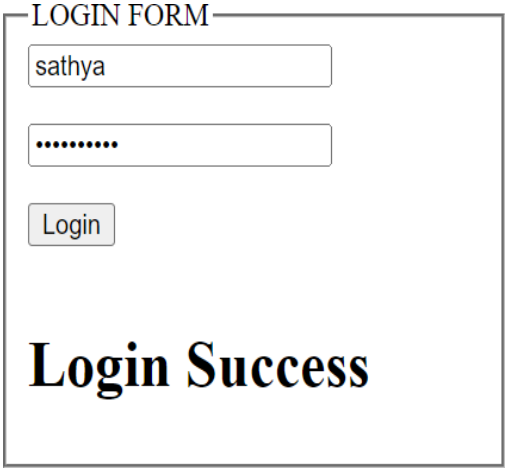
100200300
ReactJSNodeJSMongoDB
102030
undefinedundefinedundefined
undefinedHelloundefined
nullundefinednull
undefinedundefinedundefined
nullnullnull

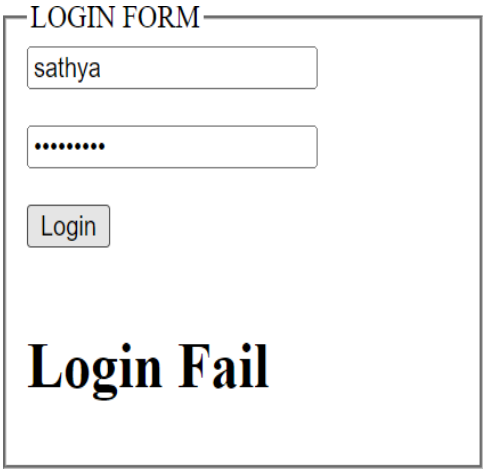
Example3:

```
<fieldset>
        <legend>LOGIN FORM</legend>
        <input type="text" id="uname" placeholder="Enter User Name">
        <br><br>
        <input type="password" id="upwd" placeholder="Enter Password">
        <br><br>
        <button onclick="login()">Login</button>
        <br><br>
        <h1 id="res"></h1>
    </fieldset>
    <script>
```

```
        let login=()=>{
            let x=document.getElementById("uname").value;
            let y=document.getElementById("upwd").value;
            if(x=="ExcelR" && y=="ExcelR@123"){
                document. getElementById("res"). innerHTML="Login Success";
            } else {
                document.getElementById("res").innerHTML="Login Fail";
            }
        }
    </script>
```

Output



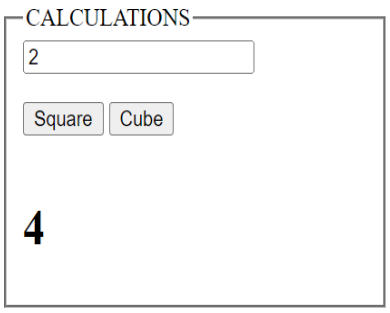


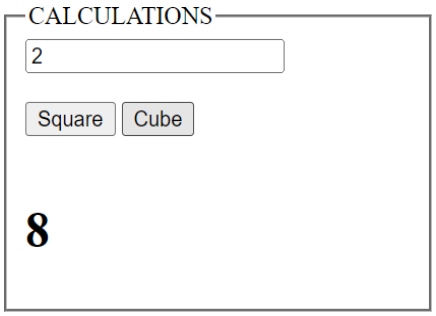
Example3:

```
<fieldset>
    <legend>CALCULATIONS</legend>
    <input type="number" id="num1" placeholder="Enter Number">
    <br><br>
    <button onclick="square()">Square</button>
    <button onclick="cube()">Cube</button>
    <br><br>
    <h1 id="res"></h1>
  </fieldset>
  <script>
    let square= () => {
        let x=parseInt (document. getElementById("num1").value);
        document.getElementById("res").innerHTML=x*x;
    };
    let cube= () => {
        let x=parseInt (document. getElementById("num1").value);
```

```
        document.getElementById("res").innerHTML=x*x*x;
    };
  </script>
```

Output:





Example4:

Function return number

```
<script>
    let func_one= () =>{
       return 100;
    }
    let x=func_one();
    document.write(x);        //100
 </script>
```

Example5:

Function return string

```
<script>
    let func_one= ()=>{
       return `ExcelR`;
    }
    let x=func_one ();
    document.write(x);  //ExcelR
  </script>
```

Example6:

Function return Boolean

```
<script>
    function func_one () {
       return true;
    }
    let x=func_one ();
    x? document.write("ReactJS"): document. write("Java");  //ReactJS
  </script>
```

Example7:
Function return array

```
<script>
  let func_one=()=>{
    return [`React`,`Angular`,`VueJS`,`NodeJS`,`MongoDB`];
  }
  let arr=func_one();
  arr.forEach((element,index)=>{
    document.write(element,`<br>`);
  });
</script>
```

Output

React
Angular
VueJS
NodeJS
MongoDB

JSON

- ✓ JSON Stands for JavaScript Object Notation
- ✓ JSON also called as JavaScript Objects
- ✓ JSON used to transfer the data over the Network
- ✓ JSON is light weight

Syntax
Objects -------- {}
Arrays---------- []
Data------------ key & value pairs
Key & value separated by using ":"

Example8:
Function should return object

```
<script>
    function func_one(){
      return {
        key1:`Hello`,
        key2:`Welcome`,
        key3:`React`
      }
    }
    let obj=func_one();
    document.write(obj.key1,obj.key2,obj.key3);
```

```
        //Hello Welcome React
    </script>
```

Example9:

Function Return Array of Objects

```
<script>
    let func_one=()=>{
        return [{e_id:111,e_name:`e_one`,e_sal:10000},
                {e_id:222,e_name:`e_two`,e_sal:20000},
                {e_id:333,e_name:`e_three`,e_sal:30000},
                {e_id:444,e_name:`e_four`,e_sal:40000},
                {e_id:555,e_name:`e_five`,e_sal:50000}];
    }
    let emps=func_one();
    document.write(`<table border="1"
                        align="center"
                        cellspacing="10px"
                        cellpadding="10px">
                    <tr>
                        <th>e_id</th>
                        <th>e_name</th>
                        <th>e_sal</th>
                    </tr>`);
    emps.forEach((element,index)=>{
        document.write(`<tr>
            <td>${element.e_id}</td>
            <td>${element.e_name}</td>
            <td>${element.e_sal}</td>
        </tr>`);
    });
    document.write(`</table>`);
</script>
```

Output:

e_id	e_name	e_sal
111	e_one	10000
222	e_two	20000
333	e_three	30000
444	e_four	40000
555	e_five	50000

Example10:

```
<script>
    function func_one(...param1){
      document.write(param1,`<br>`);
    }
    func_one(10,20,30,40,50);
                    //[10,20,30,40,50]
    func_one(`ReactJS`)
                    //[`ReactJS`]
    func_one();
                    //[]
    func_one(undefined, null);
                    //[undefined, null]
  </script>
```

Example11:

```
<script>
    function func_one(param1,param2,...param3){
      document.write(param1,param2,param3);
    }
    func_one(10,20,30);
        //10 20 [30]
    func_one(`Java`,`DotNet`,`ReactJS`,`Angular`);
        //Java DotNet [`ReactJS`,`Angular`]
    func_one();
        //undefined undefined []
  </script>
```

Example 12:

```
<script>
    function func_one(param1=`Hello`,param2=`Welcome`){
      document.write(param1,param2);
    }
    func_one();
          //Hello Welcome
    func_one(`ReactJS`,`Angular`);
          //ReactJS Angular
    func_one(undefined,undefined);
          //Hello Welcome
    func_one(null,null);
          //null null
  </script>
```

Example 13:

```
<script>
    function func_one(param1,param2="Hello",...param3){
      document.write(param1, param2, param3);
    }
    func_one();
        //undefined Hello []
    func_one("welcome",undefined,"ReactJS");
        //welcome Hello ["ReactJS"]
    func_one(undefined,undefined,undefined);
        //undefined Hello [undefined]
    func_one(null,null,null);
        //null null [null]
  </script>
```

Example 14:

```
<script>
    function func_one(param1,param2){
      if(param1=="ExcelR" && param2=="ExcelR@123")
        return "Login Success";
      else
        return "Login Fail";
    }
    let res=func_one("ExcelR","ExcelR@123");
    document.write(res);
        //Login Success
  </script>
```

Example15:

```
 <script>
    let func_one=()=>document.write("welcome to arrow functions");
    func_one();
  </script>
```

Output:

welcome to arrow functions

Example16:

```
<script>
    let func_one=()=>"welcome to arrow functions";
    let res=func_one();
    document.write(res);
        //welcome to arrow functions
</script>
```

Example17:

```
<script>
    let func_one=param1=>param1;
    let res=func_one("ReactJS");
    document.write(res);
        // /ReactJS
  </script>
```

Example18:

```
<script>
    let obj={
      sub:``,
      my_func:function(){
        let inner_func=()=>{
          this.sub=`ReactJS`;
        }
        inner_func();
      }
    };
    obj.my_func();
    document.write(obj.sub);
        //Arrow functions ===> ReactJS
        //Function expression ===> ""
  </script>
```

Practice Paper

1) What is function?
   Ans:

2)Set of statements called as _____

3)Are functions used to reuse business logic? _____

4)What are the types of functions?

5)Write the Syntax for function declaration?

6)Write the Syntax for function expression?

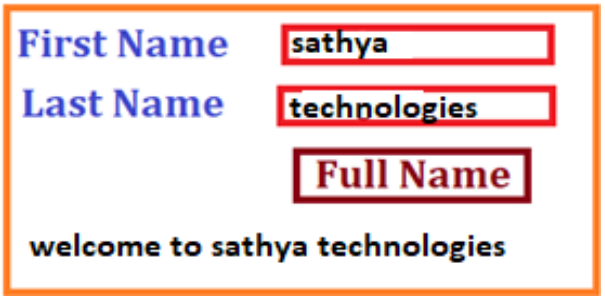
7) Write the Syntax for arrow functions?

8) Can I store function expression and arrow functions to variables? _____
9) How to represent arrow functions_____
10) Arrow functions are introduced in _____version
11) How to read **data and write** from <input> tag _____
12) How to get the Reference of <input type="text" id=" username">
Ans

13)How to set the content to HTML Element _____
14)What are the differences between innerHTML and innerText in JavaScript?
Ans:

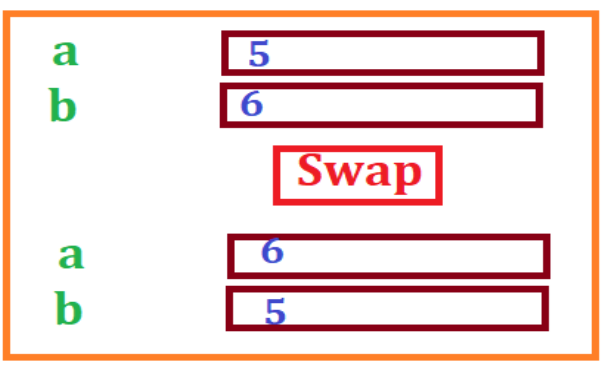
15) User Enters first name, Last name then they will click fullname. Now full name has to display in Input3

16) User Enter FirstName, Last Name Then when they click fullname then Full name display in h1 element

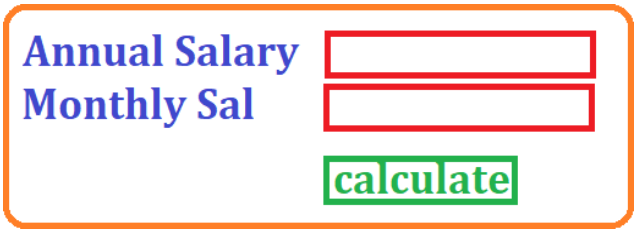


17) User Enter a, b values, then What ever Button they click that result has to Display in output area.

18) User Enter a, b values then they click on Button, now swap the numbers then display in next input controls

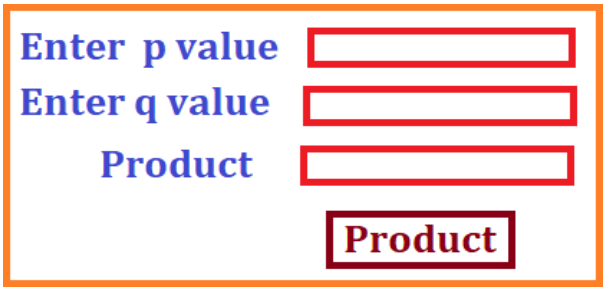


19) User Enter annual salary then They click on calculate button. Now monthly salary has to display in TextBox2

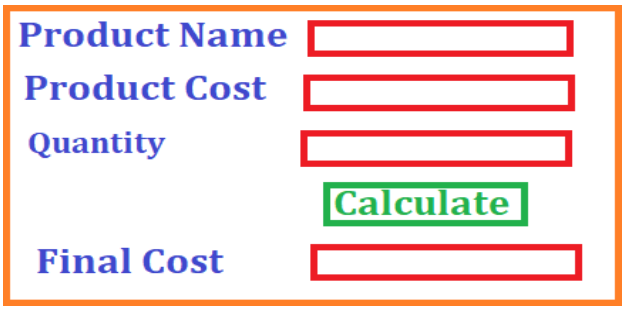


20) User Enter p value, q value

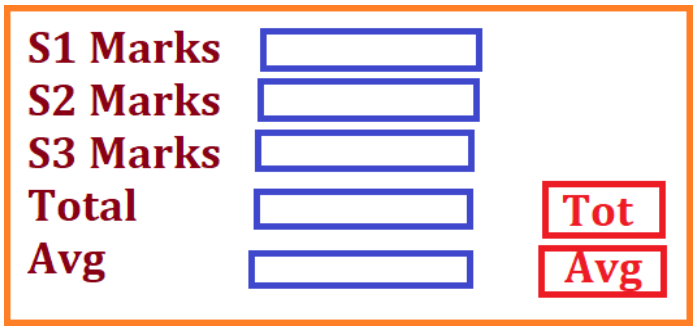
Then they click on button. Now product Has to display in input control 3



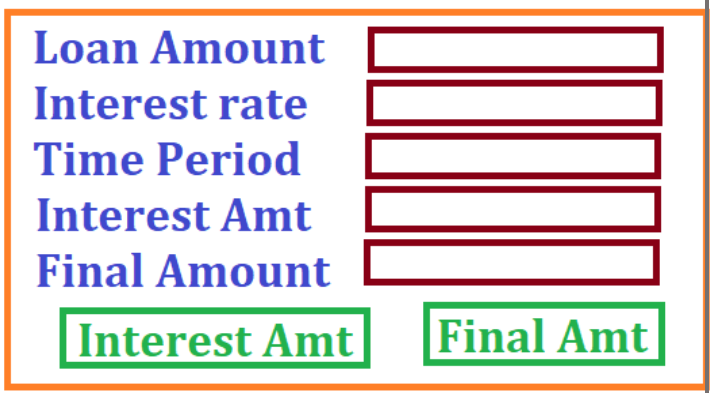
21) User Enters product name, cost, Quantity. Now if he clicks calculate button Then bill amount has to display in Input4

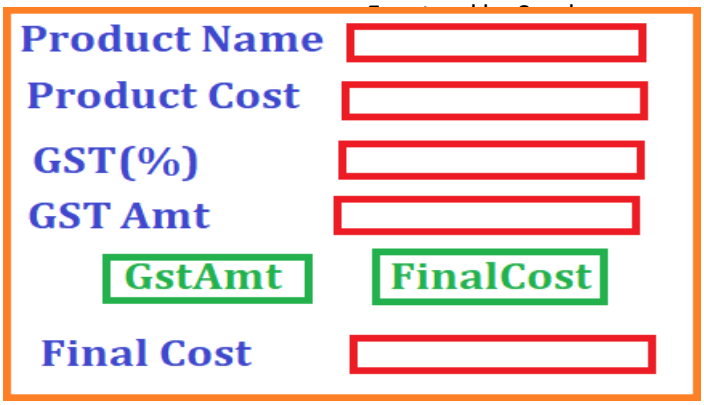


22) User Enter 3 subject marks, then if they click on total it has to display In Input4, if they click on Avg then Collect total from textbox4 and find Average then displays in Input5



23) User Enter Loan Amount, Interest rate, time period. Now if they click on interest Amount It has to calculate interest amount and display in Input4, If they click on Final Amount then It has to display in Input5





24) User Enter Product name, Cost, gst %. Now if they click on GstAmt, GstAmt has to display in Input4. Now if they click on Final Cost then Productcost+gstAmt has to display in Input5.

25) write the answers for below code

```
<script>
    function func_one (param1, param2, param3) {
        document. write (param1, param2, param3, "<br>");
    }
    func_one("ReactJS","NodeJS","MongoDB");
        Ans:
    func_one (100,200,300,400);
        Ans:
    func_one ();
        Ans:
    func_one(undefined,"Hello");
        Ans:
    func_one (null, null, null);
        Ans:
</script>
```

26) what is spread operator? Write few points
Ans:

27) how to represent spread operator_____
28) is spread operator released in ES6 version_____
29) write the rules to use spread operator?
Ans:

30) what are default parameters in functions?

Ans:

31) can we pass more than one default parameter (yes/no)? _____
32) will undefined overrides the default value (yes/no)? _____
33) will null overrides the default value (yes/no)? _____
34) write the program with function declaration, function expression and arrow function
    - name of the function is **"demo_func"**
    - should display **"welcome to named functions"**
    - call the **"demo_func"**
35) write the program with function declaration, function expression and arrow function
    - ✓ name of the function is "demo_func"
    - ✓ should contain following parameters
        1)formal parameter with "param1"
        2)default parameter with "param2" with value as "Hello"
        3)spread operator with "param3"
    - ✓ call the demo_func without any arguments (no data)

36) create the function declaration/function expression/arrow function should return number **"10"**
Ans:

37) create the function declaration/function expression/arrow function should return string **"ExcelR"**
Ans:

38) create the function declaration/function expression/arrow function should return Boolean "**true"**
Ans:

39) create the function declaration/function expression/arrow function should return **[100,200,300,400,500]**
Ans:

40) create the function declaration/function expression/arrow function should return **{sub_one: `ReactJS`, sub_two: `NodeJS`, sub_three: `MongoDB`}**
Ans:

41) function declaration/function expression/arrow function should return following array of objects

[{p_id:111, p_name:"p_one", p_cost:10000},
{p_id:222, p_name:"p_two", p_cost:20000},
{p_id:333, p_name:"three", p_cost:30000},
{p_id:444, p_name:"p_four", p_cost:40000},
{p_id:555, p_name:"p_five", p_cost:50000}]

Display result in the form of a table

42) function should accept two parameters
If param1 and param2 are "admin" return true otherwise false
And apply ternary operator

43) function should accept one parameter i.e., number Return square of a Number

44) function should accept one parameter i.e., number Return even / odd

45) function should accept one parameter i.e.; number Return prime number or not

46) function should accept three parameters
1) param1 is number
2) param2 also number
3) param3 is string (i.e., add / sub / mul / div)
4) return result based on 3rd parameter

FAQ'S

1) Explain Functions in JavaScript?
- ✓ Particular business logic called as Function
- ✓ Functions are used to reuse business logic
- ✓ Types of functions
  1) Function declaration / named functions
  2) Function Expression / anonymous functions
  3) Arrow Functions

2) what are the differences between function declaration and function expression?

Function Declaration	**Function Expression**
The function with particular name called as Function Declaration	function without name called as function expression

Syntax
Function Declaration
function functionname (param1….)
{
//business logic
}
Function Calling
functionname(arg1,…)
	Syntax
	Function Expression
	let variablename=function(param1,…) {
	//business logic
	}
	Function Calling
	variablename(arg1,…)

Function Declarations are Hoisted (Loaded in Creation Phase with Function Definition) (Fully Hoisted)	Function Expressions are not Hoisted (Wont load in creation phase) (Whenever interpreter reached then only loads into browser memory)

function declarations in callbacks, behaves like global functions
Ex.
let arr1= [100,200,300,400,500];
arr1.filter(func_one);
function func_one (element, index)
{
//business logic
}
where func_one is global function
	function expression in callbacks, not behaves like global functions
	Ex.
	let arr1= [100,200,300,400,500];
	arr1.filter(function (element, index) {
	//business logic
	});

Function declarations are not suggested for IIFE	Function expressions are suggested for IIFE
Function declarations are not secured	Function expressions are secured

3) Explain Arrow Functions?

- ✓ Arrow functions are introduced in ES6 version
- ✓ We will represent Arrow functions with “=>” symbol
- ✓ Arrow functions reduces code size
- ✓ return statement is optional for single line function
  Ex.
  let arrow_func= () =>`ExcelR`;
- ✓ functional braces “{}” are optional for single line statement
  Ex.
  let arrow_func= () =>document. write (`ExcelR`);
- ✓ Arrow functions binds the lexical context
  Example:

```
let obj={
        name: ‘ExcelR`,outer_func:
        function () {
            inner_func = () => {
                document. write(this.name);
            }
            inner_func ();
        }
    };
        obj. outer_func (); //ExcelR
```

4) Explain IIFE?

- ✓ IIFE stands for Immediately Invoked Function Expression
- ✓ IIFE is a JavaScript function that runs as soon as it is defined

Syntax:

```
(function () {
   //business logic
})()
```

5) differences Between Spread Operator and Rest Parameter?

- ✓ Both Syntax’s are same (…)
- ✓ Spread Operator Expands an Array/Object into its Elements

Ex1.

```
let arr1= [10,20,30,40,50];
let [a, b, c, d, e] =arr1;
```

Ex2:

```
let obj= {key1: `ExcelR`, key2: `Technologies`};
let {key1, key2} =obj;
```

- ✓ Rest Syntax will take multiple elements and condenses into single element

Ex1.

```
function func_one (…rest) {
   document.write(rest);
}
func_one (10,20,30,40,50);
```

6) what is lexical scope?

- ✓ a variable defined outside a function can be accessible inside another function defined after the variable declaration
- ✓ reverse not **possible**