HTML

HTML5

HTML stands for Hypertext Markup language.
It is widely used language on web to develop web pages and web applications.
It is developed by Burners Lee in late 1991.
It is a case insensitive language.
It is a tag based language.

ex: <html>

Every tag contains opening tag and closing tag.

ex: <html> -- opening tag
</html> -- closing tag

Every tag contains attributes and each attribute contains attribute name and attribute value.

ex: <body bgcolor="red">

Tag may contains multiple attributes and each attribute seperated with space.

ex: <body bgcolor="red" background="image">

HTML file is know as HTML document.
All HTML document we need to save with .html or .htm extension.
All HTML documents execute in a browser window.

ex: Chrome
Mozilla
Edge
Opera
IE and etc.

We will use following editors to write HTML code.

ex: Notepad
Notepad++
Sublime
VSC
Dreamviewer and etc.

A simple HTML document is also known as component.
HTML is used to develop client side components.
HTML 2.0 was released in the year of 1995.
HTML 4.0 was released in the year of 1999.
HTML5 which is a extension of HTML4 was released in 2012.
The main objective of HTML5 is used to develop light weight component.
HTML5 is also known as advanced hypertext markup language.

HTML skeleton

HTML

```
<!DOCTYPE>
<html>
    <head>
        -
        - // head related tags
        -
    </head>
    <body>
        -
        - // body related tags
        -
```

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

HTML5

```
<!DOCTYPE html>
<html>
    <head>
        -
        - // head related tags
        -
    </head>
    <body>
        -
        - // body related tags
        -
    </body>
</html>
```

Note: <!DOCTYPE> represent HTML document.
<!DOCTYPE html> represent HTML5 document.

<html> A <html> tag is a root tag for entire HTML document.
<head> A <head> tag is used to declare following things.
ex: title of a web page
favicon of a web page
styles
scripts
<body> A <body> tag is used to declare actual content of a web page.
ex: images
forms
tables
lists and etc.

First HTML Document

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        Welcome to HTML classes
    </body>
</html>
```

Q) How can we add title to a web page?
To add title of a web page we need to use <title> tag.

ex:
```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
</head>
<body>Welcome to HTML classes</body>
</html>
```

Q) How to add favicon to a web page?

A <link> tag is used to add favicon on a web page.

ex:
```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
    <link rel="icon" href="ihub.jpg">
</head>
<body>
    Welcome to HTML classes
</body>
</html>
```

Q) How to change the background color in html?

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
    <link rel="icon" href="ihub.jpg">
</head>
<body bgcolor="yellow">Welcome to HTML classes</body>
</html>
```

Q) How to set the background image in html?

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>
    <link rel="icon" href="ihub.jpg">
</head>
<body background="bg.jpg">

</body>
</html>
```

HTML Meta Tags

A meta tag is used to declare metadata of a document.
Here metadata means data of a data.
Metadata used by web browser, search engine and other web services.
To declare metadata we need to use <meta> tag.
A <meta> tag we can declare inside <head> tag.
A <meta> tag is used to declare following things.

1) Description
2) Author
3) keywords
4) view port
5) refresh
6) copyright

and etc.

UTF-8
UTF stands for Unicode Transformation Format.
UTF is a encloding methods which describes what character set a website is written with.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

		<!-- meta data -->
		<meta charset="utf-8">
		<meta name="description" content="Web page for learning purpose">
		<meta name="author" content="Niyaz sir">
		<meta name="keywords" content="HTML CSS JAVASCRIPT BOOSTRAP">
		<meta name="viewport" content="width=device-width, intial-scale=1.0">
		<meta http-equiv="refresh" content="05">
		<meta name="copyright" content="Copy right by Niyaz sir">
	</head>
	<body>
		This is HTML class
	</body>
</html>
```

Basic Tags in HTML
Heading Tags
It will display the text in bold and size of the text is depends upon number of heading tags.
We have 6 heading tags from h1 to h6.
ex:

```
<!DOCTYPE html>
<html>
<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">
</head>
<body>
		<h1> Heading Tag 1</h1>
		<h2> Heading Tag 2</h2>
		<h3> Heading Tag 3</h3>
		<h4> Heading Tag 4</h4>
		<h5> Heading Tag 5</h5>
		<h6> Heading Tag 6</h6>
</body>
</html>
```

Pragraph tag

A <p> tag is used to display the text in paragraph.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
	<body>
		<p>
			HTML is widely used language on web to develop
			web pages and web applications and it is developed by
			burners lee in late 1991.
		</p>
	</body>
</html>
```

center tag

A tag is used to display the text in center.

HTML5 does not support tag.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
	<body>
		<center>This is center tag</center>
	</body>
</html>
```

bold tag

A <b> tag is used to display the text in bold without importance.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>
```

```
		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
	<body>
		<b>This is bold tag</b>
	</body>
</html>
```

strong tag

A tag is used to display the text in bold with important.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">
	</head>
	<body>
		<strong>This is strong tag</strong>
	</body>
</html>
```

italic tag

A <i> tag is used to display the text in italic without force.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
	<body>
		<i>This is italic tag</i>
	</body>
</html>
```

emphasize tag

A tag is used to display the text in italic with force.

ex:

```
<!DOCTYPE html>
<html>
	<head>
```

```
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
	<body>
		<em>This is emphasize tag</em>
	</body>
</html>
```

underline tag

A <u> tag is used to display the text in underline.
HTML5 does not support <u> tag.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">
	</head>
	<body>
		<u>This is underline tag</u>
	</body>
</html>
```

font tag

A font tag is used to apply the color to the text , to increase the font size and change font family.
HTML 5 does not support <font> tag.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">
	</head>
	<body>
		<font color="blue" size="30" face="cursive">
			This is font tag
		</font>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">
	</head>
	<body>
		<font color="blue" size="30" face="monospace">
			This is font tag
		</font>
	</body>
</html>
```

breakline tag

A <br> tag is used to break the line in a web page.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
	<body>
		<i>This is italic tag</i>
		<br>
		<b>This is bold tag</b>
	</body>
</html>
```

Horizontal line

A <hr> tag is used to display horizontal line in a web page.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>

		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">

	</head>
```

```
	<body>
		<i>This is italic tag</i>
		<hr>
		<b>This is bold tag</b>
	</body>
</html>
```

marquee tag

A <marquee> tag is used to scroll the text.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<!--   title -->
		<title>MyPage!</title>
		<!-- favicon -->
		<link rel="icon" href="ihub.jpg">
	</head>
	<body>
		<marquee> This is HTML class for Freshers </marquee>
	</body>
</html>
```

Nested Tags in HTML

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<center>
			<h1>
				<font color="blue">Welcome to HTML classes</font>
			</h1>
		</center>
	</body>
</html>
```

Phrase tags in HTML

Phrase tags are special purpose tags which are used to defined structural meaning to block of code or text.

We have following list phrase tags in html.

1) abbreviation - <abr>
2) definition - <dfn>
3) short quote - <q>
4) address - <address>
5) code - <code>
6) keyboard - <kbd>
7) strike - <strike> or <s>

and etc.

1) abbreviation - <abr>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<abr title="HyperText Markup Language">
				HTML
		</abr>
		is a markup language and it is a case insensitive language.

	</body>
</html>
```

2) definition - <dfn>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<dfn>HTML</dfn>
		is a markup language and it is a case insensitive language.
	</body>
</html>
```

3) short quote - <q>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<q>I love html coding because it is easy</q>
	</body>
</html>
```

4) address - <address>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<address>
			1-4-78/1 , Nilgiri Block, Ameerpet,Hyderabad.
		</address>
	</body>
</html>
```

5) code - <code>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<code>
			void main()
			{
				clrscr();
				printf("Hello");
				getch();
			}
		</code>
	</body>
</html>
```

6) keyboard - <kbd>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<p> To copy press CTRL+C button </p>

		<p> To copy press <kbd>CTRL+C</kbd> button </p>
	</body>
</html>
```

7) strike - <strike> or <s>

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>
			<s>
				This is Java Class
			</s>
		</h1>
	</body>
</html>
```

HTML images

A <img> tag is used to display the images in a web page.

It is a opening tag with attributes and does not have any closing tag.

A <img> tag contains following attributes.

1) src - It is used to locate a file.
2) alt - It will display alternate message if image is not found.
3) width - It is used to set the width to the image.
4) height - It is used to set the height to the image.

We have following list of images.

Format	Abbreviation
JPEG	Joint Photographic Expert Group
PNG	Portable Network Graphics
SVG	Scalable Vector Graphics
GIF	Graphical Interchange Format and etc.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<img src="images/rock.jpg" />
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<img src="images/brock.jpg" alt="No Image Found"/>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<img src="images/rock.jpg"
				alt="No Image Found"
					width="200px"
					  height="200px" />
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<img 
src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTQkvL0Ec1NgCsW2Uk6SFeBkfygCxb6AKtU-g&usqp=CAU"

			alt="No Image Found"
				width="400px"
				  height="400px" />
	</body>
</html>
```

Hyperlink in HTML

A <a> anchor tag is used to display hyperlink in a web page.
Anchor tag contains "href" attribute to navigate to other resources.
ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>
			<a href="http://www.facebook.com/login"> Facebook </a>
		</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>
			<a href="http://www.facebook.com/login"
					target="_self">
				Facebook
			</a>

		</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>
			<a href="http://www.facebook.com/login"
					target="_blank">
				Facebook
			</a>

		</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<center>
			<a href="https://en.wikipedia.org/wiki/Dwayne_Johnson"
					target="_blank">

				<img src="images/rock.jpg"
					alt="no image found"
					width="300px"
					height="300px" />
			</a>
		</center>
	</body>
</html>
```

HTML Entity

HTML entity is a groupd of characters starts with '&' and ends with semicolon(;).
HTML entities are used to display reversed characters or hidden characters in html.
We have following list of html entities.

ex:

HTML entity	character
>	>
<	<
«	<<
»	>>
	(1 space)
©	
and etc.	

Q) What is difference between HTML tag and HTML element?

HTML tag

HTML tag starts with '<' symbol and ends with '>' symbol.

ex: html>,<head>,<body>,<h1> and etc.

HTML element

HTML element defines opening tag, some content and closing tag.

ex: <h1> This is heading tag </h1>
 This is paragraph tag

Q) What is the different between block elements and inline elements?

block elements

A block elements always start with new line.

They will occupy 100% of width.

ex: <h1> , , and etc

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1> This is heading tag </h1>
		<p> This is paragraph tag </p>
		<div> This is division tag </div>
	</body>
</html>
```

inline elements

Inline elements starts with same line.

They will occupy width as much as required.

ex: <b>,<i>, and etc.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<b>This is bold tag</b>
		<i>This is italic tag</i>
		<span>This is span tag</span>
	</body>
</html>
```

Q) Types of tags in HTML ?

We have two types of tags in HTML.

1) Paired Tags / Container Tags

Paired tag contains opening tag and closing tag.

ex: <html>, <head>, <body> , <title> and etc.

2) Unpaired Tags / Empty Tags

Unpaired tag contains only opening and does not have any closing tag.

ex:
 , <hr>, <img> , <link> and etc.

Q) Types of list in HTML ?

We have three types of list in HTML.

1) Ordered list
2) Unordered list
3) Description list

Ordered list

A <ol> tag is used to represent ordered list with numeric and alphabet.

Order list contains list of items.

Each list of item we can represent by using <li> tag.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ol>
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
			<li>ExpressJS</li>
			<li>NodeJS</li>
			<li>NextJS</li>
		</ol>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ol start="101">
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
			<li>ExpressJS</li>
			<li>NodeJS</li>
			<li>NextJS</li>
		</ol>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ol type="a">
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
			<li>ExpressJS</li>
			<li>NodeJS</li>
			<li>NextJS</li>
		</ol>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ol type="A">
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
			<li>ExpressJS</li>
			<li>NodeJS</li>
			<li>NextJS</li>
		</ol>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ol type="i">
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
```

```
                <li>ExpressJS</li>
                <li>NodeJS</li>
                <li>NextJS</li>
            </ol>
        </body>
</html>
```

Unordered list

A <ul> tag is used to represent unordered list with bullets.
Unorder list contains list of items.
Each list of item we can represent by using <li> tag.

ex:

```
<!DOCTYPE html>
<html>
        <head>
            <title>MyPage!</title>
        </head>
        <body>
            Courses:
            <ul type="disc">
                <li>ReactJS</li>
                <li>AngularJS</li>
                <li>VueJS</li>
                <li>ExpressJS</li>
                <li>NodeJS</li>
                <li>NextJS</li>
            </ul>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
            <title>MyPage!</title>
        </head>
        <body>
            Courses:
            <ul type="circle">
                <li>ReactJS</li>
                <li>AngularJS</li>
                <li>VueJS</li>
                <li>ExpressJS</li>
                <li>NodeJS</li>
                <li>NextJS</li>
            </ul>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ul type="square">
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
			<li>ExpressJS</li>
			<li>NodeJS</li>
			<li>NextJS</li>
		</ul>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Courses:
		<ul type="none">
			<li>ReactJS</li>
			<li>AngularJS</li>
			<li>VueJS</li>
			<li>ExpressJS</li>
			<li>NodeJS</li>
			<li>NextJS</li>
		</ul>
	</body>
</html>
```

Description list

A <dl> tag is used to represent description list.
A description list contains description term and description definition.
A <dt> tag is used to represent description term.
A <dd> tag is used to represent description definition.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
```

```
            <dl>
                <dt>HTML5</dt>
                <dd>
                    HTML5 is a advanced hypertext markup language
                    which is used to develop light weight components
                    to design interactive static web pages and web
                    applications.
                </dd>
            </dl>
        </body>
</html>
```

HTML Forms

HTML forms are used to accept the values from the user.
It will send the data to database or server for processing.
To create a form in html we need to use <form> tag.
We have following list of form components.

ex: Label, TextField, Buttons, Checkbox, RadioButton, textarea, select box, and etc.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
    </head>
    <body>
        <form>
            <label>Name:</label>
            <input type="text" name="t1"/>
            <br>
            <label>Password:</label>
            <input type="password" name="t2"/>
            <br>
            <label>Age:</label>
            <input type="number" name="t3"/>
            <br>
            <label>DOB:</label>
            <input type="date" name="t4"/>
            <br>
            <label>Gender:</label>
            <input type="radio" value="male" name="t5"/>MALE
            <input type="radio" value="female" name="t5"/>FEMALE
            <br>
            <label>Maritial Status:</label>
            <input type="checkbox" value="married" name="t6"/>MARRIED
            <input type="checkbox" value="single" name="t6"/>SINGLE
            <br>
            <label>Qualification:</label>
            <select name="t7">
                <option value="">none</option>
```

```
                        <option value="b.tech">B.Tech</option>
                        <option value="b.sc">B.SC</option>
                        <option value="b.com">B.COM</option>
                    </select>
                    <br>
                    <label> Address : <label>
                    <textarea name="t8" rows="5" cols="10"></textarea>
                    <br>
                    <input type="reset" value="reset"/>
                    <input type="submit" value="submit"/>
                </form>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                <form>
                        <label>Name:</label>
                        <input type="text" name="t1" autocomplete="off" required/>
                        <br>
                        <label>Password:</label>
                        <input type="password" name="t2" required/>
                        <br>
                        <label>Age:</label>
                        <input type="number" name="t3" required/>
                        <br>
                        <label>DOB:</label>
                        <input type="date" name="t4" required/>
                        <br>
                        <label>Gender:</label>
                        <input type="radio" value="male" name="t5"/>MALE
                        <input type="radio" value="female" name="t5"/>FEMALE
                        <br>
                        <label>Maritial Status:</label>
                        <input type="checkbox" value="married" name="t6"/>MARRIED
                        <input type="checkbox" value="single" name="t6"/>SINGLE
                        <br>
                        <label>Qualification:</label>
                        <select name="t7">
                                <option value="">none</option>
                                <option value="b.tech">B.Tech</option>
                                <option value="b.sc">B.SC</option>
                                <option value="b.com">B.COM</option>
                        </select>
                        <br>
```

```
                        <label> Address : <label>
                        <textarea name="t8" rows="5" cols="10"></textarea>
                        <br>
                        <input type="reset" value="reset"/>
                        <input type="submit" value="submit"/>
                </form>
        </body>
</html>
```

ex:1

index.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                <form name="myform" action="a.html" method="POST">
                        <label>Name:</label>
                        <input type="text" name="t1" autocomplete="off" required/>
                        <br>
                        <label>Password:</label>
                        <input type="password" name="t2" required/>
                        <br>
                        <label>Age:</label>
                        <input type="number" name="t3" required/>
                        <br>
                        <label>DOB:</label>
                        <input type="date" name="t4" required/>
                        <br>
                        <label>Gender:</label>
                        <input type="radio" value="male" name="t5"/>MALE
                        <input type="radio" value="female" name="t5"/>FEMALE
                        <br>
                        <label>Maritial Status:</label>
                        <input type="checkbox" value="married" name="t6"/>MARRIED
                        <input type="checkbox" value="single" name="t6"/>SINGLE
                        <br>
                        <label>Qualification:</label>
                        <select name="t7">
                                <option value="">none</option>
                                <option value="b.tech">B.Tech</option>
                                <option value="b.sc">B.SC</option>
                                <option value="b.com">B.COM</option>
                        </select>
                        <br>
                        <label> Address : <label>
                        <textarea name="t8" rows="5" cols="10"></textarea>
                        <br>
                        <input type="reset" value="reset"/>
```

```
                    <input type="submit" value="submit"/>
                </form>
        </body>
</html>
```

a.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>A.html</title>
        </head>
        <body bgcolor="yellow">

                <center>welcome to A.html document </center>
        </body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                <form name="myform" action="#" method="GET">
                <input type="text" name="t1" autocomplete="off" placeholder="username"/>
<br>
                <input type="password" name="t2" placeholder="password"/> <br>
                <input type="submit" value="Login"/>
                </form>
        </body>
</html>
```

HTML Table

A table is used to store the data in rows and columns.
A <table> tag is used to represent table in html.
A <tr> tag is used represent table row.
A <th> tag is used represent table heading.
A <td> tag is used to represent table data.
ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                <table border="0">
                        <tr>
                                <th>NO</th>
                                <th>NAME</th>
```

```
                        <th>ADDRESS</th>
                    </tr>
                    <tr>
                        <td>101</td>
                        <td>Alan</td>
                        <td>Texas</td>
                    </tr>
                    <tr>
                        <td>102</td>
                        <td>Jose</td>
                        <td>Chicago</td>
                    </tr>
                    <tr>
                        <td>103</td>
                        <td>Jason</td>
                        <td>Florida</td>
                    </tr>
                </table>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                <table border="1" align="center"  width="100%" bgcolor="cyan">
                <caption>Employee Details</caption>
                        <tr>
                                <th>NO</th>
                                <th>NAME</th>
                                <th>ADDRESS</th>
                        </tr>
                        <tr>
                                <td>101</td>
                                <td>Alan</td>
                                <td>Texas</td>
                        </tr>
                        <tr>
                                <td>102</td>
                                <td>Jose</td>
                                <td>Chicago</td>
                        </tr>
                        <tr>
                                <td>103</td>
                                <td>Jason</td>
                                <td>Florida</td>
                        </tr>
```

```
		</table>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<table border="1" align="center" cellspacing="10px" cellpadding="10px">
		<caption>Employee Details</caption>
			<tr bgcolor="cyan">
				<th>NO</th>
				<th>NAME</th>
				<th>ADDRESS</th>
			</tr>
			<tr>
				<td>101</td>
				<td>Alan</td>
				<td>Texas</td>
			</tr>
			<tr>
				<td>102</td>
				<td>Jose</td>
				<td>Chicago</td>
			</tr>
			<tr>
				<td>103</td>
				<td>Jason</td>
				<td>Florida</td>
			</tr>
		</table>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<table border="1" align="center">
		<caption>Employee Details</caption>
			<tr bgcolor="cyan">
				<th>NO</th>
				<th>NAME</th>
```

```
                <th>ADDRESS</th>
            </tr>
            <tr>
                <td>101</td>
                <td>Alan</td>
                <td>Texas</td>
            </tr>
            <tr>
                <td>102</td>
                <td>Jose</td>
                <td>Chicago</td>
            </tr>
            <tr>
                <td>103</td>
                <td>Jason</td>
                <td>Florida</td>
            </tr>
            <tr>
                <td colspan="3">Thank you</td>
            </tr>
        </table>
    </body>
</html>
```

HTML form using Table

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
    </head>
    <body>
        <form name="myform" action="#" method="POST">
            <table align="center">
                <tr>
                    <td><label>UserName:</label></td>
            <td><input type="text" name="t1" autocomplete="off"
required/></td>
                </tr>
                <tr>
                    <td><label>Password:</td>
                    <td><input type="password" name="t2"
required/></td>
                </tr>
                <tr>
                    <td><label>Qualitification:</label></td>
                    <td>
                        <select name="t3">
                            <option value="">none</option>
                            <option
value="btech">B.TECH</option>
```

```
                                        <option value="bsc">B.SC</option>
                                        <option
value="bcom">B.COM</option>
                                    </select>
                                </td>
                            </tr>
                            <tr>
                                <td><input type="reset" value="reset"/></td>
                                <td><input type="submit" value="submit"/></td>
                            </tr>
                        </table>
                </form>
        </body>
</html>
```

Datalist tag

A <datalist> tag/element is used to specify list of predefine options for an <input> tag/element.
A <datalist> tag/element provides autocomplete features for an <input> element/tag.
User will see a drop-down list of predefine options for an <input> tag/element.
A <datalist> tag/element "id" attribute must be same as <input> tag/element "list" attribute.
ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                Select Country : <input type="text" list="countries"/>

                <datalist id="countries">
                        <option value="India">
                        <option value="Ireland">
                        <option value="Iran">
                        <option value="Iraq">
                        <option value="Iceland">
                </datalist>
        </body>
</html>
```

<details> and <summary> tag

A <details> tag/element is used display special content where a user can open and close on demand.
A <details> tag is used to design interactive widgets where user can open and close.
A <details> tag contains <summary> tag.
We can keep any sort of tags inside <details> tags.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<details>
			<summary>HTML</summary>
			<h1>HTML is a markup language</h1>
			<p>It is a case insensitive language</p>
			<div>It is a tag based language</div>
		</details>
	</body>
</html>
```

<big> tag

The <big> tag is used to make the text one size bigger i.e from small to medium, medium to large, large to x-large.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<p>This is paragraph tag</p>

		<p>This is <big>paragraph</big> tag</p>
	</body>
</html>
```

<small> tag

The <small> HTML element represents small print like copyright and legal text.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<small>&copy;AllRight Reserved-2023</small>
	</body>
</html>
```

<cite> tag

The <cite> tag defines the title of a creative work i.e movie , songs, Poems and etc.

The text in the <cite> element usually renders in italic.

ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		Movie Title : <cite>Salaar</cite>
	</body>
</html>
```

<mark>
The HTML <mark> tag is used to mark or highlight text that has special interest.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>This is is <mark>very important</mark> for practice</h1>
	</body>
</html>
```

<sub> tag
The <sub> tag defines subscript text.
Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>H<sub>2</sub>O</h1>
	</body>
</html>
```

<sup> tag
The <sup> tag defines superscript text.
Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1>x<sup>2</sup>+y<sup>2</sup></h1>
	</body>
```

```
</html>
```

<bdo> tag

BDO stands for Bi-Directional Override.

The <bdo> tag is used to override the current text direction.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<bdo dir="rtl"> Hello </bdo>
	</body>
</html>
```

HTML <header> and <nav> tag

The <header> element represents a container for introductory content or a set of navigational links.

The <nav> tag defines a set of navigation links.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<header style="background-color:cyan">
			<center>
			<nav>
		<a		href=""		style="text-decoration:none">		HOME
</a>    
		<a		href=""		style="text-decoration:none">		ABOUTUS
</a>    
		<a		href=""		style="text-decoration:none">		SERVICES
</a>    
		<a		href=""		style="text-decoration:none">		PORTFOLIO
</a>    
		<a href="" style="text-decoration:none"> CONTACT </a>
			</nav>
			</center>
		</header>
	</body>
</html>
```

HTML <section>, <article> and <figure> tag

The <section> tag defines a section in a document.

The <article> tag specifies independent, self-contained content.

The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<section>
			<article>
				<center>
					<h1>Hollywood Actor</h1>

					<figure>
				<img src="images/rock.jpg" width="200px" height="200px"/>
					<figcaption>Dwayne Johnson</figcation>
					</figure>
					<p>
					Dwayne Douglas Johnson (born May 2, 1972), also known by his ring name the Rock, is an American actor, film producer, and retired professional wrestler. Widely regarded as one of the greatest professional wrestlers of all time,[6][7] he was integral to the development and success of the World Wrestling Federation (WWF, now WWE) during the Attitude Era, an industry boom period in the late 1990s and early 2000s. Johnson wrestled for the WWF for eight years before pursuing an acting career. His films have grossed over $3.5 billion in North America and over $10.5 billion worldwide,[8] making him one of the world's highest-grossing and highest-paid actors.
					</p>
				</center>
			</article>
		</section>
	</body>
</html>
```

HTML <footer> tag

The <footer> tag defines a footer for a document or section.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<footer>
			<center>
			<small>&copy;All Right reserved - 2023</small>
			</center>
		</footer>
	</body>
</html>
```

Notepad++ Editor Download link : https://notepad-plus-plus.org/downloads/

HTML Frames

To use frames on a web page we need to use <frameset> tag instead of <body> tag.
The <frameset> tag defines how to divide the window into frames.
The "rows" attribute of <frameset> tag defines horizontal frames.
The "cols" attribute of <frameset> tag defines vertical frames.
Each frame is indicated by <frame> tag and it defines which document should be open into that place.

ex:1

index.html

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<frameset cols="33%,*,33%">
		<frame src="a.html"/>
		<frame src="b.html"/>
		<frame src="c.html"/>
	</frameset>
</html>
```

a.html

```
<!DOCTYPE html>
<html>
	<head>
		<title>A.html</title>
	</head>
	<body bgcolor="red">

	</body>
</html>
```

b.html

```
<!DOCTYPE html>
<html>
	<head>
		<title>B.html</title>
	</head>
	<body bgcolor="green">

	</body>
</html>
```

c.html

```
<!DOCTYPE html>
<html>
	<head>
		<title>C.html</title>
	</head>
```

```
        <body bgcolor="blue"></body>
</html>
```

ex:2

index.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <frameset rows="33%,*,33%">
                <frame src="a.html"/>
                <frame src="b.html"/>
                <frame src="c.html"/>
        </frameset>
</html>
```

a.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>A.html</title>
        </head>
        <body bgcolor="orange"></body>
</html>
```

b.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>B.html</title>
        </head>
        <body bgcolor="white">
                <center>
                        <img src="images/chakra.png" width="180px" height="180px"/>
                </center>
        </body>
</html>
```

c.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>C.html</title>
        </head>
        <body bgcolor="green">

        </body>
</html>
```

HTML <iframe> tag
It is used to specify inline frame.
A <iframe> tag/element is used to embed a document into current HTML document.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
			<iframe src="http://www.ihubtalent.com" width="400px" height="400px"/>
	</body>
</html>
```

Steps to display Google Map on a web page
step1: Goto Google Maps.
ex: https://www.google.com/maps
step2: Type Ihub Talent in Google Map Search Box.
step3: Click on "menu" button.
step4: Click on "share and embed Map"
step5: Click on "embeded a map" link.
step6: Click to "Copy Html".
step7: Paste the code inside <body> tag of index.html file.
step8: Check the output on browser window.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
			<iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3806.48765650213 1!2
d78.44202777415968!3d17.436358401394536!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3
m3!1m2!1s0x3bcb919633f78bef%3A0xbb63515db9bc2eba!2siHub%20Talent%20Info%20S
ystems%20-
%20Testing%20Tools%2C%20Full%20Stack%20Java%2C%20Python%20Training%20Insti
tute%20in%20Hyderabad!5e0!3m2!1sen!2sin!4v1694234739077!5m2!1sen!2sin"
			width="100%" height="450" style="border:0;" allowfullscreen=""
loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>
	</body>
</html>
```

HTML <audio> tag

The HTML <audio> tag/element is used to play an audio file on a web page.
<audio> tag contains "controls" attribute adds audio control like play,pause ,volume and etc.
The <source > tag/element allows us to specify alternate audio file which the browser my choose.
HTML audio formats can be MP3,WAV,OGG and etc.
HTML Audio Media types are

File Formats	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB</title>
	</head>
	<body>
			<audio controls>
				<source src="media/sajda.mp3" type="audio/mpeg">
			</audio>
	</body>
</html>
```

HTML <video> tag

The <video> tag or element is used to embed video content in a document such as movie clip, other video streams.
The <video> tag contains one or more <source> tag with different video source.
There are three supported video fromats in html are MP4,webM and OGG.
HTML video media types are

File format	Media Type
MP4	video/mp4
OGG	video/ogg

and etc.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB</title>
	</head>
	<body>
			<video controls width="200px" height="200px">
				<source src="media/video.mp4" type="video/mp4">
			</video>
	</body>
</html>
```

Graphics in HTML5

There are two ways to draw the graphics in HTML5.
1)SVG 2)CANVAS

1) SVG

HTML5 uses SVG technology to derived graphics in HTML.
SVG stands for Scalable Vector Graphics.
SVG is used to draw two-dimensional vector based graphics in HTML.
World Wide Web Consortium (W3C) prefers SVG technology to draw the graphics in HTML.
A <svg> tag/element is a container tag for vector graphics.
A <svg> tag/element contains various methods to display circle, rectangle, polygon, graphic images and etc.

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<svg style="border:2px solid black" width="300px" height="300px">
		</svg>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<svg style="border:2px solid black" width="300px" height="300px">
			<!-- circle -->
			<circle cx="150px" cy="150px" r="50px"/>
		</svg>
	</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<svg style="border:2px solid black" width="300px" height="300px">
			<!-- circle -->
			<circle cx="150px" cy="150px" r="50px"
			fill="#FFFF00"  stroke="#FF0000" stroke-width="5px"/>
		</svg>
```

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

ex:4

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<svg style="border:2px solid black" width="300px" height="300px">
			<!-- reactangle -->
			<rect x="50px"   y="20px"  width="200px" height="200px"
			fill="green" stroke="blue" stroke-width="5px"/>
		</svg>
	</body>
</html>
```

ex:5

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<svg style="border:2px solid black" width="300px" height="300px">
			<!--  polygon -->
			<polygon points="10,78 100,10 250,190 160,210"
			fill="yellow" stroke="red" stroke-width="5px"/>
		</svg>
	</body>
</html>
```

2) CANVAS

A <canvas> tag is used to draw vector graphics via javascript.
A <canvas> tag/element is a container tag.
A <canvas> tag/element contains various methods to display lines, circle, rectangle, polygon, graphic images and etc.

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<canvas  style="border:2px solid black;" width="300px" height="300px">
```

```
		</canvas>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<canvas id="myId" style="border:2px solid black;"
					width="300px"
					   height="300px">

		</canvas>

		<script>
			var context=document.getElementById("myId");
			var c=context.getContext("2d");
			c.moveTo(0,0);
			c.lineTo(400,400);
			c.stroke();
		</script>

	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<canvas id="myId" style="border:2px solid black;"
					width="300px"
					   height="300px">
		</canvas>
		<script>
			var context=document.getElementById("myId");
			var c=context.getContext("2d");
			c.beginPath();
			c.arc(100,100,50,0,2*Math.PI);
			c.stroke();
		</script>
	</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>INDEX</title>
	</head>
	<body>
		<!-- container -->
		<canvas id="myId" style="border:2px solid black;"
                                    width="300px"
                                       height="300px">

		</canvas>

		<script>
			var context=document.getElementById("myId");
			var c=context.getContext("2d");
			c.font = "30px Arial";
			c.fillText("Hello World", 20, 50);
		</script>

	</body>
</html>
```

Q) What is the difference between HTML and HTML5 ?

HTML	HTML5
To represent html document we need to Use <!DOCTYPE>.	To represent html5 document we need to use <!DOCTYPE html>.
HTML is bit slow.	HTML5 is more faster because it is light weight.
It is inflexible for the programmer.	It is flexible for the programmer.
It does not support drag and drop menu.	It supports drag and drop menu.
It is less mobile friendly.	It is more mobile friendly.
It does not allow to execute javascript.	It allows to execute javascript with help of JS Web work API.
We can't play audio and video without using flash player.	We can play audio and video with the help <audio> and <video> tag without using flash player.
Vector graphics is possible by using technologies VML, silver light , adope flash and etc.	Vector graphics is possible by using internal technologies like SVG and CANVAS.

It can't handle inaccurate errors.	It can handle inaccurate errors.
Shapes like circle, rechtangle , polygon and etc are not easy to draw.	Shapes like circle, rectangle, polygon and etc are easy to draw.

Q) What is the difference between tag and tag ?

div	span
It is a block element.	It is a inline element.
It is used to wrap the sections.	It is used to wrap the small portion of a text, images and etc.
It is used to develop css based layouts.	It is used to stylize the text.

Q) What is the disadvantages of HTML5?

1. It is used to develop static web pages but not dynamic web pages.
2. Security features are not good.
3. For simple design we need to do lot of code.
4. If code is increases then it will increase the complexity.

How do HTML document will execute in a browser window

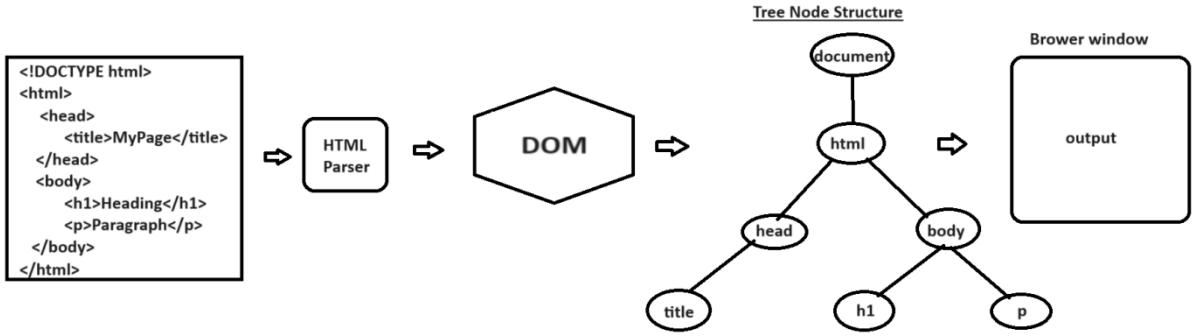

Diagram: class7.1

HTML parse will parse the data from HTML document and it will give to DOM.
DOM stands for Document Object Mode.
DOM is present in a browser window.
DOM will convert our HTML document to Tree Node structure.
Late, Tree Node structure will execute in a browser window.

CSS

CSS stands for Cascading Styles Sheet.
CSS is used to apply the styles on elements.
The latest version of CSS3 was introduced in 2001.
The main objective of CSS3 are.

1. To set the positioning of an element.
2. To apply the styles to describe how an element should look like.
3. To perform some sort of animations.

syntax:

```
<!DOCTYPE html>
<html>
	<head>
		<style>
			-
			-
			-
		</style>
	</head>
	<body>

	</body>
</html>
```

Note: Here style tag will cascade from head to body.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:#0000ff;
				background-color:#FFFF00;
			}

		</style>
	</head>
	<body>
			<h1>This is Heading Tag</h1>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
```

```
                {
                        background-color:#00FF00;
                }
                h1
                {
                        color:#0000ff;
                        font-size:50px;
                }

        </style>
    </head>
    <body>
                <h1>This is Heading Tag</h1>
    </body>
</html>
```

CSS syntax

CSS syntax contains selector and declaration block.

ex:

```
selector
|
h1{color:#0000ff;font-size:50px;}
  |----------------------------|
                |
        declaration block
```

CSS declaration block may contains multiple CSS properties.
Each css property contains property name and property value seperated with colon.
Each property ends with semicolon.

Advantages of CSS

1) It is easy to learn and easy to use.
2) It saves lot of development time.
3) It supported by all major browsers.
4) It supports global change.
5) Flexibility

Disadvantages of CSS

1) Fragmentation
2) Need to update all the versions of CSS.

Types of CSS

We have three types of CSS.

1)Inline CSS
2)Internal CSS / Embedded CSS
3)External CSS / Seperate CSS

1. Inline CSS

Inline CSS is used to apply unique styles on single element.
Using "style" attribute we can achieve inline CSS.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<h1 style="color:red;">This is Heading Tag1</h1>
		<h1 style="background-color:yellow;">This is Heading Tag 2</h1>
	</body>
</html>
```

2. Internal CSS

It is used to apply unique styles on single web page.
Using <style> tag we can achieve internal CSS.
A <style> tag we need to declare inside <head> tag.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style type="text/css">
			h1
			{
				color:blue;
				background-color:yellow;
			}
			p
			{
				font-size:50px;
				text-align:center;
			}
		</style>
	</head>
	<body>
		<h1>This is Heading Tag1</h1>
		<h1>This is Heading Tag2</h1>
		<p>This is paragraph tag</p>
	</body>
</html>
```

3. External CSS

It is used to apply the unique styles on entire web site (collection of web pages).
In external css, we need to create two files

i) .html file
ii) .css file

In external css , we will maintain html code in .html file and CSS code in .css file.
It is not possible to execute .css file seperately in a browser window.
In order to execute .css file we need to link css file with HTML file using <link> tag.
ex:

index.html

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<!-- add external css -->
		<link rel="stylesheet" href="mystyles.css">
	</head>
	<body>
		<h1> This is External CSS </h1>
	</body>
</html>
```

mystyles.css

```
body
{
	background-color:cyan;
}
h1
{
	color:blue;
	text-align:center;
}
```

CSS background property

CSS background property is used to set the background in a web page.
If we set the background to body then it will reflect to entire web page.
If we want we can apply background to elements also.
ex: <h1>, <p>, and etc.
We have following list of CSS background properties.

1) background-color
2) background-image
3) background-repeat
4) background-size
5) background-position
6) background-attachment
7) background shorthand
8) background-blend-mode

and etc.

1) background-color

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color:yellow;
			}
```

```
		</style>
	</head>
	<body>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				background-color:yellow;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag</h1>
	</body>
</html>
```

2) background-image

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				background-image: url("images/bg.jpg");
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<style>
			body
			{
				background-image: url("images/bg.jpg");
```

```
                }
            </style>
        </head>
        <body>
                <h1> This is Heading Tag</h1>
        </body>
</html>
```

3) background-repeat

```
<!DOCTYPE html>
<html>
        <head>
            <title>MyPage!</title>
            <style>
                body
                {
                    background-color:#FFFFFF;
                    background-image: url("images/bg.jpg");
                    background-repeat:no-repeat;
                }
            </style>
        </head>
        <body>
                <h1> This is Heading Tag</h1>
        </body>
</html>
```

4) background-size

```
<!DOCTYPE html>
<html>
        <head>
            <title>MyPage!</title>
            <style>
                body
                {
                    background-color:#FFFFFF;
                    background-image: url("images/bg.jpg");
                    background-repeat:no-repeat;
                    background-size: 1500px 1100px;
                }
            </style>
        </head>
        <body>
                <h1> This is Heading Tag</h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
```

```
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color:#FFFFFF;
				background-image: url("images/bg.jpg");
				background-repeat:no-repeat;
				background-size: cover;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag</h1>
	</body>
</html>
```

5) background-position

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color:#FFFFFF;
				background-image: url("images/bg.jpg");
				background-repeat:no-repeat;
				background-size: static;
				background-position: center 0px;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color:#FFFFFF;
				background-image: url("images/bg.jpg");
				background-repeat:no-repeat;
				background-size: static;
```

```
                        background-position: right 0px;
                }
        </style>
    </head>
    <body>
                <h1> This is Heading Tag</h1>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            body
            {
                background-color:#FFFFFF;
                background-image: url("images/bg.jpg");
                background-repeat:no-repeat;
                background-size: static;
                background-position: left 0px;
            }
        </style>
    </head>
    <body>
            <h1> This is Heading Tag</h1>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            body
            {
                background-color:#FFFFFF;
                background-image: url("images/bg.jpg");
                background-repeat:no-repeat;
                background-size: static;
                background-position: center 200px;
            }
        </style>
    </head>
    <body>
            <h1> This is Heading Tag</h1>
    </body>
</html>
```

6) background-attachment

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color:#FFFFFF;
				background-image: url("images/bg.jpg");
				background-repeat:no-repeat;
				background-size: static;
				background-position: center 200px;
				background-attachment:scroll;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
```

```
        <style>
            body
            {
                background-color:#FFFFFF;
                background-image: url("images/bg.jpg");
                background-repeat:no-repeat;
                background-size: static;
                background-position: center 200px;
                background-attachment:fixed;
            }
        </style>

    </head>
    <body>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
            <h1> This is Heading Tag</h1>
    </body>
</html>
```

7) background shorthand

```
background-color:#FFFFFF;
background-image: url("images/bg.jpg");
background-repeat:no-repeat;
background-position: center 200px;
background-attachment:fixed;
```

or

```
background: #FFFFFF url("images/bg.jpg") no-repeat center 200px fixed;
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background:  #FFFFFF  url("images/bg.jpg")  no-repeat  center 
200px fixed;
				background-size: static;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
			<h1> This is Heading Tag</h1>
	</body>
</html>
```

8) background-blend-mode

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<style>
			body
			{
				background-color:#FFFFFF, #FFFFFF;
```

```
                    background-image:                    url("images/bg.jpg"),
url("images/micky.png");
                    background-repeat:no-repeat, no-repeat ;
                    background-size: cover , 400px;
                    background-position: center 0px , center 100px;
                    background-attachment:fixed , fixed;
                    background-blend-mode: lighten;
                }
            </style>

        </head>
        <body>

        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>

                <style>
                        body
                        {
                                background-color:#FFFFFF, #FFFFFF;
                                background-image:                    url("images/bg.jpg"),
url("images/micky.png");
                                background-repeat:no-repeat, no-repeat ;
                                background-size: cover , 400px;
                                background-position: center 0px , center 100px;
                                background-attachment:fixed , fixed;
                                background-blend-mode: darken;
                        }
                </style>

        </head>
        <body>

        </body>
</html>
```

CSS border property

CSS border property is used to add the border to elements.
We have following list of CSS border properties.

1) border-style
2) border-color
3) border-width
4) border shorthand

1) border-style

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border-style: solid;
			}
			h2
			{
				border-style: dotted;
			}
			h3
			{
				border-style: dashed;
			}
			p
			{
				border-style:double;
			}
			div
			{
				border-style:inset;
			}
		</style>
	</head>
	<body>
			<h1> Heading Tag 1 </h1>

			<h2> Heading Tag 2 </h2>

			<h3> Heading Tag 3 </h3>

			<p> This is paragraph </p>

			<div> This is Division tag </div>
	</body>
</html>
```

2) border-color

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border-style: solid;
```

```
                        border-color: red;
                }
                h2
                {
                        border-style: dotted;
                        border-color:blue;
                }
                h3
                {
                        border-style: dashed;
                        border-color:green;
                }
                p
                {
                        border-style:double;
                        border-color:yellow;
                }
                div
                {
                        border-style:inset;
                        border-color:pink;
                }
        </style>
    </head>
    <body>
                <h1> Heading Tag 1 </h1>

                <h2> Heading Tag 2 </h2>

                <h3> Heading Tag 3 </h3>

                <p> This is paragraph </p>

                <div> This is Division tag </div>
    </body>
</html>
```

3) border-width

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                h1
                {
                        border-style: solid;
                        border-color: red;
                        border-width: 5px;
                }
                h2
```

```
                {
                        border-style: dotted;
                        border-color:blue;
                        border-width: 10px;
                }
                h3
                {
                        border-style: dashed;
                        border-color:green;
                        border-width: 15px;
                }
                p
                {
                        border-style:double;
                        border-color:yellow;
                        border-width: 20px;
                }
                div
                {
                        border-style:inset;
                        border-color:pink;
                        border-width: 25px;
                }
        </style>
    </head>
    <body>
                <h1> Heading Tag 1 </h1>

                <h2> Heading Tag 2 </h2>

                <h3> Heading Tag 3 </h3>

                <p> This is paragraph </p>

                <div> This is Division tag </div>
    </body>
</html>
```

4) border shorthand

```
border-width : 2px;
border-style : solid;
border-color : blue;
```

or

```
border : 2px solid blue;
```

ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border : 2px solid red;
			}
			h2
			{
				border : 2px dotted blue;
			}
			h3
			{
				border : 2px dashed green;
			}
			p
			{
				border : 2px double yellow;
			}
			div
			{
				border : 2px inset pink;
			}
		</style>
	</head>
	<body>
			<h1> Heading Tag 1 </h1>

			<h2> Heading Tag 2 </h2>

			<h3> Heading Tag 3 </h3>

			<p> This is paragraph </p>

			<div> This is Division tag </div>
	</body>
</html>
```

CSS Colors

In CSS , we can declare colors in following ways.

1) valid color name
2) Hexa value
3) RGB value
4) HSL value

1) valid color name

```
<!DOCTYPE html>
<html>
	<head>
```

```
		<title>MyPage!</title>
		<style>
			body
			{
				background-color: violet;
			}
			h1
			{
				color:forestgreen;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>
```

2) Hexa value

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color: #FFFF00;
			}
			h1
			{
				color:#00ff00;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>
```

3) RGB value

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				background-color: rgb(255,0,0);
			}
			h1
			{
```

```
                color:rgb(0,0,255);
            }
        </style>
    </head>
    <body>
            <h1> This is Heading Tag </h1>
    </body>
</html>
```

4) HSL value

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            body
            {
                background-color: hsl(330, 100%, 60%);
            }
            h1
            {
                color: hsl(0, 100%, 40%);
            }
        </style>
    </head>
    <body>
            <h1> This is Heading Tag </h1>
    </body>
</html>
```

CSS margin property

CSS margin property is used to give the space around the elements, outside the defined border.

CSS margin contains following properties.

1) margin-top
2) margin-right
3) margin-bottom
4) margin-left
5) margin shorthand

All the above margin properties will take following values.

i) auto : It declares the margin by the browser.
ii) value : It declares the margin in px,pt,cm,em and etc.
iii) % : It declares the margin in percentage.
iv) inherit : It take the margin from parent tag.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
```

```
		<style>
			h1
			{
				border:2px solid #000;
				margin-top:100px;
				margin-right:100px;
				margin-bottom:100px;
				margin-left:100px;
			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>

ex:
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid #000;

				margin-top:10%;
				margin-left:20%;
				margin-right:20%;
				margin-bottom:10%;

			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>

ex:
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				border:2px solid #FF0000;
```

```
                        margin:50px;
                }
                h1
                {
                        border:2px solid #000;
                        margin:inherit;
                }

        </style>
    </head>
    <body>
        <div>
            <h1> This is Heading Tag </h1>
        </div>
    </body>
</html>
```

If margin contains four values.
ex:

margin : 25px 50px 100px 150px;

top , right, bottom , left

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            h1
            {
                border:2px solid #000;

                margin: 25px 50px 100px 150px;
            }

        </style>
    </head>
    <body>
            <h1> This is Heading Tag </h1>
    </body>
</html>
```

If margin contains three values.
ex:

margin : 50px 100px 150px;

top, right and left, bottom

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid #000;

				margin : 50px  100px  150px;

			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>
```

If margin contains two values.
ex:

margin: 50px 100px;
top and bottom , left and right

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid #000;
				margin : 50px  100px;
			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>
```

If margin contains one property:
ex:

margin : 100px;

All sides are 100px

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid #000;

				margin :100px;

			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>
```

CSS border-radius property

The border-radius property defines the radius of the element's corners.
We have following list of border-radius properties.

1) border-top-left-radius
2) border-top-right-radius
3) border-bottom-right-radius
4) border-bottom-left-raidus
5) border-radius shorthand

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid black;
				border-top-left-radius: 5px;
				border-top-right-radius:10px;
				border-bottom-left-radius:15px;
				border-bottom-right-radius:20px;
			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>

	</body>
```

```
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid black;
				border-radius : 5px 10px 15px 20px;
			}

		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>

	</body>
</html>
```

Here border-radius goes to top left, top right, bottom right and bottom left.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid black;
				border-radius : 5px 20px;
			}
		</style>
	</head>
	<body>
			<h1> This is Heading Tag </h1>
	</body>
</html>
```

Here border-radius goes to top left, bottom right, top right and bottom left.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
```

```
                h1
                {
                        border:2px solid black;
                        border-radius : 20px;
                }

        </style>
    </head>
    <body>
                <h1> This is Heading Tag </h1>

    </body>
</html>
```

Here border-radius goes to equals for four sides.

CSS padding property

CSS padding property provide the space around the element content. inside the defined border.

We have following list of margin properties.

1) padding-top
2) padding-right
3) padding-bottom
4) padidng-left
5) padding shorthand property

All the above properties will take following values.

1) auto : It describes the padding by the browser.
2) value : It describes the padding in px,pt,cm,em and etc.
3) % : It describes the padding in percentage.
4) inherit : It will take the padding from parent tag.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            h1
            {
                border:2px solid red;
                padding-top:100px;
                padding-right:100px;
                padding-bottom:100px;
                padding-left:100px;
            }
        </style>
    </head>
    <body>
                <h1> Heading Tag</h1>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid red;
				padding-top:10%;
				padding-right:20%;
				padding-bottom:30%;
				padding-left:40%;
			}
		</style>
	</head>
	<body>
				<h1> Heading Tag</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				border:2px solid blue;
				padding-top:100px;
			}
			h1
			{
				border:2px solid red;
				padding-top:inherit;
			}
		</style>
	</head>
	<body>
			<div>
				<h1> Heading Tag</h1>
			</div>
	</body>
</html>
```

If padding contains four values.

ex: padding: 25px 50px 75px 100px;
top , right, bottom , left

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid red;
				padding: 25px 50px 75px 100px;
			}
		</style>
	</head>
	<body>
			<h1> Heading Tag</h1>
	</body>
</html>
```

If padding contains three values.

ex: padding: 25px 75px 100px;

top , left and right, bottom

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				border:2px solid red;
				padding: 25px 75px 100px;
			}
		</style>
	</head>
	<body>
			<h1> Heading Tag</h1>
	</body>
</html>
```

If padding contains two values.

ex: padding: 75px 100px;

top and bottom , left and right

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
```

```
                h1
                {
                        border:2px solid red;
                        padding: 75px 100px;
                }
        </style>
    </head>
    <body>
                <h1> Heading Tag</h1>
    </body>
</html>
```

If padding contains one value.
ex: padding: 50px;
all sides are 50px.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            h1
            {
                border:2px solid red;
                padding: 50px;
            }
        </style>
    </head>
    <body>
            <h1> Heading Tag</h1>
    </body>
</html>
```

CSS Text property

color

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            h1
            {
                    color:blue;
            }
        </style>
    </head>
    <body>
            <h1> Heading Tag</h1>
    </body>
```

```
</html>
```

background-color

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:blue;
				background-color:yellow;
			}
		</style>
	</head>
	<body>
			<h1> Heading Tag</h1>
	</body>
</html>
```

text-align

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:blue;
				background-color:yellow;
				text-align:right;
			}
		</style>
	</head>
	<body>
			<h1> Heading Tag</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:blue;
				background-color:yellow;
```

```
                                        text-align:center;
                        }
                </style>
        </head>
        <body>
                        <h1> Heading Tag</h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>

                        h1
                        {
                                        color:blue;
                                        background-color:yellow;
                                        text-align:left;
                        }
                </style>
        </head>
        <body>
                        <h1> Heading Tag</h1>
        </body>
</html>
```

text-transform

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                        color:blue;
                                        background-color:yellow;
                                        text-align:center;
                                        text-transform:uppercase;
                        }
                </style>
        </head>
        <body>
                        <h1> heading tag</h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:blue;
				background-color:yellow;
				text-align:center;
				text-transform:lowercase;
			}
		</style>
	</head>
	<body>
			<h1> heading tag</h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:blue;
				background-color:yellow;
				text-align:center;
				text-transform:capitalize;
			}
		</style>
	</head>
	<body>
			<h1> heading tag</h1>
	</body>
</html>
```

<u>text-decoration</u>

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				color:blue;
```

```
                                        background-color:yellow;
                                        text-align:center;
                                        text-transform:capitalize;
                                        text-decoration:underline;
                        }
                </style>
        </head>
        <body>
                        <h1> heading tag</h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                        color:blue;
                                        background-color:yellow;
                                        text-align:center;
                                        text-transform:capitalize;
                                        text-decoration:none;
                        }
                </style>
        </head>
        <body>
                        <h1> heading tag</h1>
        </body>
</html>
```

letter-spacing

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                        color:blue;
                                        background-color:yellow;
                                        text-align:center;
                                        text-transform:capitalize;
                                        text-decoration:none;
                                        letter-spacing:4px;
                        }
                </style>
        </head>
```

```
        <body>
                <h1> heading tag</h1>
        </body>
</html>
```

font-size

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                color:blue;
                                background-color:yellow;
                                text-align:center;
                                text-transform:capitalize;
                                text-decoration:none;
                                letter-spacing:4px;
                                font-size:80px;
                        }
                </style>
        </head>
        <body>
                <h1> heading tag</h1>
        </body>
</html>
```

font-family

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                color:blue;
                                background-color:yellow;
                                text-align:center;
                                text-transform:capitalize;
                                text-decoration:none;
                                letter-spacing:4px;
                                font-size:80px;
                                font-family:cursive;
                        }
                </style>
        </head>
        <body>
```

```
                <h1> heading tag</h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                color:blue;
                                background-color:yellow;
                                text-align:center;
                                text-transform:capitalize;
                                text-decoration:none;
                                letter-spacing:4px;
                                font-size:80px;
                                font-family:monospace;
                        }
                </style>
        </head>
        <body>
                <h1> heading tag</h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                color:blue;
                                background-color:yellow;
                                text-align:center;
                                text-transform:capitalize;
                                text-decoration:none;
                                letter-spacing:4px;
                                font-size:80px;
                                font-family:Fantasy ;
                        }
                </style>
        </head>
        <body>
                <h1> heading tag</h1>
        </body>
```

```
</html>
```

font-weight

ex:

--

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>

			span
			{
				font-weight:bold;
			}

		</style>
	</head>
	<body>
			<span>This is span tag</span>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>

			span
			{
				font-weight:600;
			}

		</style>
	</head>
	<body>
			<span>This is span tag</span>
	</body>
</html>
```

CSS overflow property

The overflow property specifies what should happen if content overflow.

This property spcifies weither to clip content or to add scrollbars when an element content is too big to fit in a specified area.

Note: The overflow property works for block elements with a specified height.

value		Description
visible	The overflow is not clipped.	It rendered outside the element 's box and

	it is default value.
hidden	the overflow is clipped and rest of the content will be invisible.
scroll	The overflow is clipped,but a scroll-bar is added to see the rest of the content .
auto	The overflow is clipped,a scroll-bar should be added to the rest of the content.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				border:2px solid black;
				overflow:visible;
			}
		</style>
	</head>
	<body>
			<div>
				Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS). It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
			</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				border:2px solid black;
				overflow:hidden;
			}
		</style>
	</head>
	<body>
			<div>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS). It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

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

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
                                height:200px;
                                border:2px solid black;
                                overflow:scroll;
                        }

                </style>
        </head>
        <body>
                        <div>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS). It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

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

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
                                height:200px;
                                border:2px solid black;
                                overflow:auto;
```

```
                }
            </style>
        </head>
        <body>
                <div>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites.
                </div>
        </body>
</html>
```

CSS box-shadow property

The box-shadow property attaches one or more shadows to an element.

Syntax box-shadow: none |h-offset v-offset blur spread color

ex: box-shadow: 2px 2px 3px 10px blue;

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
                                height:200px;
                                margin:50px auto;
                                border-radius:15px;
                                overflow:hidden;
                                box-shadow: 2px 2px 25px 6px #FF0000;
                        }
                </style>
        </head>
        <body>
                <div>

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

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
```

```
                height:200px;
                margin:50px auto;
                border-radius:15px;
                overflow:hidden;
                box-shadow: 2px 2px 25px 6px #FF0000 inset;
            }
        </style>
    </head>
    <body>
            <div>

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

HTML table creation

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>

        </style>
    </head>
    <body>
        <table border="1" text-align="center" width="100%">
            <thead>
                <tr>
                    <th>No</th>
                    <th>NAME</th>
                    <th>ADDRESS</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>101</td>
                    <td>Alan</td>
                    <td>Texas</td>
                </tr>
                <tr>
                    <td>102</td>
                    <td>Jose</td>
                    <td>Florida</td>
                </tr>
                <tr>
                    <td>103</td>
                    <td>Nancy</td>
                    <td>Miami</td>
                </tr>
            </tbody>
```

```
            </table>
        </body>
</html>
```

CSS Sample Design

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        body
                        {
                                background-image: url("images/bg.jpg");
                                background-repeat:no-repeat;
                                background-size:cover;
                                background-position:center 0px;
                                background-attachment:fixed;
                        }
                        div
                        {
                                width:400px;
                                height:400px;
                                box-shadow:2px 2px 20px 6px #FFF;
                                background-color:transparent;
                                margin:50px auto;
                        }
                        p
                        {
                                color:#FFF;
                                font-weight:bold;
                                padding:20px;
                                text-align:justify;
                        }
                </style>
        </head>
        <body>
                        <div>
                                <p>
```

Lesson Summary. Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Lesson Summary. Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Lesson Summary. Web technology refers to the means by which.

Lesson Summary. Web technology refers to the means by which.

Lesson Summary. Web technology refers to the means by which.

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

CSS float property

It is widely used property on a web page.

The float property specifies how an element should float.

value	Description
none	The element does not float.
left	the element floats to the left of its container.
right	The element floats to the right of its container.

ex:1

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            div
            {
                width:300px;
                height:300px;
                border:2px solid blue;
                float:none;
            }
        </style>
    </head>
    <body>
            <div>

            </div>
            <div>

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

ex:2

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            div
            {
                width:300px;
```

```
                        height:300px;
                        border:2px solid blue;
                        float:left;
                }
        </style>
    </head>
    <body>
                <div>

                </div>
                <div>

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

ex:3

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            div
            {
                width:300px;
                height:300px;
                border:2px solid blue;
                float:right;
            }
        </style>
    </head>
    <body>
                <div>

                </div>
                <div>

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

ex:4

```
<!DOCTYPE html>
<html>
    <head>

        <title>MyPage!</title>
        <style>
            div
            {
```

```
                    width:300px;
                    height:300px;
                    border:2px solid blue;
                    float:left;
                    margin:50px 140px;
                }
            </style>
        </head>
        <body>
                <div>

                </div>
                <div>

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

Types of selectors in CSS

We have five selectors in CSS.

1) Element selector

The element selector selects HTML elements based on element name.

ex:

```
<!DOCTYPE html>
<html>
        <head>
            <title>MyPage!</title>
            <style>
                h1
                {
                    color:blue;
                    background-color:cyan;
                    text-align:center;
                }
            </style>
        </head>
        <body>
                <h1> Heading Tag </h1>
                <h1> Heading Tag </h1>
        </body>
</html>
```

2) Id selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element.

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			#id1
			{
				color:blue;
				background-color:cyan;
			}
			#id2
			{
				text-align:center;
			}
		</style>
	</head>
	<body>
			<h1 id="id1"> Heading Tag </h1>
			<h1 id="id2"> Heading Tag </h1>
	</body>
</html>
```

Once element can accept only one ID at a time.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			#id1
			{
				color:blue;
				background-color:cyan;
			}
			#id2
			{
				text-align:center;
			}
		</style>
	</head>
	<body>
			<h1 id="id1"> Heading Tag </h1>
			<h1 id="id2"> Heading Tag </h1>
			<h1 id="id1 id2"> Heading Tag</h1>
	</body>
</html>
```

3) Class selector
The class selector selects HTML elements with a specific class attribute.
To select elements with a specific class, write a period (.) character,
followed by the class name.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			.myClass1
			{
				color:blue;
				background-color:cyan;
			}
			.myClass2
			{
				text-align:center;
			}
		</style>
	</head>
	<body>
			<h1 class="myClass1"> Heading Tag </h1>
			<h1 class="myClass2"> Heading Tag </h1>
	</body>
</html>
```

We can declare multiple classes in a single element.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			.myClass1
			{
				color:blue;
				background-color:cyan;
			}
			.myClass2
			{
				text-align:center;
			}
		</style>
	</head>
	<body>
			<h1 class="myClass1"> Heading Tag </h1>
			<h1 class="myClass2"> Heading Tag </h1>
			<h1 class="myClass1 myClass2"> Heading Tag</h1>
```

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

4) Group selector
The grouping selector selects all the HTML elements with the same style definitions
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1,p,div
			{
				color:red;
				background-color:rgb(255, 255, 0);
				text-align:center;
			}
		</style>
	</head>
	<body>
			<h1>Heading Tag</h1>
			<p>Paragraph Tag</p>
			<div>Division T ag</div>
	</body>
</html>
```

5) Universal selector
The universal selector (*) selects all HTML elements on the page.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			*
			{
				color:red;
				background-color:rgb(255, 255, 0);
				text-align:center;
			}
		</style>
	</head>
	<body>
			<h1>Heading Tag</h1>
			<p>Paragraph Tag</p>
			<div>Division T ag</div>
	</body>
</html>
```

CSS clear property

The clear property controls the flow next to floated elements.

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			#box1
			{
				width:200px;
				height:200px;
				background-color:rgb(255,0,0);
				float:left;
			}
			#box2
			{
				width:200px;
				height:200px;
				background-color:hsl(60, 100%, 50%);
				float:left;
			}
		</style>
	</head>
	<body>
			<div id="box1"></div>
			<div id="box2"></div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			#box1
			{
				width:200px;
				height:200px;
				background-color:rgb(255,0,0);
				float:left;
			}
			#box2
			{
				width:200px;
				height:200px;
				background-color:hsl(60, 100%, 50%);
				float:right;
			}
```

```
		</style>
	</head>
	<body>
			<div id="box1"></div>
			<div id="box2"></div>
	</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			#box1
			{
				width:200px;
				height:200px;
				background-color:rgb(255,0,0);
				float:left;
			}
			#box2
			{
				width:200px;
				height:200px;
				background-color:hsl(60, 100%, 50%);
				float:right;
			}
			#box3
			{
				width:100%;
				height:200px;
				background-color:#00ff00;
				clear:none;
			}
		</style>
	</head>
	<body>
			<div id="box1"></div>
			<div id="box2"></div>
			<div id="box3"></div>
	</body>
</html>
```

ex:4

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
```

```
                #box1
                {
                        width:200px;
                        height:200px;
                        background-color:rgb(255,0,0);
                        float:left;
                }
                #box2
                {
                        width:200px;
                        height:200px;
                        background-color:hsl(60, 100%, 50%);
                        float:right;
                }
                #box3
                {
                        width:100%;
                        height:200px;
                        background-color:#00ff00;
                        clear:both;
                }
        </style>
    </head>
    <body>
                <div id="box1"></div>
                <div id="box2"></div>
                <div id="box3"></div>
    </body>
</html>
```

ex:5

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                #box1
                {
                        width:200px;
                        height:300px;
                        background-color:rgb(255,0,0);
                        float:left;
                }
                #box2
                {
                        width:200px;
                        height:200px;
                        background-color:hsl(60, 100%, 50%);
                        float:right;
                }
```

```
                #box3
                {
                        width:100%;
                        height:200px;
                        background-color:#00ff00;
                        clear:left;
                }
        </style>
    </head>
    <body>
                <div id="box1"></div>
                <div id="box2"></div>
                <div id="box3"></div>
    </body>
</html>
```

ex:6

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                #box1
                {
                        width:200px;
                        height:300px;
                        background-color:rgb(255,0,0);
                        float:left;
                }
                #box2
                {
                        width:200px;
                        height:200px;
                        background-color:hsl(60, 100%, 50%);
                        float:right;
                }
                #box3
                {
                        width:100%;
                        height:200px;
                        background-color:#00ff00;
                        clear:right;
                }
        </style>
    </head>
    <body>
                <div id="box1"></div>
                <div id="box2"></div>
                <div id="box3"></div>
    </body>
```

</html>

CSS Display property

The display property specifies the display behavior of an element.

syntax : display: value;

value	Description
none	The element is completely removed.
inline	displays an element as on inline element. Any height and width properties will not effect.
block	Displays an element as block element. IT starts on a new line and takes up the whole width.
inline-block	Displays an element as an inline-level container. The element itself is formatting as an inline element. but we can apply height and width values.
inherit	Inherits this property from its parent element.

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				display:none;
			}
			p
			{
				display:none;
			}
		</style>
	</head>
	<body>
			<h1>Heading Tag</h1>
			<p>Paragraph Tag</p>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				display:inline;
			}
		</style>
	</head>
	<body>
```

```
                <div>Division Tag</div>
                <div>Division Tag</div>
                <div>Division Tag</div>
        </body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        span
                        {
                                display:block;
                                border:2px solid black;
                                width:50%;
                        }
                </style>
        </head>
        <body>
                        <span>Span Tag</span>
                        <span>Span Tag</span>
                        <span>Span Tag</span>
        </body>
</html>
```

ex:4

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        span
                        {
                                display:inline-block;
                                border:2px solid black;
                                width:200px;
                        }
                </style>
        </head>
        <body>
                        <span>Span Tag</span>
                        <span>Span Tag</span>
                        <span>Span Tag</span>
        </body>
</html>
```

box-sizing property

The box-sizing property defines how the width and height of an element are calculated: should they include padding and borders, or not.

box-sizing: content-box

It will take seperate width,border and padding.But it will not take margin.

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:300px;
				height:300px;
				background-color:yellow;
				border:2px solid red;
				padding:10px;
				box-sizing:content-box;
			}
		</style>
	</head>
	<body>
			<div></div>
	</body>
</html>
```

box-sizing: border-box

It includes width, border and padding. But it will not include marging.

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:300px;
				height:300px;
				background-color:yellow;
				border:2px solid red;
				padding:10px;
				box-sizing:border-box;
			}
		</style>
	</head>
	<body>  <div>      </div>  </body>
</html>
```

CSS Design 1

Task1
|
|----images
|
|----rock.png
|-----css
|
|----mystyles.css
|
|----index.html

index.html

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage</title>

		<!-- add external css -->
		<link rel="stylesheet" href="css/mystyles.css">
	</head>
	<body>
			<div class="container">
				<div class="box1">
					<img src="images/rock.png"/>
				</div>
				<div class="box2">
					<h3>Dwayne Johnson</h3>
					<p>
Dwayne Douglas Johnson (born May 2, 1972), also known by his ring name the Rock, is an American actor and retired professional wrestler. Widely regarded as one of the greatest professional wrestlers of all time,[6][7] he was integral to the development and success of the World Wrestling Federation (WWF, now WWE) during the Attitude Era, an industry boom period in the late 1990s and early 2000s. Johnson wrestled for the WWF for eight years
					</p>
					<p>
Dwayne Douglas Johnson (born May 2, 1972), also known by his ring name the Rock, is an American actor and retired professional wrestler. Widely regarded as one of the greatest professional wrestlers of all time,[6][7] he was integral to the development and success of the World Wrestling Federation (WWF, now WWE) illion worldwide,[8] making him one of the world's highest-grossing and highest-paid actors
					</p>
				</div>
			</div>
	</body>
</html>
```

mystyles.css

```
*
{
	margin:0;
	padding:0;
}
.container
{
	width:800px;
	height:400px;
	box-sizing:border-box;
	background-color: #34ace0;
	margin: 150px auto;
	box-shadow: 2px 2px 16px 6px #c3c3c3;
}
.box1
{
	width:400px;
	height:400px;
	box-sizing:border-box;
	background-color:#33d9b2;
	float:left;
}
.box1  img
{
	width:100%;
	height:100%;
}
.box2
{
	width:400px;
	height:400px;
	box-sizing:border-box;
	background-color:#f7f1e3;
	float:right;
	padding:20px;
}
.box2 h3
{
	text-align:center;
	padding:10px;
	text-transform:uppercase;
}
.box2 p
{
	text-align:justify;
}
```

CSS Design2

Task1

```
|
|-----css
        |
        |----mystyles.css
|
|----index.html
```

index.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage</title>

                <!-- add external css -->
                <link rel="stylesheet" href="css/mystyles.css">
        </head>
        <body>
                <header class="header">
                        <h3>I<span>HUB</span>TALENT</h3>
                        <ul>
                                <li><a href="">Home</a></li>
                                <li><a href="">About</a></li>
                                <li><a href="">Service</a></li>
                                <li><a href="">Portfolio</a></li>
                                <li><a href="">Contact</a></li>
                        </ul>
                </header>
        </body>
</html>
```

mystyles.css

```
*
{
        margin:0;
        padding:0;
}
.header
{
        width:100%;
        background-color:#3dc1d3;
        padding:20px;
}
.header h3
{
        float:left;
}
.header ul
{
```

```
	float:right;
	list-style-type:none;
}
.header span
{
	font-style:italic;
	color:#FFF;
}
.header ul li
{
	display:inline-block;
	margin: 0px  25px;
	line-height:23px;
}
.header ul li a
{
	text-decoration:none;
	color:#000;
	font-size:13px;
	text-transform:uppercase;
}
.header ul li a:hover
{
	color:#FFF;
}
```

CSS transform property

CSS tranform property allows use to move ,rotate or skew elements.

CSS tranform property contains following transformation methods.

ex:

translate()
rotate()
scaleX()
scaleY()
skewX()
skewY()
skew()

transform: translate()

The translate() method moves an element from its current position to the parameters given by the X-axis and the Y-axis.

ex:

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
```

```
                {
                        width:200px;
                        height:200px;
                        border:2px solid black;
                        box-sizing:border-box;
                        transform:translate(200px,100px);
                }
        </style>
    </head>
    <body>
                <div>
                        CSS is used to apply the styles on elements to describe how an
                        element should look like and to perform some sort of
animations.
                </div>
    </body>
</html>
```

transform: scaleX()
The scaleX() method increases and decreases the width of the element.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB TALENT</title>
        <style>
            div
            {
                    width:300px;
                    height:200px;
                    border:2px solid black;
                    margin:100px auto;
                    transform:scaleX(2);
            }
        </style>

    </head>
    <body>
        <div>
            HTML is widely used language on web to develop web pages and web
applications
        </div>
    </body>
</html>
```

transform: scaleY()
The scaleX() method increases and decreases the height of the element.
ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>IHUB TALENT</title>
		<style>
			div
			{
				width:300px;
				height:200px;
				border:2px solid black;
				margin:100px auto;
				transform:scaleY(2);
			}
		</style>
	</head>
	<body>
		<div>
			HTML is widely used language on web to develop web pages and web applications
		</div>
	</body>
</html>
```

transform : scale()

The scaleX() method increases and decreases the width and height of the element.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
		<style>
			div
			{
				width:300px;
				height:200px;
				border:2px solid black;
				margin:100px auto;
				transform:scale(2);
			}
		</style>

	</head>
	<body>
		<div>
			HTML is widely used language on web to develop web pages and web applications
		</div>
	</body>
</html>
```

transform: skewX()
The skewX() method skews an element along the X-axis by the given angle.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
		<style>
			div
			{
				width:300px;
				height:200px;
				border:2px solid black;
				margin:100px auto;
				transform:skewX(30deg);
			}
		</style>
	</head>
	<body>
		<div>
			HTML is widely used language on web to develop web pages and web applications
		</div>
	</body>
</html>
```

transform: skewY()
The skewY() method skews an element along the Y-axis by the given angle.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
		<style>
			div
			{
				width:300px;
				height:200px;
				border:2px solid black;
				margin:100px auto;
				transform:skewY(30deg);
			}
		</style>

	</head>
	<body>
		<div>
			HTML is widely used language on web to develop web pages and web applications
		</div>
```

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

transform: skew()

The skew() method skews an element along the X-axis and Y-axis by the given angle.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>
		<style>
			div
			{
				width:300px;
				height:200px;
				border:2px solid black;
				margin:100px auto;
				transform:skew(30deg);
			}
		</style>

	</head>
	<body>
		<div>
			HTML is widely used language on web to develop web pages and web applications
		</div>
	</body>
</html>
```

transform: rotate()

The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

ex:8

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				border:2px solid black;
				box-sizing:border-box;
				margin:50px auto;
				transform:rotate(30deg);

			}
```

```
		</style>
	</head>
	<body>
			<div>
				CSS is used to apply the styles on elements to describe how an
				element should look like and to perform some sort of
animations.
			</div>
	</body>
</html>
```

ex:9

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				border:2px solid black;
				box-sizing:border-box;
				margin:50px auto;
				transform:rotate(-30deg);

			}
		</style>
	</head>
	<body>
			<div>
				CSS is used to apply the styles on elements to describe how an
				element should look like and to perform some sort of
animations.
			</div>
	</body>
</html>
```

CSS transition property

CSS transition property allows us to change property value smoothly , over a given duration.
To see the effects on an element ,we need to use mouse over to the element.
We have following properties in CSS transition.
ex:

- transition-delay
- transition-duration
- transition-property
- transition-timing-function
- or
- transition

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				background-color:red;
				box-sizing:border-box;
				transition-property:width,height;
			}
			div:hover
			{
				width:400px;
				height:400px;
			}
		</style>
	</head>
	<body>
			<div>
			</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				background-color:red;
				box-sizing:border-box;
				transition-property:all;
			}
			div:hover
			{
				width:400px;
				height:400px;
			}
		</style>
	</head>
	<body>
			<div>
```

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

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
                                height:200px;
                                background-color:red;
                                box-sizing:border-box;
                                transition-property:all;
                                transition-duration:3s;
                        }
                        div:hover
                        {
                                width:400px;
                                height:400px;
                        }
                </style>
        </head>
        <body>
                        <div>
                        </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
                                height:200px;
                                background-color:red;
                                box-sizing:border-box;
                                transition-property:all;
                                transition-duration:3s;
                                transition-timing-function:linear;
                        }
                        div:hover
                        {
```

```
                        width:400px;
                        height:400px;
                }
        </style>
    </head>
    <body>
                <div>
                </div>
    </body>
</html>

ex:
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            div
            {
                width:200px;
                height:200px;
                background-color:red;
                box-sizing:border-box;
                transition-property:all;
                transition-duration:3s;
                transition-timing-function:ease-in;
            }
            div:hover
            {
                width:400px;
                height:400px;
            }
        </style>
    </head>
    <body>
                <div>
                </div>
    </body>
</html>

ex:
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            div
            {
                width:200px;
                height:200px;
```

```
                        background-color:red;
                        box-sizing:border-box;
                        transition-property:all;
                        transition-duration:3s;
                        transition-timing-function:ease-out;
                }
                div:hover
                {
                        width:400px;
                        height:400px;
                }
        </style>
</head>
<body>
        <div>
        </div>
</body>
</html>

ex:
<!DOCTYPE html>

<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                width:200px;
                                height:200px;
                                background-color:red;
                                box-sizing:border-box;
                                transition-property:all;
                                transition-duration:3s;
                                transition-timing-function:ease-in-out;
                        }
                        div:hover
                        {
                                width:400px;
                                height:400px;
                        }
                </style>
        </head>
        <body>
                <div>
                </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				background-color:red;
				box-sizing:border-box;
				transition-property:all;
				transition-duration:3s;
				transition-timing-function:ease-in-out;
				transition-delay:5s;
			}
			div:hover
			{
				width:400px;
				height:400px;
			}
		</style>
	</head>
	<body>
			<div>
			</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>

<html>
	<head>
		<title>MyPage!</title>
		<style>
			div
			{
				width:200px;
				height:200px;
				background-color:red;
				box-sizing:border-box;
				transition:all 2s linear 3s;
			}
			div:hover
			{
				width:400px;
				height:400px;
				background-color:blue;
```

```
                }
        </style>
    </head>
    <body>
                <div>
                </div>
    </body>
</html>
```

CSS Design 1

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            img
            {
                width:250px;
                height:250px;
                margin:100px 400px;
                transition:all 2s linear;
            }
            img:hover
            {
                transform:rotate(360deg);
            }
        </style>
    </head>
    <body>
            <img src="images/rock.jpg"/>
    </body>
</html>
```

CSS Design 2

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>

            img
            {
                width:300px;
                height:300px;
                margin:100px 30px;
                filter:grayscale(100%);
                transition:all 1s linear;
            }
            img:hover
            {
```

```
                transform:scale(1.2);
                filter:grayscale(0);
            }
        </style>
    </head>
    <body>
            <div>
            <img src="images/girl1.jpeg"/>
            <img src="images/girl2.jpg"/>
            <img src="images/girl3.jpg"/>
            </div>

    </body>
</html>
```

CSS position property

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).

The following are the list of values to position property.

static

It is default value. Elements render in order, as they appear in the document flow.

absolute

The element is positioned relative to its first positioned (not static) ancestor element.

fixed

The element is positioned relative to the browser window.

relative

The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position.

sticky

The element is positioned based on the user's scroll position

css position property mandatory should have following properties.

i) left
ii) right
iii) top
iv) bottom

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            img
            {
                width:200px;
                height:200px;
                border:2px solid black;
                position:static;
                top:0;
                left:0;
```

```
                }
            </style>
        </head>
        <body>
                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <img src="images/rock.png"/>

                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>

                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>

                <p>
                    Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

</body>
</html>

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			img
			{
				width:200px;
				height:200px;
				border:2px solid black;
				position:absolute;
				top:100px;
				left:100px;
			}
```

```
        </style>
    </head>
    <body>
            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
            <img src="images/rock.png"/>

            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>

            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>

            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
            <p>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

</body>
</html>

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			img
			{
				width:200px;
				height:200px;
				border:2px solid black;
				position:relative;
				top:100px;
				left:100px;
			}
		</style>
```

```
</head>
<body>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <img src="images/rock.png"/>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

</body>
</html>

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			img
			{
				width:200px;
				height:200px;
				border:2px solid black;
				position:fixed;
				top:100px;
				left:100px;
			}
		</style>
```

```
</head>
<body>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <img src="images/rock.png"/>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

</body>
</html>

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			img
			{
				width:200px;
				height:200px;
				border:2px solid black;
				position:sticky;
				top:100px;
				left:100px;
			}
		</style>
```

```
</head>
<body>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <img src="images/rock.png"/>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>

        <p>
            Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
        </p>
        <p>
```

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

</body>
</html>

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage</title>
		<style>

			.myClass1
			{
				width:200px;
				height:200px;
				position:relative;
				left:420px;

			}
			.myClass2
```

```
                {
                        width:200px;
                        height:200px;
                }

        </style>
    </head>
    <body>
        <img src="images/girl3.jpg" class="myClass1"/>
        <img src="images/girl2.jpg" class="myClass2"/>
    </body>
</html>
```

CSS z-index property

The z-index property specifies the stack order of an element.
An element with greater stack order is always in front of an element with a lower stack order.
Note: z-index only works on positioned elements.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            img
            {
                width:200px;
                height:200px;
                border:2px solid black;
                background-color:#FFF;
                position:absolute;
                top:0;
                left:0;
                z-index:1;
            }
        </style>
    </head>
    <body>
            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
            <p>
                Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
            </p>
```

```
                <p>
                        Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <p>
                        Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <img src="images/rock.png"/>


        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        img
                        {
                                width:200px;
                                height:200px;
                                border:2px solid black;
                                background-color:#FFF;
                                position:absolute;
                                top:0;
                                left:0;
                                z-index:-1;
                        }
                </style>
        </head>
        <body>
                <p>
                        Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <p>
                        Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
```

```
                <p>
                        Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <p>
                        Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. Web technology involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).
                </p>
                <img src="images/rock.png"/>
        </body>
</html>
```

CSS opacity property

The opacity property sets the opacity level for an element.

The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
                        {
                                margin:50px auto;
                                background-color:red;
                                width:200px;
                                height:200px;
                                opacity:1;
                        }
                </style>
        </head>
        <body>
                        <div>

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

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        div
```

```
                {
                        margin:50px auto;
                        background-color:red;
                        width:200px;
                        height:200px;
                        opacity:0.5;
                }
        </style>
    </head>
    <body>
                <div>

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

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            div
            {
                margin:50px auto;
                background-color:red;
                width:200px;
                height:200px;
                opacity:0.1;
            }
        </style>
    </head>
    <body>
                <div>

                </div>


    </body>
</html>
```

CSS Gradients

CSS gradients let you display smooth transitions between two or more specified colors.
CSS defines three types of gradients:

1) Linear Gradients (goes down/up/left/right/diagonally)
2) Conic Gradients (rotated around a center point)
3) Radial Gradients (defined by their center)

ex:

```
<!DOCTYPE html>
<html>
```

```
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				height:100vh;
				background:linear-gradient(yellow,red);
			}
		</style>
	</head>
	<body>

	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				height:100vh;
				background:linear-gradient(to left,yellow,red);
			}
		</style>
	</head>
	<body>

	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				height:100vh;
				background-image:conic-gradient(yellow,red);
			}
		</style>
	</head>
	<body>

	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			body
			{
				height:100vh;
				background-image:radial-gradient(yellow,red);
			}
		</style>
	</head>
	<body>

	</body>
</html>
```

CSS flexbox

It is a one-dimensional layout methods for laying out our items in rows and columns.
CSS flexbox is a better way to align items into a container.
Flexbox= flexible + box.
To create a flexbox model, we need to define a "flex container".
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			.container
			{
				width:100%;
				height:400px;
				border:2px solid black;
				display:flex;
				flex-direction:row;
			}
			.container .item
			{
				width:150px;
				height:150px;
				border:2px solid black;
			}

		</style>
	</head>
	<body>
			<div class="container">
				<div class="item">Item1</div>
				<div class="item">Item2</div>
```

```
                        <div class="item">Item3</div>
                        <div class="item">Item4</div>
                        <div class="item">Item5</div>
                        <div class="item">Item6</div>
                    </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        .container
                        {
                                width:100%;
                                height:400px;
                                border:2px solid black;
                                display:flex;
                                flex-direction:row-reverse;
                        }
                        .container .item
                        {
                                width:150px;
                                height:150px;
                                border:2px solid black;
                        }

                </style>
        </head>
        <body>
                        <div class="container">
                                <div class="item">Item1</div>
                                <div class="item">Item2</div>
                                <div class="item">Item3</div>
                                <div class="item">Item4</div>
                                <div class="item">Item5</div>
                                <div class="item">Item6</div>
                        </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        .container
```

```
                {
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:flex;
                        flex-direction:column;
                }
                .container .item
                {
                        width:150px;
                        height:150px;
                        border:2px solid black;
                }

        </style>
    </head>
    <body>
                <div class="container">
                        <div class="item">Item1</div>
                        <div class="item">Item2</div>
                        <div class="item">Item3</div>
                        <div class="item">Item4</div>
                        <div class="item">Item5</div>
                        <div class="item">Item6</div>
                </div>
    </body>
</html>

ex:
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                .container
                {
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:flex;
                        flex-direction:column-reverse;
                }
                .container .item
                {
                        width:150px;
                        height:150px;
                        border:2px solid black;
                }
        </style>
    </head>
```

```
	<body>
		<div class="container">
			<div class="item">Item1</div>
			<div class="item">Item2</div>
			<div class="item">Item3</div>
			<div class="item">Item4</div>
			<div class="item">Item5</div>
			<div class="item">Item6</div>
		</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			.container
			{
				width:100%;
				height:400px;
				border:2px solid black;
				display:flex;
				flex-direction:row;
				flex-wrap:no-wrap;
			}
			.container .item
			{
				width:150px;
				height:150px;
				border:2px solid black;
			}

		</style>
	</head>
	<body>
		<div class="container">
			<div class="item">Item1</div>
			<div class="item">Item2</div>
			<div class="item">Item3</div>
			<div class="item">Item4</div>
			<div class="item">Item5</div>
			<div class="item">Item6</div>
		</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			.container
			{
				width:100%;
				height:400px;
				border:2px solid black;
				display:flex;
				flex-direction:row;
				flex-wrap:wrap;
			}
			.container .item
			{
				width:150px;
				height:150px;
				border:2px solid black;
			}

		</style>
	</head>
	<body>
			<div class="container">
				<div class="item">Item1</div>
				<div class="item">Item2</div>
				<div class="item">Item3</div>
				<div class="item">Item4</div>
				<div class="item">Item5</div>
				<div class="item">Item6</div>
			</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			.container
			{
				width:100%;
				height:400px;
				border:2px solid black;
				display:flex;
				flex-flow:row  wrap;
			}
```

```
                .container .item
                {
                        width:150px;
                        height:150px;
                        border:2px solid black;
                }

        </style>
    </head>
    <body>
                <div class="container">
                        <div class="item">Item1</div>
                        <div class="item">Item2</div>
                        <div class="item">Item3</div>
                        <div class="item">Item4</div>
                        <div class="item">Item5</div>
                        <div class="item">Item6</div>
                </div>
    </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                .container
                {
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:flex;
                        flex-flow:row  wrap;
                        justify-content:center;
                }
                .container .item
                {
                        width:150px;
                        height:150px;
                        border:2px solid black;
                }

        </style>
    </head>
    <body>
                <div class="container">
                        <div class="item">Item1</div>
                        <div class="item">Item2</div>
                        <div class="item">Item3</div>
```

```
                        <div class="item">Item4</div>
                        <div class="item">Item5</div>
                        <div class="item">Item6</div>
                    </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        .container
                        {
                                width:100%;
                                height:400px;
                                border:2px solid black;
                                display:flex;
                                flex-flow:row  wrap;
                                justify-content:center;
                                align-items:center;
                        }
                        .container .item
                        {
                                width:150px;
                                height:150px;
                                border:2px solid black;
                        }

                </style>
        </head>
        <body>
                        <div class="container">
                                <div class="item">Item1</div>
                                <div class="item">Item2</div>
                                <div class="item">Item3</div>
                                <div class="item">Item4</div>
                                <div class="item">Item5</div>
                                <div class="item">Item6</div>
                        </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
```

```
                .container
                {
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:flex;
                        flex-flow:row  wrap;
                        align-content:center;
                }
                .container .item
                {
                        width:150px;
                        height:150px;
                        border:2px solid black;
                }

        </style>
    </head>
    <body>
                <div class="container">
                        <div class="item">Item1</div>
                        <div class="item">Item2</div>
                        <div class="item">Item3</div>
                        <div class="item">Item4</div>
                        <div class="item">Item5</div>
                        <div class="item">Item6</div>
                </div>
    </body>
</html>
```

CSS Grid layout

The CSS grid layout module offers a grid-based layout system with rows and columns.

Grid layout makes easier to design web pages without having a use of floats and positioning tag.

A grid layout consists of a parent element , with one or more child elements.

ex:

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                .container
                {
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:grid;
                }
                .item1{background-color:#EE5A24;}
```

```
                .item2{background-color:#12CBC4;}
                .item3{background-color:#C4E538;}
                .item4{background-color:#B53471;}
                .item5{background-color:#0652DD;}
                .item6{background-color:#6F1E51;}

        </style>
    </head>
    <body>
        <div class="container">
            <div class="item1">Item1</div>
            <div class="item2">Item2</div>
            <div class="item3">Item3</div>
            <div class="item4">Item4</div>
            <div class="item5">Item5</div>
            <div class="item6">Item6</div>
        </div>
    </body>
</html>

ex:
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
            .container
            {
                width:100%;
                height:400px;
                border:2px solid black;
                display:grid;
                grid-template-rows:150px 150px;
                grid-template-columns:150px 150px 150px;
            }
            .item1{background-color:#EE5A24;}
            .item2{background-color:#12CBC4;}
            .item3{background-color:#C4E538;}
            .item4{background-color:#B53471;}
            .item5{background-color:#0652DD;}
            .item6{background-color:#6F1E51;}

        </style>
    </head>
    <body>
        <div class="container">
            <div class="item1">Item1</div>
            <div class="item2">Item2</div>
            <div class="item3">Item3</div>
            <div class="item4">Item4</div>
```

```
                        <div class="item5">Item5</div>
                        <div class="item6">Item6</div>
                    </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        .container
                        {
                                width:100%;
                                height:400px;
                                border:2px solid black;
                                display:grid;
                                grid-template-rows:150px 150px;
                                grid-template-columns:150px 150px 1fr;
                        }
                        .item1{background-color:#EE5A24;}
                        .item2{background-color:#12CBC4;}
                        .item3{background-color:#C4E538;}
                        .item4{background-color:#B53471;}
                        .item5{background-color:#0652DD;}
                        .item6{background-color:#6F1E51;}

                </style>
        </head>
        <body>
                        <div class="container">
                                <div class="item1">Item1</div>
                                <div class="item2">Item2</div>
                                <div class="item3">Item3</div>
                                <div class="item4">Item4</div>
                                <div class="item5">Item5</div>
                                <div class="item6">Item6</div>
                        </div>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        .container
                        {
```

```
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:grid;
                        grid-template-rows:repeat(2,150px);
                        grid-template-columns:repeat(3,1fr);
                }
                .item1{background-color:#EE5A24;}
                .item2{background-color:#12CBC4;}
                .item3{background-color:#C4E538;}
                .item4{background-color:#B53471;}
                .item5{background-color:#0652DD;}
                .item6{background-color:#6F1E51;}

        </style>
    </head>
    <body>
                <div class="container">
                        <div class="item1">Item1</div>
                        <div class="item2">Item2</div>
                        <div class="item3">Item3</div>
                        <div class="item4">Item4</div>
                        <div class="item5">Item5</div>
                        <div class="item6">Item6</div>
                </div>
    </body>
</html>

ex:
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <style>
                .container
                {
                        width:100%;
                        height:400px;
                        border:2px solid black;
                        display:grid;
                        grid-template-rows:repeat(2,150px);
                        grid-template-columns:repeat(3,1fr);
                        grid-row-gap:10px;
                        grid-column-gap:10px;
                }
                .item1{background-color:#EE5A24;}
                .item2{background-color:#12CBC4;}
                .item3{background-color:#C4E538;}
                .item4{background-color:#B53471;}
                .item5{background-color:#0652DD;}
```

```
                .item6{background-color:#6F1E51;}

            </style>
        </head>
        <body>
                <div class="container">
                    <div class="item1">Item1</div>
                    <div class="item2">Item2</div>
                    <div class="item3">Item3</div>
                    <div class="item4">Item4</div>
                    <div class="item5">Item5</div>
                    <div class="item6">Item6</div>
                </div>
        </body>
</html>

ex:
<!DOCTYPE html>
<html>
        <head>
            <title>MyPage!</title>
            <style>
                .container
                {
                    width:100%;
                    height:400px;
                    border:2px solid black;
                    display:grid;
                    grid-template-rows:repeat(2,150px);
                    grid-template-columns:repeat(3,1fr);
                    grid-gap:20px;
                }
                .item1{background-color:#EE5A24;}
                .item2{background-color:#12CBC4;}
                .item3{background-color:#C4E538;}
                .item4{background-color:#B53471;}
                .item5{background-color:#0652DD;}
                .item6{background-color:#6F1E51;}

            </style>
        </head>
        <body>
                <div class="container">
                    <div class="item1">Item1</div>
                    <div class="item2">Item2</div>
                    <div class="item3">Item3</div>
                    <div class="item4">Item4</div>
                    <div class="item5">Item5</div>
                    <div class="item6">Item6</div>
                </div>
```

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

CSS Google Fonts

If we do not want to use any of the standard fonts in HTML, you can use Google Fonts.
Google Fonts are free to use, and have more than 1000 fonts to choose.
To use any google fonts we need to use below url.
ex: https://fonts.google.com/

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<style>
			@import
url('https://fonts.googleapis.com/css2?family=Ultra&display=swap');
			h1
			{
				font-family: 'Ultra', serif;
			}
		</style>
	</head>
	<body>
			<h1>This is Heading Tag</h1>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<link	href="https://fonts.googleapis.com/css2?family=Ultra&display=swap"
rel="stylesheet">
		<style>

			h1
			{
				font-family: 'Ultra', serif;
			}
		</style>
	</head>
	<body>
			<h1>This is Heading Tag</h1>
	</body>
</html>
```

FontAwsome Icons

If we want to take icons in a CSS then we need to use fontawesome icons.
All fontawesome icons are font based.
To work with fontawesome icons we need to add below CDN link.
ex:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
	</head>
	<body>
				<i class="fa fa-heart"></i>
				<i class="fa fa-home"></i>
				<i class="fa fa-phone"></i>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
	</head>
	<body>
				<i class="fa fa-heart" style="font-size:30px;color:red;"></i>
				<i class="fa fa-home" style="font-size:30px;color:blue;"></i>
				<i class="fa fa-phone" style="font-size:30px;color:green;"></i>
	</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
	</head>
	<body>
		<i class="fa fa-facebook" style="color:blue;"></i>
```

```
                <i class="fa fa-instagram" style="color:pink;"></i>
                <i class="fa fa-twitter" style="color:lightblue;"></i>
                <i class="fa fa-youtube" style="color:Red;"></i>
                <i class="fa fa-linkedin" style="color:blue;"></i>
        </body>
</html>
```

CSS Cursor property

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                text-align:center;
                        }
                        h1:hover
                        {
                                cursor:pointer;
                        }
                </style>
        </head>
        <body>
                                <h1>Mouse Over Here </h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <style>
                        h1
                        {
                                text-align:center;
                        }
                        h1:hover
                        {
                                cursor:zoom-in;
                        }
                </style>
        </head>
        <body>
                                <h1>Mouse Over Here </h1>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				text-align:center;
			}
			h1:hover
			{
				cursor:zoom-out;
			}
		</style>
	</head>
	<body>
				<h1>Mouse Over Here </h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
				text-align:center;
			}
			h1:hover
			{
				cursor:progress;
			}
		</style>
	</head>
	<body>
				<h1>Mouse Over Here </h1>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<style>
			h1
			{
```

```
                        text-align:center;
                }
                h1:hover
                {
                        cursor:copy;
                }
        </style>
    </head>
    <body>
                        <h1>Mouse Over Here </h1>
    </body>
</html>
```

CSS Form

Form is used to accept the values from the enduser.
It will send the data to server or database for processing.
ex:

```
Task3
|
|-----css
        |
        |---mystyles.css
|-----images
        |
        |---micky.png
|
|-----index.html
```

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage</title>

        <!-- add external css -->
        <link rel="stylesheet" href="css/mystyles.css">

    </head>
    <body>
        <div class="container">
            <div class="box1">
                <img src="images/user.png"/>

                <form action="">
<input type="text" name="t1" autocomplete="off" placeholder="username" required/> <br>
                    <input type="password" name="t2"
placeholder="password" required/> <br>
                    <input type="submit" value="Login"/>
                </form>
```

```
                        </div>
                        <div class="box2">
                                <img src="images/micky.png"/>
                        </div>
                </div>
        </body>
</html>
```

mystyles.css

```
*
{
        margin:0;
        padding:0;
}
body
{
        height:100vh;
        display:flex;
        justify-content:center;
        align-items:center;
        background: linear-gradient(yellow,red);
}
.container
{
        width:800px;
        height:400px;
        box-sizing:border-box;
}
.container .box1
{
        width:400px;
        height:400px;
        box-sizing:border-box;
        float:left;
        box-shadow:2px 2px 12px 6px #FFF;
}
.container .box1 img
{
        width:100px;
        height:100px;
        display:block;
        margin:20px auto;
        border-radius:50%;
        border:1px solid black;

}
.container .box1 input[type="text"],input[type="password"]
{
        width:300px;
        height:35px;
```

```
        display:block;
        margin:10px auto;
        border:2px solid #FFF;
        background-color:transparent;
        border-radius:15px;
}
.container .box1 input[placeholder]
{
        text-align:center;
        color:#FFF;
        font-size:20px;
}


.container .box1 input[type="submit"]
{
        display:block;
        margin:10px auto;
        width:100px;
        padding:10px;
        border:none;
        font-family: cursive;
        font-size:15px;
        background-color:#FFF;
        transition:1s all linear;
}
.container .box1 input[type="submit"]:hover
{
        width:200px;
        border-radius:25px;
        border:1px solid #FFF;
        background:none;
        color:#FFF;
        cursor:pointer;
}

.container .box2
{

        width:400px;
        height:400px;
        box-sizing:border-box;
        float:right;
        box-shadow:2px 2px 12px 6px #FFF;
}
```

Java Script

Q) What is difference between Java and JavaScript?

Java	JavaScript
It is a non-scripting language.	It is a scripting language.
It is a object oriented programming language.	It is a object based programming language.
It is strongly typed checking language.	It is a weakly typed checking language.
It does not required browser window for execution.	It requires browser window for execution.
We can run individually.	We can't run individually.
It is a complex language.	It is easy language.

History of JavaScript

Original name of javascript is LiveScript.
LiveScript is developed by Netscape Corporation in late 1990's."
LiveScript was developed by using C language syntax's.
In 1995, Brenden Eich the popular scientist of Netscape Corporation renamed LiveScript to JavaScript.
Official name of JavaScript is ECMA script.
Here ECMA stands for European Computer Manufacturers Association.

Advantages of JavaScript

1) It is used to create interactive web pages.
2) It is used to display dialog boxes and popup boxes.
3) It is used to perform client side form validation.
4) It is used to add dynamic content to a web page.
5) It is used to develop responsive(dynamic) web pages.
6) It supports objects like String, Arrays, RegEx and etc.
7) It supports Date and Time.
8) It supports Cookies.
9) It supports dropdown menu. and etc.

Javascript syntax

To write javascript code we need to use <script> tag.

ex:
```
<script type="text/javascript">
    stmt1;
    stmt2;
    stmt3;
</script>
```
Here "type" attribute is optional.
Here simicolon(;) is optional.

or

ex:
```
<script >
    stmt1
    stmt2
    stmt3
</script>
```

Sublime Editor

Download link: https://www.sublimetext.com/download

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script type="text/javascript">
		document.write("Welcome to JavaScript");
	</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		document.write("Welcome to JavaScript class")
	</script>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		document.writeln("Welcome to JavaScript class");
		document.writeln("This is Ihub Talent");
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		document.writeln("Welcome to JavaScript class");
		<br>
```

```
            document.writeln("This is Ihub Talent");
        </script>
</body>
</html>
```
Note: We can't mix markup language in scripting language.

ex:
```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln("Welcome to JavaScript class");
            document.writeln("<br>");
            document.writeln("This is Ihub Talent");
        </script>
</body>
</html>
```

ex:
```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln("<h1 style='color:red'>Welcome to JavaScript class</h1>");
            document.writeln("<br>");
            document.writeln("<p style='color:blue'>This is Ihub Talent</p>");
        </script>
</body>
</html>
```
Note: If a program contains HTML, CSS and JavaScript then it is called DHTML.
DHTML stands for Dynamic Hypertext Markup language.

JavaScript Engine

JavaScript engine allows us to execute script code on browser window.
It converts user understandable scripting language to machine understandable scripting language.
By default , every browser contains javascript engine so we don't need to arrange it seperately.
We have following list of javascript engines.

ex:	Browser	JavaScript Engine	
	Chrome	V8 Engine	
	Mozilla	Spider monkey	
	Edge	Chakra	and etc.

Comments in Javascript

Comments are created for documentation purpose.
Comments are used to improve readability of our code.
In javascript comments are divided into two types.

1) Single line comment

It is used to comment a single line.

ex: // comment here

2) Multiple line comment

It is more convenient because we can comment single line and multiple lines.

```
ex:     /*
                -
                - comment here
                -
        */
```

Note: HTML -- <!-- comment here -->
CSS -- /* comment here */

Output statement in javascript

Whenever we want to display any userdefined statements or data then we need to use output statement.
We have following output statements in javascript.

1) document.writeln()
2) console.log()

1) document.writeln()

It is used to display the output on browser window.
ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                document.writeln("Hello World");
        </script>
</body>
</html>
```

2) console.log()

It is used to display the output on browser console.
In order to see browser console we need to press F12 function key.
ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>        console.log("Hello World");             </script>
</body>
</html>
```

JavaScript variables

A name which is given to a memory location is called variable.

Purpose of variable is used to store the data.

Javascript variables contains same rules as we have for identifiers.

Rule1:

A variable can starts with alphebet, underscore or dollar symbol but not with digit.

ex:

```
var  abcd;  //valid
var  _abcd; //valid
var $abcd; //valid
var  a1234; //valid
var  1abcd; //invalid
```

Rule2:

Every variable is a case sensitive.

ex:

```
var number;
var NUMBER;
var NumBer;
```

In javascript, we have two types of variables.

1) Local variable
2) Global variable

1) Local variable

A variable which is declared inside the function or block is called local variable.

A local variable we can access within the function but not outside the function.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			function f1()
			{
				//local variable
				var i=10;

				document.writeln(i+"<br>");
			}

			//calling
			f1();
	</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>

			function f1()
			{
				//local variable
				var i=10;
				document.writeln(i+"<br>");
			}
			function f2()
			{
				document.writeln(i+"<br>"); //Error
			}

			//calling
			f1();
			f2();

	</script>
</body>
</html>
```

2) Global variable

A variable which is declare outside the function or a block is called global variable.
A global variable we can access within the function and outside the function.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			//global variable
			var i=20;
			function f1()
			{
				document.writeln(i+"<br>");
			}
			//calling
			f1();
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			//global variable
			var i=20;
			function f1()
			{
				document.writeln(i+"<br>");
			}
			function f2()
			{
				document.writeln(i+"<br>");
			}
			//calling
			f1();
			f2();
	</script>
</body>
</html>
```

Note: One function can refer to another function.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			function f1()
			{
				document.writeln("F1-Function1 <br>");
				f2();
			}
			function f2()
			{
				document.writeln("F1-Function2 <br>");
			}
			//calling
			f1();
	</script>
</body>
</html>
```

Datatypes

Javascript is a dynamically typed language it means we don't need to define datatype at the time of variable declaration.

To declare a variable we will use var keyword.

Javascript internally uses javascript engine to determine perticular datatype based on the value.

In Javascript, we have two types datatypes.

1) Primitive datatypes
2) Non-Primitive datatypes

1) Primitive datatypes

We have following list of primitive datatypes.

Datatype	Description
1) number	It is used to represent numbers
2) boolean	It is used to represent boolean values.
3) String	It is used to represent string.
4) null	It is used to represent null.
5) undefined	It is used to represent undefined.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			document.writeln(i+"<br>");//10

			var j=10.5;
			document.writeln(j+"<br>");//10.5
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=true;
			document.writeln(i+"<br>");//true

			var j=false;
			document.writeln(j+"<br>");//false
	</script>
```

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

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i="hi";
			document.writeln(i+"<br>");//hi

			var j='bye';
			document.writeln(j+"<br>");//bye
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=null;
			document.writeln(i+"<br>");//null

			var j;
			document.writeln(j+"<br>");//undefined
	</script>
</body>
</html>
```

2) Non-Primitive datatypes

We have following list of non-primitive datatypes.

Datatype	**Description**
Object	It is used to create an instance through which we can access the members.
Arrays	It is used to represent similar elements.
RegEx	It is used to represent regular expression.

Types of javascript

We have two types of javascript.

1) Internal javascript / Embedded javascript
2) External javascript / Seperate javascript

1) Internal javascript

In internal javascript we will maintain html code and javascript code in .html file.

Advantages:

There is no confusion of multiple files.

We can maintain HTML code and javascript code seperately.

Disadvantages:

If code increases then it will increase complexity.

2) External javascript

In external javascript we will maintain HTML code in .html file and javascript code in .js file. But we can't execute .js file seperately in a browser window.

Advantages:

We can maintain html code in .html file and javascript code in .js file.

If code increases then it will not increase complexity.

Disadvantage:

Confusion of multiple files.

Operators

Operator is a symbol which is used to perform some operations on operands.

ex: c = a + b;

Here a , b and c are operands.

Here = and + are operators.

It can perform arithmetic operations, logical operations, conditional operations, assignment operators and etc.

We have following list of operators present in javascript.

1) Assignment operators
2) Conditional operators
3) Bitwise operators
4) Logical operators
5) Relational operators
6) Special operators

1) Assignment operators

We have following list of assignment operators.

Operator	**Description**
=	equals to
+=	addition and equals to
-=	subtraction and equals to
*=	multiplication and equals to
/=	division and equals to
%=	modules and equals to
++	incrementation
--	decremenataion

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
```

```
	<script>
			var i=10;
			document.writeln(i+"<br>");//10

			var j=10;
			j+=5;
			document.writeln(j+"<br>");//15

			var k=10;
			k-=5;
			document.writeln(k+"<br>");//5

	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			i/=2;
			document.writeln(i+"<br>");//5

			var j=10;
			j/=20;
			document.writeln(j+"<br>");//0.5

	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			i%=2;
			document.writeln(i+"<br>");//0

			var j=10;
			j%=20;
```

```
                document.writeln(j+"<br>");//10

        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var i=10;
                document.writeln(i++ + ++i+"<br>");//10 + 12 = 22

                var j=10;
                document.writeln(j-- - --j+"<br>");//10-8 =2

        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var i=10;
                var j=i++ + i++ - --i - --i;
                document.writeln(i+" "+j);//10   0

        </script>
</body>
</html>
```

2) Conditional operators

We have following list of conditional operators.

operator	description
>	greater then
>=	greater then equalsto
<	less then
<=	less then equals to
==	equals to
!=	not equals to

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			document.writeln((10>20)+"<br>");//false
			document.writeln((10>=10)+"<br>");//true
			document.writeln((10<20)+"<br>");//true
			document.writeln((10<=20)+"<br>");//true
			document.writeln((10==20)+"<br>");//false
			document.writeln((10==10)+"<br>");//true
			document.writeln((10!=20)+"<br>");//true
			document.writeln((10!=10)+"<br>");//false
	</script>
</body>
</html>
```

Q)What is the difference between = = and = = = ?

= =

It is used to check values are same or not.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			document.writeln((1 == 1)+"<br>");//true
			document.writeln((1 == true)+"<br>");//true
			document.writeln((0 == false)+"<br>");//true
			document.writeln(("1" == 1)+"<br>");//true
	</script>
</body>
</html>
```

= = =

It is used to check values and datatypes are same or not.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			document.writeln((1 === 1)+"<br>");//true
```

```
                document.writeln((1 === true)+"<br>");//false
                document.writeln((0 === false)+"<br>");//false
                document.writeln(("1" === 1)+"<br>");//false
        </script>
</body>
</html>
```

3) Assignment operators

We have following list of assignment operators.

operator	description
+=	addition and equals to
-=	subtraction and equals to
*=	multiplication and equals to
/=	division and equals to
%=	modules and equals to
==	equals to
!=	not equals to

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var i=10;
                i+=2;
                document.writeln(i); // 12
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var i=10;
                i%=2;
                document.writeln(i); // 0
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			i%=20;
			document.writeln(i); // 10
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			i/=20;
			document.writeln(i); // 0.5
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			i*=20;
			document.writeln(i); // 200
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
```

```
</head>
<body>
	<script>
			document.writeln((10 == 10) +"<br>"); // true
			document.writeln((10 == 20) +"<br>"); // false
			document.writeln((10 != 20) +"<br>" ); // true
			document.writeln((10 != 10) +"<br>" ); // false
	</script>
</body>
</html>
```

4) Logical operators

We have following list of logical operators.

operator	description
&&	logical AND
||	logical OR
!	logical NOT

Logical AND operator (&&)

Truth table

```
T	T	= T
T	F	= F
F	T	= F
F	F	= F
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			document.writeln(((5>2) && (6<10)) +"<br>"); // true
			document.writeln(((5>20) && (6<1)) +"<br>"); // false
			document.writeln(((5>20) && (6<10)) +"<br>"); // false
	</script>
</body>
</html>
```

Logical OR operator (||)

Truth table

```
T	T	= T
T	F	= T
F	T	= T
F	F	= F
```

ex:

```
<!DOCTYPE html>
<html>
```

```
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			document.writeln(((5>2) || (6<10)) +"<br>"); // true
			document.writeln(((5>20) || (6<1)) +"<br>"); // false
			document.writeln(((5>20) ||  (6<10)) +"<br>"); // true
	</script>
</body>
</html>
```

Logical NOT operator (!)

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			document.writeln((!(5>2)) +"<br>"); // false
			document.writeln((!(5>20)) +"<br>"); // true
	</script>
</body>
</html>
```

5) Bitwise Operators

We have following list of bitwise operators.

ex:

operator	description
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
>>	Right shift
<<	left shift

Bitwise AND (&)

Bitwise AND operator deals with binary numbers.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var a=10, b=5;
			var c = a & b;
```

```
                document.writeln(c); //0
                /*
                        10 - 1010
                        5  - 0101
                        -----------
                        &  - 0000
                */
        </script>
</body>
</html>
```

Bitwise OR

Bitwise OR operator deals with binary numbers.

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var a=10, b=5;
                var c = a | b;
                document.writeln(c); //15
                /*
                        10 - 1010
                        5  - 0101
                        -----------
                        |  - 1111
                */
        </script>
</body>
</html>
```

Bitwise XOR operator (^)

Bitwise XOR operator deals with binary numbers.

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var a=10, b=15;
                var c = a ^ b;
                document.writeln(c); //5
                /*
                        10 - 1010
                        15 - 1111
```

```
                        ----------
                        ^  - 0101
                */
        </script>
</body>
</html>
```

Bitwise NOT operator (~)

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var a=~10;
                document.writeln(a); //-11
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var a=~50;
                document.writeln(a); //-51
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var a=~(-10);
                document.writeln(a); //9
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var a= 10 >> 2;  //10 / 4
			document.writeln(a); //2
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var a= 100 >> 4;  // 100 / 16
			document.writeln(a); //6
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var a= 100 << 3;  // 100 * 8
			document.writeln(a); //800
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
```

```
                    var a= 10 << 6  // 10 * 64
                    document.writeln(a); //640
        </script>
</body>
</html>
```

6) Special operators

We have three types of special operators.

operator	description
?:	It is used to make a conditions
new	It is used to create a instance.
typeof	It will return type of a instance.

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                    (5>2)?document.writeln("Hi"):document.writeln("Bye");
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                    (5>20)?document.writeln("Hi"):document.writeln("Bye");//Bye
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                ((5>20) && (6<10))?document.writeln("Hi"):document.writeln("Bye"); //Bye
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var i=10;
			document.writeln(typeof(i)+"<br>"); //number
			var j=true;
			document.writeln(typeof(j)+"<br>"); //boolean
			var k="hi";
			document.writeln(typeof(k)+"<br>"); //String
			var l=null;
			document.writeln(typeof(l)+"<br>"); //object
			var m;
			document.writeln(typeof(m)+"<br>"); //undefined
	</script>
</body>
</html>
```

Q) Difference between null and undefined?

In JavaScript, undefined is a type, whereas null is an object.

undefined

It means a variable declared, but no value has been assigned a value.

Ex:

```
<script>
	var x;
	document.writeln(x); //undefined
</script>
```

null

A null in JavaScript is an assignment value.

Ex:

```
<script>
	var x=null;
	document.writeln(x); //null
</script>
```

Q) Types of functions in JavaScript?

We have three types of functions.

Named Functions

These types of functions contain name at the time of declaration.

Ex:

```
function f1()
{
document.writeln(“hello world”);
}
f1();
```

Anonymous Functions

These types of functions do not have any name.

They are declared dynamically at runtime.

Ex:
```
var f1=function()
{
document.writeln('hello world');
}
f1();
```

Arrow functions

Arrow functions are more secure when compare to other functions.
As per ES6 standards we need to use arrow function.

ex:
```
var f1=()=>
        {
                document.writeln("Hello World");
        }
```

JavaScript If Else Stmt

We have three forms of JavaScript if else stmt.

1) If stmt
2) If else stmt
3) if else if stmt

1) <u>If stmt</u>

It will evaluate the code only if our condition is true.

syntax:
```
if(condition)
{
        -
        - //code to be evaluate
        -
}
```

ex:1
```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                        if(0,1,2,3,4,5)
                        {
                                document.writeln("Hello World");
                        }
        </script>
</body>
</html>
```

ex:
```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
```

```
<body>
	<script>
			if(5,4,3,2,1,0)
			{
				document.writeln("Hello World"); // Nothing
			}
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			if(-1)
			{
				document.writeln("Hello World");
			}
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			if((5>2) && (6<10))
			{
				document.writeln("Hello World");
			}
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
```

```
                        if(!(5>2))
                        {
                                document.writeln("Hello World");//nothing
                        }
        </script>
</body>
</html>
```

2) If else stmt

It will evaluate the code either our condition is true or false.

syntax:

```
if(condition)
        {
                - // code to be evaluate if cond is true
        }
        else
        {
                - // code to be evaluate if cond is false
        }
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                        if(true)
                        {
                                document.writeln("It is true");
                        }
                        else
                        {
                                document.writeln('It is false');
                        }
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                        if(false)
                        {
                                document.writeln("It is true");
                        }
```

```
                else
                {
                        document.writeln('It is false');
                }
        </script>
</body>
</html>
```

Q) Write a javascript program to find out given number is even or odd without using modules operator (%)?

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var val=prompt("Enter the number :");

                var n=parseInt(val); // 10

                if( (n & 1) == 0)
                        document.writeln("It is even");
                else
                        document.writeln("It is odd");

                /*
                        10 - 1010
                        1  - 0001
                        ---------
                        &  - 0000
                */

        </script>
</body>
</html>
```

Q) Write a javascript program to check given number is positive or negative?

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var val=prompt("Enter the number :");

                var n=parseInt(val); // 10
```

```
		if(n>0)
			document.writeln("It is positive number");
		else
			document.writeln("It is negative number");
	</script>
</body>
</html>
```

iii) if else if statement
It will execute the code based on multiple conditions.
syntax:

```
if(cond1)
	{
		- //code to be evaluate if cond1 is true
	}
	else if(cond2)
	{
		- //code to be evaluate if cond2 is true
	}
	else if(cond3)
	{
		- //code to be evaluate if cond3 is true
	}
	else
	{
		- //code to be evaluate if all conditions are false
	}
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var val=prompt("Enter the option :");

		var n=parseInt(val); // 10

		if(n==100)
			document.writeln("It is police number");
		else if(n==103)
			document.writeln("It is enquiry number");
		else if(n==108)
			document.writeln("It is emergency number");
		else
			document.writeln("invalid option");
	</script>
</body>
</html>
```

JavaScript Switch
It is similar to if else if statement.
It will evaluate the code base on multiple conditions.
syntax: switch(condition)

```
        {
                case1 value:  //code to be execute
                        break stmt;
                case2 value:  //code to be execute
                        break stmt;
                -
                default:      //code to be execute
        }
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var val=prompt("Enter the option :");

                var n=parseInt(val); // 10

                switch(n)
                {
                        case 100: document.writeln("It is police number"); break;
                        case 103: document.writeln("It is enquiry number"); break;
                        case 108: document.writeln("It is emergency number"); break;
                        default: document.writeln("invalid option");
                }
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var ch=prompt("Enter the character :");
                switch(ch)
                {
                        case 'a': document.writeln("It is a vowel"); break;
                        case 'e': document.writeln("It is a vowel"); break;
                        case 'i': document.writeln("It is a vowel"); break;
```

```
                case 'o': document.writeln("It is a vowel"); break;
                case 'u': document.writeln("It is a vowel"); break;
                default: document.writeln("It is not a vowel");
        }
    </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        var str=prompt("Enter the String :");

        switch(str)
        {
            case "one": document.writeln("It is a January"); break;
            case "two": document.writeln("It is a February"); break;
            case "three": document.writeln("It is a March"); break;
            case "four": document.writeln("It is a April"); break;
            case "five": document.writeln("It is a May"); break;
            default: document.writeln("coming soon...");
        }
    </script>
</body>
</html>
```

JavaScript LOOPS

We have four types of loops in javascript.

1) do while loop
2) while loop
3) for loop
4) for in loop

1) do while loop

It will evaluate the code untill our condition is true.

In do while loop, our code will execute atleast for one time.

syntax:

```
do
    {
        -
        - //code to be evaluate
        -
    }while(condition);
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var i=1;
		do
		{
			document.writeln(i+" "); //1  2 3 4 5 6 7 8 9 10
			i++;
		}
		while(i<=10);
	</script>
</body>
</html>
```

Q) Write a javascript program to perform sum of 10 natural numbers?

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var i=1, sum=0;
		do
		{
			sum+=i;
			i++;
		}
		while(i<=10);
		document.writeln(sum);
	</script>
</body>
</html>
```

2) <u>while loop</u>

It will evalute the code untill our condition is true.

syntax:

```
while(condition)
	{
		-
		- //code to be evaluate
		-
	}
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var i=10;
		while(i>=1)
		{
			document.writeln(i+" ");//10 9 8 7 6 5 4 3 2 1
			i--;
		}
	</script>
</body>
</html>
```

Q) Write a javascript program to dispaly multiplication table of a given number?

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var n=5;
		var i=1;
		while(i<=10)
		{
			document.writeln(n+" * "+i+" = "+n*i+"<br>");
			i++;
		}
	</script>
</body>
</html>
```

3) <u>for loop</u>

It will evalute the code untill our condition is true.

syntax:

```
for(initialization;condition;incrementation/decrementation)
	{
		-
		- //code to be evaluate
		-
	}
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
```

```
</head>
<body>
	<script>
		for(var i=1;i<=10;i++)
		{
			document.writeln(i+" ");
		}
	</script>
</body>
</html>
```

Q) Write a javascript program to find out factorial of a given number?

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var n=5,fact=1;
		for(var i=n;i>=1;i--)
		{
			fact*=i;
		}
		document.writeln(fact);
	</script>
</body>
</html>
```

4) <u>for in loop</u>

It is used to iterate elements from array.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var arr=[10,20,30];
		for(var i in arr)
		{
			document.writeln(arr[i]+" ");
		}
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		var arr=["hi","hello","bye"]
		for(var i in arr)
		{
			document.writeln(arr[i]+" ");
		}
	</script>
</body>
</html>
```

Note:
If number of iterations are known by the user then we need to use for loop.
If number of iterations are not known by the user then we need to use while loop.
If number of iterations are not known by the user but code must execute atleast for one time then we need to use do while loop.

Javascript Functions

Function is a block of scope which is used to perform particular task.
Functions are designed to maintain business logic.
Using functions we can achieve reusability.
To declare a function in javascript we will use function keyword.
We can declare a function in javascript followed by a function name, and parenthesis().
Javascript function parenthesis contains list of arguments seperated with comma.

```
syntax: function  <function_name>()
	{
		-
		- //code to be evaluate
		-
	}
```

Functions are executed at the time when they we have requested or when event occur.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		function  f1()
		{
			document.bgColor="red";
		}
```

```
		//calling
		f1();

	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		function  f1()
		{
			for(var i=1;i<=4;i++)
			{
				for(var j=1;j<=4;j++)
				{
					document.writeln("*");
				}
				//new line
				document.writeln("<br>");
			}
		}
		//calling
		f1();
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
		function  f1()
		{
			for(var i=1;i<=4;i++)
			{
				for(var j=1;j<=i;j++)
				{
					document.writeln("*");
				}
				//new line
```

```
                document.writeln("<br>");
            }
        }
        //calling
        f1();
        f1();

    </script>
</body>
</html>
```

In javascript ,every function is a case sensitive.

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <script>
        function  f1()
        {
            document.writeln("F1 Function");
        }
        //calling
        F1();
    </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
    <button onclick="f1()"> Click Here </button>
    <script>
            function f1()
            {
                document.bgColor="blue";
            }
    </script>
</body>
</html>
```

Q) What is JavaScript Closure?
A closure is the combination of a functions bundled together along lexical scope.
In JavaScript, closures are created every time when functions are created.
In other words, a closure gives you ,access to an outer function's scope from an inner function.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			//lexical scope
			var a=10;
			function f1()
			{
				//lexical scope
				var b=20;

				function f2()
				{
					var c=30;

					document.writeln(a+" "+b+" "+c);
				}
				//calling
				f2();
			}

			//caling
			f1();
	</script>
</body>
</html>
```

Javascript Object
A javascript object is an entity which is having states and behaviours.
In general, javascript object is a collection of properties and functions.
Javascript is a object based language because everything is present in objects.
Javascript is a template based but not class based.We don't need to create a class to get the object.We can create object directly.
There are three ways to create javascript objects.

1) By using Object literal
2) By creating instance of an Object i.e using new keyword.
3) By using Object constructor i.e using new keyword.

1) <u>By using Object literal</u>

```
<!DOCTYPE html>
<html>
```

```
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			emp={
				eid:101,
				ename:"Alan Morries",
				esal:10000
			};
			document.writeln("Employee Id:"+emp.eid+"<br>");
			document.writeln("Employee Name:"+emp.ename+"<br>");
			document.writeln("Employee Salary:"+emp.esal+"<br>");
		</script>
	</body>
</html>
```

2) By creating instance of an Object

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			var emp=new Object();
			emp.eid=102;
			emp.ename="Erick Anderson";
			emp.esal=20000;
			document.writeln("Employee Id:"+emp.eid+"<br>");
			document.writeln("Employee Name:"+emp.ename+"<br>");
			document.writeln("Employee Salary:"+emp.esal+"<br>");

		</script>
	</body>
</html>
```

3) By using Object constructor

Here we need to create a function with parameters and each parameter must assign in the current object by using this keyword.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			function emp(eid,ename,esal)
			{
```

```
                    this.eid=eid;
                    this.ename=ename;
                    this.esal=esal;
                }
                e=new emp(103,"Ana Julie",30000);
                document.writeln("Employee Id :"+e.eid+"<br>");
                document.writeln("Employee Name :"+e.ename+"<br>");
                document.writeln("Employee Sal :"+e.esal+"<br>");
        </script>
    </body>
</html>
```

Javascript Array

In javascript , Array is an object which contains similar elements.
Array index always starts with '0' because it is a logical process.
There are three ways to create an array in javascript.

1) By using array literal
2) By creating instance of an array i.e using new operator.
3) By creating array constructor i.e using new operator.

1) By using array literal

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var arr=[10,20,30,40];
            for(var i=0;i<arr.length;i++)
            {
                document.writeln(arr[i]+" ");
            }
        </script>
    </body>
    </body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
    <head>
        <title>IHUB Talent</title>
    </head>
    <body>
        <script type="text/javascript">
            var arr=[10,20,30,40];
            for(var i in arr)
            {
                document.writeln(arr[i]+" ");
```

```
                }
            </script>
        </body>
        </body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
        <head>
            <title>IHUB Talent</title>
        </head>
        <body>
            <script type="text/javascript">
                var arr=["html","css","js"];
                for(var i in arr)
                {
                    document.writeln(arr[i]+" ");
                }
            </script>
        </body>
        </body>
</html>
```

2) By creating instance of an array i.e using new operator

```
<!DOCTYPE html>
<html>
        <head>
            <title>IHUB Talent</title>
        </head>
        <body>
            <script type="text/javascript">
                var arr=new Array();
                arr[0]=10;
                arr[1]=20;
                arr[2]=30;
                for(var i in arr)
                {
                    document.writeln(arr[i]+" ");
                }
            </script>
        </body>
        </body>
</html>
```

3) By creating array constructor i.e using new operator

```
<!DOCTYPE html>
<html>
        <head>
            <title>IHUB Talent</title>
```

```
	</head>
	<body>
		<script type="text/javascript">
			var arr=new Array(10,20,30,40,50);
			for(var i in arr)
			{
				document.writeln(arr[i]+" ");
			}
		</script>
	</body>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
</head>
<body>
	<script type="text/javascript">
		var arr=[];
		arr.push(10);
		arr.push(20);
		arr.push(30);
		for (i in arr)
		{
			document.write(arr[i]+" ");
		}
	</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
</head>
<body>
	<script type="text/javascript">
		var arr=[];
		arr.push(10);
		arr.push(20);
		arr.push(30);
		arr.pop();
		for (i in arr)
		{
			document.write(arr[i]+" ");
		}
```

```
	</script>
</body>
</html>
```

Javascript String
In javascript , string is an object which contains collection of characters.
There are two ways to create a string in javascript.
1) By using string literal
2) By creating instance of a string.

1) By using string literal

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			var str1="bhaskar";
			document.writeln(str1+"<br>");
			var str2='solution';
			document.writeln(str2+"<br>");
		</script>
	</body>
	</body>
</html>
```

2) By creating instance of a string.

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			var str=new String("bhaskar");
			document.writeln(str);
		</script>
	</body>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
```

```
		<script type="text/javascript">
			var str="bhaskar";
			document.writeln(str.length);
		</script>
	</body>
	</body>
</html>

ex:2
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			var str="bhaskar";
			document.writeln(str.toUpperCase());
			var str2="BHASKAR";
			document.writeln(str.toLowerCase());
		</script>
	</body>
	</body>
</html>

ex:3
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			var str1="ihub";
			var str2="talent";
			document.writeln(str1.concat(str2));
		</script>
	</body>
	</body>
</html>

ex:4
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
		<script type="text/javascript">
			var str1="ihub";
```

```
                document.writeln(str1.charAt(2));
            </script>
        </body>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
            <title>IHUB Talent</title>
        </head>
        <body>
            <script type="text/javascript">
                var str="ihub";
                var arr=str.split('');
                for(var i in arr)
                {
                    document.writeln(arr[i]+"<br>");
                }
            </script>
        </body>
        </body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
            <title>IHUB Talent</title>
        </head>
        <body>
            <script type="text/javascript">
                var str="ihub";
                var arr=str.split('');
                for(var i=arr.length-1;i>=0;i--)
                {
                    document.writeln(arr[i]);
                }
            </script>
        </body>
        </body>
</html>
```

Q) Write a javascript program to perform sum of array elements?

```
<!DOCTYPE html>
<html>
<head>
        <title>My Page!</title>
</head>
```

```
<body>
		<script>
			var arr=[10,20,30,40];
			var sum=0;
			for(var i in arr)
			{
				sum+=arr[i];
			}
			document.writeln(sum);
		</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
	<script>
			var str="racar";
			var carr=str.split('');// h  e  l   l  o
			var rev="";
			//reading reverse
			for(var i=carr.length-1;i>=0;i--)
			{
				rev+=carr[i];
			}
			if(str==rev)
				document.writeln("It is palindrome ");
			else
				document.writeln("it is not palindrome");
	</script>
</body>
</html>
```

BOM (Browser Object Model)

The Browser Object Model is used to interact with browser.

The default object for a browser is window object.It means we can call all the functions by using window or directly.

ex: window.alert("Welcome to JavaScript");

or

alert("Welcome to JavaScript");

window object

It is used to create a window on a browser.

A window object is created automically by the browser.

A "window" is a object of browser but not javascript.

Javascript objects are String,Array,Date and etc.

A "window" object is used to write programming related to browser.
With the help of window object we can perform following activities very easily.

1)It display dialog boxes and pop boxes.
2)We can find width and height of a browser.
3)We can move or resize the browser.
4)Scroll to the browser.
5)Get URL,hostname,protocol and etc of a browser.
6)We can get javascript history.

1) alert()
It will display alert dialog box. It has message with ok button.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function f1()
				{
					alert("Welcome to JavaScript");
				}

			</script>

			<button onclick="f1()">click</button>
	</body>
</html>
```

2) confirm()
It will dispaly confirm dialog box.It has message with ok button and cancel button.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function f1()
				{
					var v=confirm("Do you wants to delete ?");
					if(v==true)
					{
						alert("ok");
					}
					else
					{
						alert("cancel");
```

```
                                        }
                                }
                        </script>
                        <button onclick="f1()">delete</button>
        </body>
</html>
```

3) prompt()
It will display prompt dialog box.It contains message with textfield.
ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                        <script type="text/javascript">
                                function f1()
                                {
                                        var v=prompt("Who are you?");
                                        alert("Welcome :"+v);
                                }
                        </script>

                        <button onclick="f1()">click</button>
        </body>
</html>
```

innerWidth and innerHeight

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                        <script type="text/javascript">
                                var w=window.innerWidth;
                                var h=window.innerHeight;
                                document.writeln("Width :"+w+"<br>");
                                document.writeln("Height :"+h+"<br>");
                        </script>
        </body>
</html>
```

Note: Press "CTRL + +" for zoomin.
Press "CTRL + -" for zoomout.

4) window.open()

ex:1

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function openWindow()
				{
					window.open("http://www.google.com");
				}
			</script>
			<button onclick="openWindow()">open a new window</button>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function openWindow()
				{
					window.open("http://www.google.com","_blank");
				}
			</script>
			<button onclick="openWindow()">open a new window</button>
	</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function openWindow()
				{

	window.open("http://www.google.com","_blank","width=200px,height=200px");
				}
			</script>
```

```
                <button onclick="openWindow()">open a new window</button>
        </body>
</html>
```

close()

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
                        var myWindow;
                        function openWindow()
                        {

myWindow=window.open("http://www.google.com","","width=300px,height=300px");
                        }
                        function closeWindow()
                        {
                                myWindow.close();
                        }
                </script>

                <button onclick="openWindow()">open a new window</button>
                <button onclick="closeWindow()">close a window</button>
        </body>
</html>
```

Whenever we open a new window , it takes left top alignment.
In order to move the window we need to use moveTo() or moveBy() function.

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
                        var myWindow;
                        function openWindow()
                        {

myWindow=window.open("http://www.google.com","","width=300px,height=300px");
                        }
                        function moveWindow()
                        {
                                myWindow.moveTo(100,100);
                        }
```

```
                </script>

                <button onclick="openWindow()">open a new window</button>
                <button onclick="moveWindow()">move window</button>
        </body>
</html>
```

Note: Here we can't move window because in browser console we will get one error.

To over come this limitation we need to use custom window.

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
        </head>
        <body>
                <script type="text/javascript">
                        var myWindow;
                        function openWindow()
                        {

myWindow=window.open("","_blank","width=300px,height=300px");


                        }
                        function moveWindow()
                        {
                                myWindow.moveTo(100,100);
                        }

                </script>

                <button onclick="openWindow()">open a new window</button>
                <button onclick="moveWindow()">move window</button>
        </body>
</html>
```

Note: MoveTo() function will move from absolute position.
MoveBy() function will move from relative position.

<u>setTimeout()</u>
The setTimeout() is executed only once.
If you need repeated executions, use setInterval() instead.
ex:

```
<!DOCTYPE html>
```

```
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function f1()
				{
					setTimeout(function f1()
								{
								alert("Hello World")
								},4000);
				}
			</script>

			<button onclick="f1()">click</button>

	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
	</head>
	<body>
			<script type="text/javascript">
				function setTimeOut()
				{
					setTimeout(Anim,4000);
				}
				function Anim()
				{
					alert("Yahoo! this is javascript");
				}
			</script>

			<button onclick="setTimeOut()">click</button>

	</body>
</html>
```

clearTimeout()

The clearTimeout() method clears a timer set with the setTimeout() method.

ex:

```
<!DOCTYPE html>
<html>
```

```
<head>
	<title>IHUB Talent</title>
</head>
<body>
		<script type="text/javascript">
			var myId;
			function setTimeOut()
			{
				myId=setTimeout(Anim,4000);
			}
			function Anim()
			{
				alert("Yahoo! this is javascript");
			}
			function removeTimeOut()
			{
				clearTimeout(myId);
			}
		</script>

		<button onclick="setTimeOut()">set time</button>
		<button onclick="removeTimeOut()">remove time</button>

</body>
</html>
```

setInterval()

A setInterval() method calls a function to evaluate the expression at specified interval(milliseconds).

A setInterval() method calls continously function untill we call clearInterval() method or window is closed.

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB Talent</title>
		<style>
			div
			{
				width:150px;
				height: 150px;
				background-color: #FF0000;
			}
		</style>
	</head>
	<body>
		<script type="text/javascript">
				var a=0;
				setInterval(Anim,1000);
```

```
                        function Anim()
                        {
                                a = a + 10;
                var target=document.getElementById("myId");
                                target.style.marginLeft= a + 'px';
                        }

                </script>

                <div id="myId"></div>
        </body>
</html>
```

clearInterval()

A clearInterval() function is used to clear the timer set on setInterval() function.
An id which is return from setInterval() function will use as parameter to clearInterval().
ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>IHUB Talent</title>
                <style>
                        div
                        {
                                width:150px;
                                height: 150px;
                                background-color: #FF0000;
                        }
                </style>
        </head>
        <body>
                <script type="text/javascript">
                        var a=0;
                        var id=setInterval(Anim,1000);

                        function Anim()
                        {
                                a = a + 10;
                                if(a==100)
                                {
                                        clearInterval(id);
                                }

                                var target=document.getElementById("myId");
                                target.style.marginLeft= a + 'px';
                        }

                </script>

                <div id="myId"></div>
```

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

window history

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
	<style>
		body
		{
			height: 100vh;
			display: flex;
			justify-content: center;
			align-items: center;
		}
		a
		{
			text-decoration: none;
			color:blue;
		}
	</style>
</head>
<body>
		&laquo;
		<a href="javascript:history.back()">Previous</a>
		     
		<a href="javascript:history.forward()">Next</a>
		&raquo;
</body>
</html>
```

localStorage

A localStorage properties allows us to save key/value pairs in a browser window.
A localStorage allows us to store the data with no-expiry.It means our data will not be delete even if we close the browser.It will be present for next day.
A localStorage is a read-only.
To add the data in a localStorage we need to use setItem(key,value) function.
To read the data from localStorage we need to use getItem(key) function.
To remove perticular data from localStorage we need to use removeItem(key) function.
To remove all the data from localStorage we need to use clear() function.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
				localStorage.setItem("name","Alan");
```

```
                    localStorage.setItem("age",25);
                    document.writeln(localStorage.getItem("name")+"<br>");
                    document.writeln(localStorage.getItem("age")+"<br>");

                    localStorage.removeItem("age");
                    document.writeln(localStorage.getItem("name")+"<br>");
                    document.writeln(localStorage.getItem("age")+"<br>");

                    localStorage.clear();
                    document.writeln(localStorage.getItem("name")+"<br>");
                    document.writeln(localStorage.getItem("age")+"<br>");
            </script>
</body>
</html>
```

sessionStorage

A sessionStorage properties allows us to save key/value pair in a browser window.
A sessionStorage store the data with respect to one session.It means our data will be deleted once if we close the browser window.
To add the data in a sessionStorage we need to use setItem(key,value) function.
To read the data from sessionStorage we need to use getItem(key) function.
To remove perticular data from sessionStorage we need to use removeItem(key) function.
To remove all the data from sessionStorage we need to use clear() function.
ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
            <script>
                    sessionStorage.setItem("name","Jose");
                    sessionStorage.setItem("age",27);
                    document.writeln(sessionStorage.getItem("name")+"<br>");
                    document.writeln(sessionStorage.getItem("age")+"<br>");

                    sessionStorage.removeItem("age");
                    document.writeln(sessionStorage.getItem("name")+"<br>");
                    document.writeln(sessionStorage.getItem("age")+"<br>");

                    sessionStorage.clear();
                    document.writeln(sessionStorage.getItem("name")+"<br>");
                    document.writeln(sessionStorage.getItem("age")+"<br>");
            </script>
</body>
</html>
```

DOM

The document object represent whole HTML document.
When HTML document is loaded in a browser it represent document object.
Here HTML document is represented in a tree node hierarcy.
A document object is a root node for entire html document.
DOM always looks for three nodes.

1)Element node
2)Atribute node
3)Text node

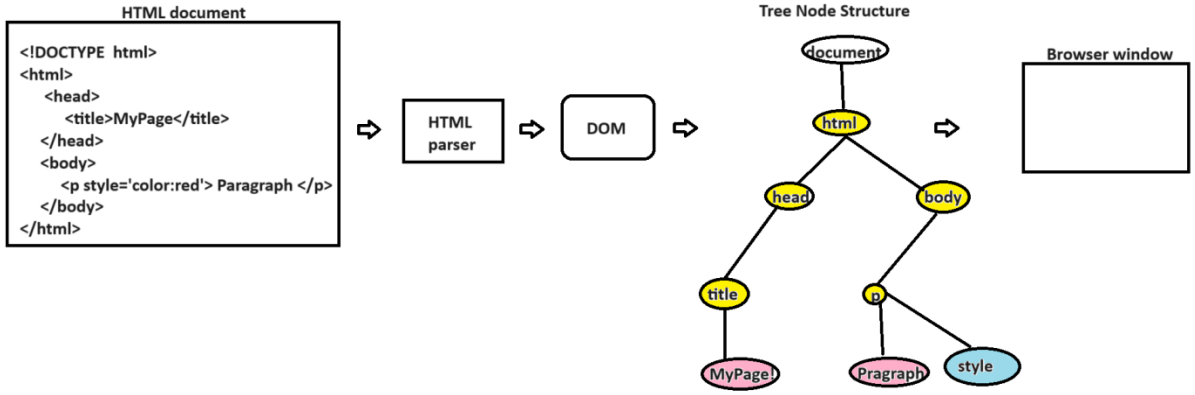

Diagram: class22.1

Using document object we can add dynamic content to the web page.
A document object is a property of window.It means we can call document object directory or by using window.

ex: window.document
or
document

document.write()

It is used to display data or custom messages without space.
ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<script type="text/javascript">
		document.write("This is First Stmt");
		document.write("This is Second Stmt");
		</script>
	</body>
</html>
```

document.writeln()

It will display the output with space at last.
ex:

```
<!DOCTYPE html>
<html>
	<head>
```

```
		<title>MyPage!</title>
	</head>
	<body>
		<script type="text/javascript">
		document.writeln("This is First Stmt");
		document.writeln("This is Second Stmt");
		</script>
	</body>
</html>
```

document.getElementById()

It is used to read the elements based on id.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		Name: <input type="text" id="t1"/>
		<button onclick="f1()"> submit </button>
		<script>
			function f1()
			{
				var val=document.getElementById('t1').value;
				document.writeln("Welcome :"+val);
			}
		</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		Name: <input type="text" id="t1"/>
		<button onclick="f1()"> submit </button>
		<br>
		<h1 id="result"></h1>
		<script>
			function f1()
			{
				var val=document.getElementById('t1').value;

				document.getElementById('result').innerHTML="Welcome
:"+val;
			}
```

```
		</script>
</body>
</html>
```

Adding two text fields and display the result using javascript

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>

		<table align="center">
			<tr>
				<td>No1:</td>
				<td><input type="text" id="t1"/></td>
			</tr>
			<tr>
				<td>No2:</td>
				<td><input type="text" id="t2"/></td>
			</tr>
			<tr>
				<td> </td>
				<td><button onclick="f1()">ADD</button></td>
			</tr>
			<tr>
				<td>Result:</td>
				<td><h2 id="result"></h2></td>
			</tr>
		</table>
		<script type="text/javascript">

			function f1()
			{
				var val1=document.getElementById('t1').value;
				var val2=document.getElementById('t2').value;
				var a=parseInt(val1)
				var b=parseInt(val2);
				var c=a+b;
				document.getElementById('result').innerHTML=c;
			}
		</script>
</body>
</html>
```

Hide and show the portion of a form page using javascript

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
```

```
</head>
<body>
		<fieldset id="cur_id">
		<legend>Current Address:</legend>
		<table align="center">
			<tr>
				<td>House No:</td>
				<td><input type="text" id="t1"/></td>
			</tr>
			<tr>
				<td>Locality:</td>
				<td><input type="text" id="t2"/></td>
			</tr>
			<tr>
				<td>State:</td>
				<td><input type="text" id="t3"/></td>
			</tr>
		</table>
		</fieldset>
		<br>

		<input type="checkbox" id="box" onclick="f1()" />
		Permanent Address same as Current Address?

		<br>
		<br>

		<fieldset id="per_id">
		<legend>Permanent Address:</legend>
		<table align="center">
			<tr>
				<td>House No:</td>
				<td><input type="text" id="t1"/></td>
			</tr>
			<tr>
				<td>Locality:</td>
				<td><input type="text" id="t2"/></td>
			</tr>
			<tr>
				<td>State:</td>
				<td><input type="text" id="t3"/></td>
			</tr>
		</table>
		</fieldset>

		<script type="text/javascript">
			function f1()
			{
				if(document.getElementById('box').checked)
				{
```

```
        document.getElementById('per_id').style.display="none";
                                }
                                else
                                {

        document.getElementById('per_id').style.display="block";
                                }
                        }
                </script>

</body>
</html>
```

Ex:-

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <fieldset id="cur_id">
                <legend>Current Address:</legend>
                <table align="center">
                        <tr>
                                <td>House No:</td>
                                <td><input type="text" id="t1"/></td>
                        </tr>
                        <tr>
                                <td>Locality:</td>
                                <td><input type="text" id="t2"/></td>
                        </tr>
                        <tr>
                                <td>State:</td>
                                <td><input type="text" id="t3"/></td>
                        </tr>
                </table>
                </fieldset>
                <br>

                <input type="checkbox" id="box" onclick="f1()" />
                Permanent Address same as Current Address?

                <br>
                <br>

                <fieldset id="per_id">
                <legend>Permanent Address:</legend>
                <table align="center">
                        <tr>
```

```
                        <td>House No:</td>
                        <td><input type="text" id="t4"/></td>
                </tr>
                <tr>
                        <td>Locality:</td>
                        <td><input type="text" id="t5"/></td>
                </tr>
                <tr>
                        <td>State:</td>
                        <td><input type="text" id="t6"/></td>
                </tr>
        </table>
        </fieldset>

        <script type="text/javascript">
                function f1()
                {
                        if(document.getElementById('box').checked)
                        {
                                var val1=document.getElementById('t1').value;
                                var val2=document.getElementById('t2').value;
                                var val3=document.getElementById('t3').value;

                                document.getElementById('t4').value=val1;
                                document.getElementById('t5').value=val2;
                                document.getElementById('t6').value=val3;
                        }
                        else
                        {
                                document.getElementById('t4').value="";
                                document.getElementById('t5').value="";
                                document.getElementById('t6').value="";
                        }
                }
        </script>

</body>
</html>
```

document.getElementsByName()

The getElementsByName() method returns a collection of elements with a specified name.

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        Name: <input type="text" name="t1"/>
```

```
		<button onclick="f1()"> click </button>
		<br>
		<h1 id="result"> </h1>

		<script>
			function f1()
			{
				var name=document.getElementsByName('t1')[0].value;
				document.getElementById('result').innerHTML=name;
			}
		</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		Name: <input type="text" name="t1"/>
		<button onclick="f1()"> click </button>
		<br>
		<h1 id="result"> </h1>
		<script>
			function f1()
			{
				var name=document.getElementsByName('t1')[0].tagName;
				document.getElementById('result').innerHTML=name;
			}
		</script>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		First Name: <input type="text" name="t1"/>
		Last Name: <input type="text" name="t1"/>
		<button onclick="f1()"> click </button>
		<br>
		<h1 id="result"> </h1>

		<script>
			function f1()
```

```
                {
                        var name=document.getElementsByName('t1').length;
                        document.getElementById('result').innerHTML=name;
                }
        </script>
</body>
</html>
```

ex:4

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <h2>Course Completed?</h2>
        <button onclick="f1()">Select All</button>
        <br>
        <input type="checkbox" name="box" value="html"/> HTML <br>
        <input type="checkbox" name="box" value="css"/> CSS <br>
        <input type="checkbox" name="box" value="js"/> JavaScript <br>
        <input type="checkbox" name="box" value="bs"/> Bootstrap <br>
        <script>
                function f1()
                {
                        var x=document.getElementsByName('box');
                        for(var i=0;i<x.length;i++)
                        {
                                if(x[i].type="checkbox")
                                {
                                        x[i].checked=true;
                                }
                        }
                }
        </script>
</body>
</html>
```

document.getElementsByTagName()

The getElementsByTagName() method returns a collection of all elements with a specified tag name.

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        Name : <input type="text" name="t1" id="t1"/>
        <button onclick="f1()"> click </button>
```

```
		<h1 id="result"></h1>
		<script>
			function f1()
			{
				var name=document.getElementsByTagName('input')[0].value;
				document.getElementById('result').innerHTML=name;
			}
		</script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<div> This is div1 </div>
		<div> This is div2 </div>
		<div> This is div3 </div>
		<button onclick="f1()"> change </button>
		<script>
			function f1()
			{
				var x=document.getElementsByTagName('div');
				x[0].innerHTML="This is change1";
				x[1].innerHTML="This is change2";
				x[2].innerHTML="This is change3";
			}
		</script>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<div> This is div1 </div>
		<div> This is div2 </div>
		<div> This is div3 </div>
		<button onclick="f1()"> change </button>
		<script>
			function f1()
			{
				var x=document.getElementsByTagName('div');
				x[0].style.textAlign="center";
```

```
                x[1].style.color="blue";
                x[2].style.backgroundColor="yellow";
        }
    </script>
</body>
</html>
```

document.getElementsByClassName()

The getElementsByClassName() method returns a collection of child elements with a given class name.

ex:1

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style>
        .c1
        {
            text-align: center;
            color:blue;
            background-color: yellow;
        }
    </style>
</head>
<body>
        <div class="c1"> This is Division Tag </div>
        <button onclick="f1()"> click </button>
        <h1 id="result"></h1>
        <script>
            function f1()
            {
                var name=document.getElementsByClassName('c1').length;
                document.getElementById('result').innerHTML=name;
            }
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style>
        .c1
        {
            text-align: center;
            color:blue;
            background-color: yellow;
        }
```

```
        </style>
</head>
<body>
                <div class="c1"> This is Division Tag </div>
                <button onclick="f1()"> click </button>
                <h1 id="result"></h1>
                <script>
                        function f1()
                        {
                                var x=document.getElementsByClassName('c1');
                                x[0].innerHTML="This is css";
                                x[0].style.backgroundColor="cyan";
                        }
                </script>
</body>
</html>
```

addEventListener()

It is used to add the handler to the function.

ex:1

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <h1> Click Here </h1>
                <script>

                        addEventListener("click",function(){
                                alert("You Have Clicked");
                        });
                </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
                <h1> Click Here </h1>
                <script>
                        addEventListener("click",f1);
                        function f1()
                        {
                                alert("Yahoo! You have clicked");
                        }
```

```
		</script>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<h1 id="hover"> Mouse Over Here  </h1>
		<br>
		<p id="result"></p>
		<script>
			var y=document.getElementById('hover');
			y.addEventListener('mouseover',f1);
			function f1()
			{
			document.getElementById('result').innerHTML="Mouse over here";
			}
		</script>
</body>
</html>
```

ex:4

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<h1 id="hover"> Mouse Over Here  </h1>
		<br>
		<p id="result"></p>
		<script>
			var y=document.getElementById('hover');
			y.addEventListener('mouseout',f1);
			function f1()
			{
			document.getElementById('result').innerHTML="Mouse out here";
			}
		</script>
</body>
</html>
```

removeEventListener()

It is used to remove the handler from function.

ex:

```
<!DOCTYPE html>
```

```
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<h1 id="hover"> Mouse Over Here  </h1>
		<button onclick="f2()"> stop </button>
		<br>
		<p id="result"></p>
		<script>
			var y=document.getElementById('hover');
			y.addEventListener('mouseover',f1);
			function f1()
			{
			document.getElementById('result').innerHTML+="<div>Mouse    over
here</div>";
			}
			function f2()
			{
				y.removeEventListener('mouseover',f1);
				document.getElementById('result').innerHTML+="Event
Stopped!";
			}
		</script>
</body>
</html>
```

Javascript program to convert feet to inches using addEventListener

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<table align="center">
			<tr>
				<td>Feet</td>
				<td> </td>
				<td>Inches</td>
			</tr>
			<tr>
				<td><input type="text" id="feet"/></td>
				<td><big>=</big></td>
				<td><input type="text" id="inches"/></td>
			</tr>
		</table>
		<script>
			var feet=document.getElementById("feet");
			var inches=document.getElementById("inches");
```

```
                feet.addEventListener('input',function(){
                        var f=this.value;
                        var i=f*12;
                        inches.value=i;
                });

                inches.addEventListener('input',function(){
                        var i=this.value;
                        var f=i/12;
                        if(!Number.isInteger(f))
                        {
                                f = f.toFixed(2);
                        }
                        feet.value=f;
                })

        </script>

</body>
</html>
```

JavaScript Date

A javascript Date object is used to display Date and time.
Using javascript javascript Date object we can display timer also.
ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var date=new Date();
                document.writeln(date);
        </script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>MyPage!</title>
</head>
<body>
        <script>
                var d=new Date();
                var dd=d.getDate();
                var mm=d.getMonth()+1;
                var yyyy=d.getFullYear();
```

```
                document.writeln(dd+"/"+mm+"/"+yyyy);
        </script>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
</head>
<body>
        <script>
                var d=new Date();
                var h=d.getHours();
                var m=d.getMinutes();
                var s=d.getSeconds();
                document.writeln(h+":"+m+":"+s);
        </script>
</body>
</html>
```

ex:4

```
<!DOCTYPE html>
<html>
<head>
    <title>MyPage!</title>
    <style>
        body
        {
            height:100vh;
            background: linear-gradient(yellow,red);
            display:flex;
            justify-content: center;
            align-items: center;
        }
        span
        {
            box-shadow:2px 2px 12px 6px #FFF;
            width:147px;
            height:42px;
            font-size:40px;
            color:#FFF;
            font-weight:bold;
        }

    </style>
</head>
<body>
        <span id="result"> </span>
```

```
		<script>
			window.onload= function(){ getTime(); }
			function getTime()
			{
				var d=new Date();
				var h=d.getHours();
				var	m=d.getMinutes();
				var s=d.getSeconds();
				m=check(m);
				s=check(s);

	document.getElementById('result').innerHTML=h+":"+m+":"+s;
				setInterval(getTime,1000);
			}

			function check(i)
			{
				if(i<10)
				{
					i = "0"+i;
				}
				return i;
			}
		</script>
</body>
</html>
```

Interview Questions

Q) What is the difference between innerText vs innerHTML ?

innerText

The innerText property is used to write the simple text using JavaScript dynamically.

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<button onclick="f1()"> click Here </button>
		<br>
		<div id="result"></div>
		<script>
			function f1()
			{
			document.getElementById('result').innerText="<p>This is javascript
innerText</p>";
			}
		</script>
</body>
</html>
```

innerHTML:
The innerHTML property is used to write the HTML code using JavaScript dynamically.
ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>MyPage!</title>
</head>
<body>
		<button onclick="f1()"> click Here </button>
		<br>
		<div id="result"></div>
		<script>
			function f1()
			{
			document.getElementById('result').innerHTML="<p
style='color:red'>This is javascript innerHTML</p>";
			}
		</script>
</body>
</html>
```

Javascript program to hide and show the password in a textfield

```
<!DOCTYPE html>
<html>
	<head>
		<title>IHUB TALENT</title>

		<!-- fontawesome icon cdn link -->
	<link		rel="stylesheet"		href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css" />

	<style type="text/css">
		.myClass
		{
			padding-right:25px;
		}
		#myId
		{
			position: relative;
			right:25px;
		}
	</style>
	</head>
	<body>
			Password: <input type="password" id="t1" class="myClass"/>
	<span class="fa fa-eye" id="myId" onclick="f1()"></span>
			<script>
				function f1()
				{
```

```
                                        var x=document.getElementById('t1');
                                        if(x.type=="password")
                                        {
                                                x.type="text";
                                        }
                                        else
                                        {
                                                x.type="password";
                                        }
                                }
                        </script>
        </body>
</html>
```

JavaScript Regular Expression

Regular expressions are patterns used to match character combinations in strings.
In JavaScript, regular expressions are also objects.

JavaScript Form validation using RegularExpression

To generate proper regular expression we can login below url.

ex: https://regex101.com/

ex:

```
<!DOCTYPE html>
<html>
<style>
input[type=text],input[type=password], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #45a049;
}

div {
```

```
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
  width:500px;
  position: relative;
  left:200px;
  top:20px;
}
</style>

<script type="text/javascript">

  function validate()
  {
    var name=document.getElementById('name').value;
    var pwd=document.getElementById('pwd').value;
    var phone=document.getElementById('phone').value;
    var email=document.getElementById('email').value;
    var country=document.getElementById('country').value;

    var namecheck=/[A-Za-z. ]{6,20}$/;

    var pwdcheck=/(?=.*[0-9])(?=.*[!@#$%^&*])(?=.*[A-Z])[a-zA-Z0-
9!@#$%^&*]{10,30}$/;

    var phonecheck=/[789][0-9]{9}$/;

    var emailcheck=/[A-Za-z.]{1,}@[A-Za-z]{2,15}[.][A-Za-z]{3,}$/;


    if(!(namecheck.test(name)))
    {
       alert("UserName must be 6 characters");
       document.getElementById('name').value="";
       document.getElementById('name').focus();
       return false;
    }

    if(!(pwdcheck.test(pwd)))
    {
      alert("password must have 1 uppercase, 1 special symbol and 1 digit");
      document.getElementById('pwd').value="";
      document.getElementById('pwd').focus();
       return false;
    }

    if(!(phonecheck.test(phone)))
    {
      alert("Phone must start with 7,8,9 series with  10 digits");
      document.getElementById('phone').value="";
```

```
    document.getElementById('phone').focus();
     return false;
   }

   if(!(emailcheck.test(email)))
   {
    alert("Please insert valid email");
    document.getElementById('email').value="";
    document.getElementById('email').focus();
     return false;
   }

   if(country=="")
   {
    alert("Please select the country option ");
    return false;
   }
   return true;
}

</script>

<body>


<div>
  <form action="/action_page.php" onsubmit="validate()">

    <label for="name">UserName</label>
    <input type="text" id="name" name="name" placeholder="Your username.."/>

    <label for="pwd">Password</label>
    <input type="text" id="pwd" name="pwd" placeholder="Your password.."/>

    <label for="phone">Phone</label>
    <input type="text" id="phone" name="phone" placeholder="Your phone.."/>

     <label for="email">Email</label>
    <input type="text" id="email" name="email" placeholder="Your email.."/>

    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="">none</option>
      <option value="australia">Australia</option>
      <option value="canada">Canada</option>
      <option value="usa">USA</option>
    </select>

    <input type="submit" value="Submit">
  </form>
```

```
</div>

</body>
</html>
```

JavaScript Interview Questions

Synchronous and Asynchronous in JavaScript

Synchronous JavaScript:

As the name suggests synchronous means to be in a sequence, i.e. every statement of the code gets executed one by one. So, basically a statement has to wait for the earlier statement to get executed.

ex:

```
<script>
   document.write("Hi"); // First
   document.write("<br>");
   document.write("IHUB TALENT") ;// Second
   document.write("<br>");
   document.write("How are you"); // Third
</script>
```

Asynchronous JavaScript:

Asynchronous code allows the program to be executed immediately where the synchronous code will block further execution of the remaining code until it finishes the current one. This may not look like a big problem but when you see it in a bigger picture you realize that it may lead to delaying the User Interface.

ex:

```
<script>
   document.write("Hi");
   document.write("<br>");
   setTimeout(function() {
      document.write("Let us see what happens");
   }, 2000);
   document.write("<br>");
   document.write("End");
   document.write("<br>");
</script>
```

Javascript promises

Promises are used to handle asynchronous operations in JavaScript.
They can handle multiple asynchronous operations easily and provide better error handling than callbacks and events.

A Promise has four states:

1. fulfilled: Action related to the promise succeeded
2. rejected: Action related to the promise failed
3. pending: Promise is still pending i.e. not fulfilled or rejected yet
4. settled: Promise has fulfilled or rejected

A promise can be created using Promise constructor.

Syntax:
```
var promise = new Promise(function(resolve, reject){
//do something
});
```

ex:1

```
<script>
var promise = new Promise(function(resolve, reject) {
        resolve('IHub Talent');
})
promise
        .then(function(successMessage) {
        //success handler function is invoked
                console.log(successMessage);
        }, function(errorMessage) {
                console.log(errorMessage);
        })
</script>
```

ex:2

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                        <script>
                                var promise= new Promise(function(resolve,reject){
                                                reject("Reject");
                                })
                                promise
                                .then(function(successMessage){
                                        document.writeln(successMessage);

                                },function(errorMessage)
                                {
                                        document.writeln(errorMessage);
                                })
                        </script>
        </body>
</html>
```

Q) What is the difference between var , let and const ?

var	**let**	**const**
It is a functional scope. We can declare without initialization.	It is a block scope. We can declare without initialization.	It is a block scope. We can't declare without initialization.
It can be updated. It can be redeclared.	It can be updated. It can't be redeclared.	It can't be updated. It can't be redeclared.
It can be accessible without declaration and default value is undefined.	It can't be accessible without declaration.	It can't be accessible without declaration.

initialization
ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
            var i;
            document.writeln(i); //undefined
      </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
            let i;
            document.writeln(i); //undefined
      </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
            const i;
            document.writeln(i); //invalid
      </script>
</body>
</html>
```

update
ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
```

```
<body>
      <script>
         var i=10;
         i=20;
         document.writeln(i); //20
      </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
         let i=10;
         i=20;
         document.writeln(i); //20
      </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
         const i=10;
         i=20;
         document.writeln(i); //invalid
      </script>
</body>
</html>
```

Redeclared

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
         var i=10;
         var i=20;
```

```
            document.writeln(i); //20
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
        <script>
            let i=10;
            let i=20;
            document.writeln(i); //invalid
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
        <script>
            const i=10;
            const i=20;
            document.writeln(i); //invalid
        </script>
</body>
</html>
```

Accessible

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
        <script>
            document.writeln(i); //undefined
            var i=10;
        </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
       <script>
          document.writeln(i); //invalid
          let i=10;
       </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
       <script>
          document.writeln(i); //invalid
          const i=10;
       </script>
</body>
</html>
```

Q) What is javascript hoisting?

Hoisting is the default behavior of JavaScript where all the variable and function declarations are moved on top.

This means that irrespective of where the variables and functions are declared, they are moved on top of the scope.

ex:1

```
 i=10;                          var i;
document.writeln(i); ==>        i=10;
var i;                          document.writeln(i);
```

ex:2

```
f1();                                          function f1()
                                               {
function f1()                                          document.writeln("Hello");
{                                              }
     document.writeln("Hello");      ==>
}                                              f1();
```

Q) What is spread operator in JavaScript?

The spread operator is used to spreading an array.

```
<!DOCTYPE html>
<html>
<head>
```

```
   <title>MyPage!</title>
</head>
<body>
      <script>
         var sum=0;
         function f1(num1,num2,num3,num4)
         {
            sum=sum+num1+num2+num3+num4;
            document.writeln(sum);
         }
         let arr=[10,20,30,40];
         f1(...arr);
      </script>
</body>
</html>
```

Q) What is javascript recursion?
A function which call itself for many number of times is called recursion.

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
         function f1(i)
         {
               if(i<=10)
               {
                  document.writeln(i);// 1 2 3 4 5 6 7 8 9 10
                  f1(i+1);
               }
         }
         f1(1);
      </script>
</body>
</html>
```

Q) What is map() method in javascript?
A map() creates a new array from calling a function for every array element.

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
      <script>
         var arr=[10,20,30];
         var newArr=arr.map(f1);
         function f1(ele)
```

```
        {
           return ele;
        }
        document.writeln(newArr);
    </script>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
        </head>
        <body>
                        <script>
                                var arr=[10,20,30];
                                var newArr=arr.map(f1);
                                function f1(ele)
                                {
                                        return ele+100;
                                }
                                document.writeln(newArr);
                        </script>
        </body>
</html>
```

Q) What is reduce function in javascript?

The reduce() method executes a reducer function for array element.

The reduce() method returns a single value.

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>
</head>
<body>
    <script>
        var arr=[100,20,30];
        var res=arr.reduce(f1);
        function f1(total,ele)
        {
            return total-ele;
        }
        document.writeln(res);
    </script>
</body>
</html>
```

Javascript OOPS

OOPS stands for Object oriented Programming System/Structure.

A technology is said to be object oriented if it support following features.

class
object
abstraction
encapsulation
inheritance
and
polymorphism.

Javascript is not a object oriented programming language. It is a object based programming language because javascript contains objects like Arrays, Strings, RegEx, Date and etc.

class

It is a template for an object.

In javascript, class will not consider as an object.

To declare a class we need to use class keyword following by any perticular name.

As per naming conventions in javascript , A class name must starts with uppercase letter.

```
ex:     class  Example
        {
                -
                -
                -
        }
```

constructor

A constructor is a special function which is used to initialized or create an object.

A constructor will be called when memory is allocated.

To declare a constructor we will use constructor keyword.

A class can have only constructor.

```
ex:     class  Example
        {
                constructor()
                {
                        -
                        -
                        -
                }
        }
```

object

Object is an entity which is having state and behaviours (properties and functions).

To create object in javascript we will use new keyword.

```
syntax:         var  ref_var=new Object_name();
ex:             var  e = new Example();
```

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
```

```
	</head>
	<body>
		<script>
			class Example
			{
				constructor()
				{
					document.writeln("constructor <br>");
				}
			}
			var e1=new Example();
			var e2=new Example();
		</script>
	</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
	</head>
	<body>
		<script>
			class Example
			{
				constructor(id,name,sal)
				{
					document.writeln("Employee Id :"+id+"<br>");
					document.writeln("Employee Name :"+name+"<br>");
					document.writeln("Employee Salary :"+sal+"<br>");
				}
			}
			var e1=new Example(101,'Alan',10000);
			var e2=new Example(102,'Jose',20000);
		</script>
	</body>
</html>
```

Abstraction

Hiding internal implementation and highlighting the set of services is called abstraction.
The best example of abstraction is GUI ATM machine where bank people will hide internal implementation and highlights the set of services like banking,withdrawl,mini stmt and etc.

Encapsulation

The process of encapsulating or grouping properties and it's associate functions is called encapsulation.
In encapsulation for every property we need to declare setter and getter methods.

```
<!DOCTYPE html>
<html>
```

```
<head>
	<title>MyPage!</title>
</head>
<body>
		<script>
			class Example
			{
				//setter method
				setId(id)
				{
					this.id=id;
				}
				setName(name)
				{
					this.name=name;
				}
				setSalary(sal)
				{
					this.sal=sal
				}

				//getter methods
				getId()
				{
					return this.id;
				}
				getName()
				{
					return this.name;
				}
				getSalary()
				{
					return this.sal;
				}
			}
			var e=new Example();
			e.setId(201);
			e.setName("Lara");
			e.setSalary(10000);
			document.writeln("Employee Id :"+e.getId()+"<br>");
			document.writeln("Employee Name :"+e.getName()+"<br>");
			document.writeln("Employee Salary :"+e.getSalary()+"<br>");
		</script>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
```

```
        <title>MyPage!</title>
    </head>
    <body>
            <script>
                class Example
                {
                    constructor(id,name,sal)
                    {
                        this.id=id;
                        this.name=name;
                        this.sal=sal;
                    }
                    //getter methods
                    getId()
                    {
                        return this.id;
                    }
                    getName()
                    {
                        return this.name;
                    }
                    getSalary()
                    {
                        return this.sal;
                    }
                }
                var e=new Example(301,"Nelson",20000);
                document.writeln("Employee Id :"+e.getId()+"<br>");
                document.writeln("Employee Name :"+e.getName()+"<br>");
                document.writeln("Employee Salary :"+e.getSalary()+"<br>");
            </script>
    </body>
</html>
```

Inheritance

Inheritance is a mechanism where one class will inherit the properties of another class.
Inheritance is a mechanism where one class will derive in the presence of another class.

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
    </head>
    <body>
            <script>
                class A
                {
                    f1()
                    {
                        document.writeln("A-F1 Function <br>");
                    }
```

```
                }
                class B extends A
                {
                        f2()
                        {
                                document.writeln("B-F2 Function <br>");
                        }
                }
                var a=new A();
                a.f1();
                var b=new B();
                b.f1();
                b.f2();
        </script>
    </body>
</html>
```

Polymorphism

Polymorphism has taken from Greek word.
Poly means many and morphism means forms.
The ability to represent in different forms is called polymorphism.

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
    </head>
    <body>
        <script>
            class A
            {
                display()
                {
                    document.writeln("A-class <br>");
                }
            }
            class B extends A
            {
                display()
                {
                    document.writeln("B-class <br>");
                }
            }
            var a=new A();
            a.display(); // A-class

            var b=new B();
            b.display(); // B-class
        </script>
    </body>
</html>
```

Bootstrap5

It is a most popular HTML, CSS and Javascript framework developed by twitter for developing web applications.
It is absolutely free to download and use.
It is developed by Mark Otto and Jacob Thornton at Twitter.
It is released as open source product on GITHUB in 2011.
Bootstrap contains HTML and CSS based designed templates for typography (fonts), forms, tables, list and etc.
It also contains javascript plugin's.
Bootstrap4 was released in the month of Augest 18, 2018.
Bootstrap5 which is extension of Bootstrap4 was release in May 5 , 2021.

Advantages of Bootstrap

- It is easy to learn and easy to use.
- It is used to develop responsive designs.
- It saves lot of development time.
- It is a consistent framework.
- Good Documentation and Community support.
- It supports greate Grid System.
- It supports javascript plugins along with Jquery.

Boostrap color

By using utility classes change the color of a text.

```
<!DOCTYPE html>
<html>
<head>
	<title>mypage!</title>
	<!-- bootstrap cdn link -->
	<meta charset="utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<link  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

</head>
<body>
	<h1 class="text-primary"> text primary </h1>
	<h1 class="text-secondary"> text secondary </h1>
	<h1 class="text-info"> text info </h1>
	<h1 class="text-warning"> text warning </h1>
	<h1 class="text-success"> text success </h1>
	<h1 class="text-danger"> text danger </h1>
	<h1 class="text-light"> text light </h1>
	<h1 class="text-dark"> text dark </h1>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>mypage!</title>
	<!-- bootstrap cdn link -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>

   <style type="text/css">
	.text-brown
	{
		color:#B33771;
	}
   </style>

</head>
<body>
	<h1 class="text-primary"> text primary </h1>
	<h1 class="text-secondary"> text secondary </h1>
	<h1 class="text-info"> text info </h1>
	<h1 class="text-warning"> text warning </h1>
	<h1 class="text-success"> text success </h1>
	<h1 class="text-danger"> text danger </h1>
	<h1 class="text-light"> text light </h1>
	<h1 class="text-dark"> text dark </h1>
	<h1 class="text-brown"> text Brown </h1>
</body>
</html>
```

Bootstrap background

In bootstrap background color can be applied to any element/tag.

```
<!DOCTYPE html>
<html>
<head>
	<title>mypage!</title>
	<!-- bootstrap cdn link -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
   <script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>
   <style type="text/css">
         .bg-brown
         {
                  background-color:#B33771;
         }
   </style>
</head>
<body>
         <h1 class="bg-primary"> text primary </h1>
         <h1 class="bg-secondary"> text secondary </h1>
         <h1 class="bg-info"> text info </h1>
         <h1 class="bg-warning"> text warning </h1>
         <h1 class="bg-success"> text success </h1>
         <h1 class="bg-danger"> text danger </h1>
         <h1 class="bg-light"> text light </h1>
         <h1 class="bg-dark"> text dark </h1>
         <h1 class="bg-brown"> text Brown </h1>
</body>
</html>
```

Boostrap border

Border utility is used to change quickly the border-style and border-radius of an element.
It is mainly used for images, buttons and etc.
we can use border classes to an element to remove all borders or add some borders.
Add classes to an element to easily rounds its corners.

```
<!DOCTYPE html>
<html>
<head>
         <title>mypage!</title>
         <!-- bootstrap cdn link -->
         <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>
</head>
<body>
         <span class="border"> span1 tag</span>
         <span class="border-top"> span2 tag </span>
         <span class="border-start"> span3 tag </span>
         <span class="border-bottom"> span4 tag </span>
         <span class="border-end"> span5 tag </span>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>mypage!</title>
	<!-- bootstrap cdn link -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>

</head>
<body>
	<span class="border border-primary"> span1 tag</span>
	<span class="border-top border-info"> span2 tag </span>
	<span class="border-start border-danger"> span3 tag </span>
	<span class="border-bottom border-success"> span4 tag </span>
	<span class="border-end border-dark"> span5 tag </span>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
	<title>mypage!</title>
	<!-- bootstrap cdn link -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>


</head>
<body>
		<img src="images/rock.png"
				width="200px"
					height="200px"
						class="border border-dark rounded"/>
```

```
<img src="images/rock.png"
        width="200px"
            height="200px"
                class="border border-dark rounded-top"/>

<img src="images/rock.png"
        width="200px"
            height="200px"
                class="border border-dark rounded-start"/>

<img src="images/rock.png"
        width="200px"
            height="200px"
                class="border border-dark rounded-bottom"/>

<img src="images/rock.png"
        width="200px"
            height="200px"
                class="border border-dark rounded-end"/>

</body>
</html>
```

Boostrap sizing

IT is used to size the element by using sizing utility classes.
we can easily make an element wide or a toll with our width and height utility.
It suports for 25%,50%,75% and 100% bydefault.

```
<!DOCTYPE html>
<html>
<head>
   <title>MyPage!</title>

   <!-- add bootstrap plugins -->

   <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

   <style>
      .w-60
      {
         width:60%;
      }
   </style>
```

```
</head>
<body>
	<br>
	<div class="bg-primary w-25">Div1</div>
	<div class="bg-info w-50">Div2</div>
	<div class="bg-warning w-75">Div3</div>
	<div class="bg-success w-100">Div4</div>
	<div class="bg-danger w-60">Div5</div>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
		<title>mypage!</title>
		<!-- bootstrap cdn link -->
		<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link		href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>
</head>
<body>
			<img src="images/rock.png"
					class="border border-dark rounded w-25 h-25"/>
			<br>
			<img src="images/rock.png"
					class="border border-dark rounded w-50 h-50"/>
			<br>
			<img src="images/rock.png"
					class="border border-dark rounded w-75 h-75"/>
			<br>
			<img src="images/rock.png"
					class="border border-dark rounded w-100 h-100"/>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
		<title>mypage!</title>
		<!-- bootstrap cdn link -->
		<meta charset="utf-8">
```

```
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>

        <style type="text/css">
                .w-60
                {
                        width:60%;
                }
                .h-60
                {
                        height:60%;
                }
        </style>
</head>
<body>
                <img src="images/rock.png"
                                class="border border-dark rounded w-25 h-25"/>
                <br>
                <img src="images/rock.png"
                                class="border border-dark rounded w-50 h-50"/>
                <br>
                <img src="images/rock.png"
                                class="border border-dark rounded w-60 h-60"/>
                <br>
                <img src="images/rock.png"
                                class="border border-dark rounded w-75 h-75"/>
                <br>
                <img src="images/rock.png"
                                class="border border-dark rounded w-100 h-100"/>

</body>
</html>
```

Boostrap containers

Containers are used to pad the content inside of them.
In boostrap there are two containers classes available.

1. container-fluid
2. container

```
<!DOCTYPE html>
<html>
<head>
        <title>mypage!</title>
        <!-- bootstrap cdn link -->
        <meta charset="utf-8">
```

```
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
   <script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>
</head>
<body>
                <div class="container-fluid bg-info">
                        <h1>Heading Tag</h1>
                        <p>Paragraph Tag</p>
                </div>
</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
<head>
        <title>mypage!</title>
        <!-- bootstrap cdn link -->
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
   <script    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
></script>
</head>
<body>
                <div class="container bg-warning">
                        <h1>Heading Tag</h1>
                        <p>Paragraph Tag</p>
                </div>
</body>
</html>
```

Bootstrap5 Display

Bootstrap5 contains following list of properties.

ex: d-none
d-block
d-inline
d-inline-block

1) d-none

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
	</head>
	<body>
			<div class="container">
				<h1 class="d-none"> This is Heading Tag </h1>
				<p> This is Paragraph Tag </p>
			</div>
	</body>
</html>
```

2) d-block

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<meta charset="utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
	</head>
	<body>
			<div class="container">
				<span class="d-block bg-warning">This is span1</span>
				<span class="d-block bg-info">This is span2</span>
				<span class="d-block bg-primary">This is span3</span>
			</div>
	</body>
</html>
```

3) d-inline

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<meta charset="utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
	</head>
	<body>
			<div class="container">
				<div class="d-inline bg-primary">This is div1</div>
				<div class="d-inline bg-success">This is div2</div>
```

```
                    <div class="d-inline bg-info">This is div3</div>
                </div>
    </body>
</html>
```

4) d-inline-block

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>
        <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
    </head>
    <body>
        <div class="container">
            <span class="bg-primary d-inline-block w-25">This is span1</span>
            <span class="bg-success d-inline-block w-25">This is span2</span>
            <span class="bg-info d-inline-block w-25">This is span3</span>
        </div>
    </body>
</html>
```

Bootstrap spacing

Spacing utility is used to assign the responsive friendly margin or padding values to an element or a subset of its sides with shorthand classes.
It includes individual properties , all properties ,vertical properties and horizontal properties.

bootstrap margin

m - for classes that set margin
Where sides is one of:
t - for classes that set margin-top or padding-top
b - for classes that set margin-bottom or padding-bottom
s - (start) for classes that set margin-left or padding-left in LTR, margin-right or padding-right in RTL
e - (end) for classes that set margin-right or padding-right in LTR, margin-left or padding-left in RTL
x - for classes that set both *-left and *-right
y - for classes that set both *-top and *-bottom
blank - for classes that set a margin or padding on all 4 sides of the element
Where size is one of:

0 - for classes that eliminate the margin or padding by setting it to 0
1 - (by default) for classes that set the margin or padding to $spacer * .25
2 - (by default) for classes that set the margin or padding to $spacer * .5
3 - (by default) for classes that set the margin or padding to $spacer *
4 - (by default) for classes that set the margin or padding to $spacer * 1.5

5 - (by default) for classes that set the margin or padding to $spacer * 3
auto - for classes that set the margin to auto

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>

	</head>
	<body>
		<div class="container">
			<h1 class="border border-primary m-1">This is heading 1</h1>
			<h1 class="border border-primary m-2">This is heading 2</h1>
			<h1 class="border border-primary m-3">This is heading 3</h1>
			<h1 class="border border-primary m-4">This is heading 4</h1>
			<h1 class="border border-primary m-5">This is heading 5</h1>
		</div>
	</body>
</html>
```

ex:

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<meta charset="utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
	</head>
	<body>
		<div class="container-fluid">
			<h1 class="border border-primary mt-5">This is heading 1</h1>
			<h1 class="border border-primary mb-5">This is heading 2</h1>
			<h1 class="border border-primary ms-5">This is heading 3</h1>
			<h1 class="border border-primary me-5">This is heading 4</h1>
			<h1 class="border border-primary mx-5">This is heading 5</h1>
```

```
                <h1 class="border border-primary my-5">This is heading 6</h1>
            </div>
        </body>
</html>
```

Bootstrap padding

p - for classes that set padding
Where sides is one of:
t - for classes that set margin-top or padding-top
b - for classes that set margin-bottom or padding-bottom
s - (start) for classes that set margin-left or padding-left in LTR, margin-right or padding-right in RTL
e - (end) for classes that set margin-right or padding-right in LTR, margin-left or padding-left in RTL
x - for classes that set both *-left and *-right
y - for classes that set both *-top and *-bottom
blank - for classes that set a margin or padding on all 4 sides of the element
Where size is one of:

0 - for classes that eliminate the margin or padding by setting it to 0
1 - (by default) for classes that set the margin or padding to $spacer * .25
2 - (by default) for classes that set the margin or padding to $spacer * .5
3 - (by default) for classes that set the margin or padding to $spacer
4 - (by default) for classes that set the margin or padding to $spacer * 1.5
5 - (by default) for classes that set the margin or padding to $spacer * 3
auto - for classes that set the margin to auto

ex:1

```
<!DOCTYPE html>
<html>
        <head>
                <title>MyPage!</title>
                <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
        </head>
        <body>
                        <div class="container-fluid">
                                <h1 class="border border-primary p-1">This is heading 1</h1>
                                <h1 class="border border-primary p-2">This is heading 2</h1>
                                <h1 class="border border-primary p-3">This is heading 3</h1>
                                <h1 class="border border-primary p-4">This is heading 4</h1>
                                <h1 class="border border-primary p-5">This is heading 5</h1>
                        </div>
        </body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>
		<meta charset="utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
	</head>
	<body>
		<div class="container-fluid">
			<h1 class="border border-primary pt-5">This is heading 1</h1>
			<h1 class="border border-primary pb-5">This is heading 2</h1>
			<h1 class="border border-primary ps-5">This is heading 3</h1>
			<h1 class="border border-primary pe-5">This is heading 4</h1>
			<h1 class="border border-primary px-5">This is heading 5</h1>
			<h1 class="border border-primary py-5">This is heading 6</h1>
			</div>
	</body>
</html>
```

Bootstrap 5 Grid System

Bootstrap's grid system is built with flexboxes.

To create a row in a grid system we need to use "row" class.

It allows to create 12 columns in a single row and It is not required that we must use all 12 columns.

The Grid system is responsive and the column will rearrange automatically dependent upon the screen size.

Grid class

According Bootstrap5 ,A grid system contains 6 classes.

- col- - extra small device
- col-sm- - small devices
- col-md- - medium devices
- col-lg- - large devices
- col-xl- - xlarge devices.
- col-xxl- - too xlarge device

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container-fluid mt-5">
                        <div class="row">
                                <!-- 1 column occupies 12 column space -->
                                <div class="col bg-primary">column1</div>
                        </div>
                </div>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container-fluid mt-5">
                        <div class="row">
                                <!-- 2 columns occupies 6 column space -->
                                <div class="col bg-primary">column1</div>
                                <div class="col bg-info">column2</div>
                        </div>
                </div>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
```

```
   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container-fluid mt-5">
                        <div class="row">
                                <!-- 3 columns occupies 4 column space -->
                                <div class="col bg-primary">column1</div>
                                <div class="col bg-info">column2</div>
                                <div class="col bg-warning">column3</div>
                        </div>
                </div>
</body>
</html>

ex:4
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>
        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link                                    zhref="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container-fluid mt-5">
                        <div class="row">
                                <!-- 4 columns occupies 3 column space -->
                                <div class="col bg-primary">column1</div>
                                <div class="col bg-info">column2</div>
                                <div class="col bg-warning">column3</div>
                                <div class="col bg-success">column4</div>
                        </div>
                </div>
</body>
</html>

ex:5
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">
```

```
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

</head>
<body>
                <div class="container-fluid mt-5">

                        <div class="row">
                                <!-- 5 columns occupies 3 column space -->
                                <div class="col bg-primary">column1</div>

                                <div class="col bg-info">column2</div>

                                <div class="col bg-warning">column3</div>

                                <div class="col bg-success">column4</div>

                                <div class="col bg-danger">column5</div>
                        </div>

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

ex:6
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

</head>
<body>
                <div class="container-fluid mt-5">
```

```
                        <div class="row">

                                <div class="col-2 bg-primary">column1</div>

                                <div class="col-4 bg-info">column2</div>

                                <div class="col-6 bg-warning">column3</div>

                        </div>

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

ex:7

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

</head>
<body>
                <div class="container-fluid mt-5">

                        <div class="row">
                                <div class="col-lg-4 bg-primary">column1</div>
                                <div class="col-lg-4 bg-info">column2</div>
                                <div class="col-lg-4 bg-warning">column3</div>
                        </div>
                        <br><br>
                        <div class="row">
                                <div class="col-lg-4 col-md-4 bg-primary">column1</div>
                                <div class="col-lg-4 col-md-4 bg-info">column2</div>
                                <div class="col-lg-4 col-md-4 bg-warning">column3</div>
                        </div>
                        <br><br>
                        <div class="row">
                                <div        class="col-lg-4        col-md-4        col-sm-4        bg-
primary">column1</div>
```

```
                        <div class="col-lg-4 col-md-4 col-sm-4 bg-info">column2</div>
                        <div class="col-lg-4 col-md-4 col-sm-4 bg-warning">column3</div>
                    </div>

                    <br><br>
                    <div class="row">
                        <div class="col-lg-4 col-md-4 col-sm-4 col-12 bg-primary">column1</div>
                        <div class="col-lg-4 col-md-4 col-sm-4 col-12 bg-info">column2</div>
                        <div class="col-lg-4 col-md-4 col-sm-4 col-12 bg-warning">column3</div>
                    </div>

            </div>
</body>

</html>
```

Bootstrap images
3 classes to display the images.
1)rounded image (rounded)
2)circle image (rounded-circle)
3)thumbnail (img-thumbnail)

1)<u>rounded image</u>

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>

    <!-- add bootstrap 5 plugins -->
    <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
            <div class="container-fluid mt-5">
                <img src="images/rock.jpg" class="w-25 h-25 border border-secondary rounded"/>
            </div>
</body>
</html>
```

2)circle image

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

</head>
<body>
		<div class="container-fluid mt-5">
			<img src="images/rock.jpg" class="w-25 h-25 border border-secondary rounded-circle"/>
		</div>
</body>
</html>
```

3)thumbnail image

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

</head>
<body>
		<div class="container-fluid mt-5">
			<img src="images/rock.jpg" class="w-25 h-25 img-thumbnail"/>
		</div>
</body>
</html>
```

Bootstrap float

Bootstrap float is used for alignment of an element.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link 	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
  <script 	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
	<div class="container-fluid mt-5">
		<img src="images/rock.jpg" class="w-25 h-25 img-thumbnail float-start"/>
		<img src="images/rock.jpg" class="w-25 h-25 img-thumbnail float-end"/>
	</div>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link 	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
  <script 	src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container-fluid mt-5">
			<img src="images/rock.jpg"
			class="w-25 h-25 img-thumbnail m-auto d-block"/>
		</div>
</body>
</html>
```

Bootstrap Design

```
<!DOCTYPE html>
<html>
<head>
```

```
        <title>IHUB TALENT</title>
        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >
   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container-fluid mt-5">
                        <div class="row">
                                <div class="col-lg-4 col-md-4  col-sm-6 col-12">
                                        <img src="images/rock.jpg"
                                         class="w-100 h-100 img-thumbnail"/>
                                </div>
                                <div class="col-lg-4 col-md-4 col-sm-6 col-12">
                                        <img src="images/rock.jpg"
                                        class="w-100 h-100 img-thumbnail"/>
                                </div>
                                <div class="col-lg-4 col-md-4 col-sm-6 col-12">
                                        <img src="images/rock.jpg"
                                        class="w-100 h-100 img-thumbnail"/>
                                </div>
                        </div>
                </div>
</body>
</html>
```

Bootstrap tables

Tables are used to store the data in the form of rows and columns.
A "table" class is used to represent a table in bootstrap.

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container mt-5">

                        <table class="table">
```

```
                        <tr>
                                <th>ID</th>
                                <th>NAME</th>
                                <th>ADDRESS</th>
                        </tr>
                        <tr>
                                <td>101</td>
                                <td>Alan</td>
                                <td>USA</td>
                        </tr>
                        <tr>
                                <td>102</td>
                                <td>Jose</td>
                                <td>UAE</td>
                        </tr>
                        <tr>
                                <td>103</td>
                                <td>Nelson</td>
                                <td>UK</td>
                        </tr>
                </table>

        </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
        <div class="container mt-5">

                <table class="table table-bordered">
                        <tr>
                                <th>ID</th>
                                <th>NAME</th>
                                <th>ADDRESS</th>
```

```
                    </tr>
                    <tr>
                        <td>101</td>
                        <td>Alan</td>
                        <td>USA</td>
                    </tr>
                    <tr>
                        <td>102</td>
                        <td>Jose</td>
                        <td>UAE</td>
                    </tr>
                    <tr>
                        <td>103</td>
                        <td>Nelson</td>
                        <td>UK</td>
                    </tr>
                </table>

            </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
            <div class="container mt-5">

                <table class="table table-borderless">
                    <tr>
                        <th>ID</th>
                        <th>NAME</th>
                        <th>ADDRESS</th>
                    </tr>
                    <tr>
                        <td>101</td>
                        <td>Alan</td>
```

```
                    <td>USA</td>
                </tr>
                <tr>
                    <td>102</td>
                    <td>Jose</td>
                    <td>UAE</td>
                </tr>
                <tr>
                    <td>103</td>
                    <td>Nelson</td>
                    <td>UK</td>
                </tr>
            </table>

        </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
        <div class="container mt-5">

            <table class="table table-bordered table-striped">
                <tr>
                    <th>ID</th>
                    <th>NAME</th>
                    <th>ADDRESS</th>
                </tr>
                <tr>
                    <td>101</td>
                    <td>Alan</td>
                    <td>USA</td>
                </tr>
                <tr>
                    <td>102</td>
```

```
                                <td>Jose</td>
                                <td>UAE</td>
                        </tr>
                        <tr>
                                <td>103</td>
                                <td>Nelson</td>
                                <td>UK</td>
                        </tr>
                </table>

        </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
        <div class="container mt-5">

                <table class="table table-bordered table-hover">
                        <tr>
                                <th>ID</th>
                                <th>NAME</th>
                                <th>ADDRESS</th>
                        </tr>
                        <tr>
                                <td>101</td>
                                <td>Alan</td>
                                <td>USA</td>
                        </tr>
                        <tr>
                                <td>102</td>
                                <td>Jose</td>
                                <td>UAE</td>
                        </tr>
                        <tr>
```

```
                                <td>103</td>
                                <td>Nelson</td>
                                <td>UK</td>
                        </tr>
                </table>

        </div>

</body>
</html>
```

Bootstrap list

It is used to represent list of items.
A list-group is applicable for unorder list.
A list-group-item is applicable for list of items.

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
        <div class="container mt-5">
                Courses:
                <ul class="list-group">
                        <li class="list-group-item">HTML</li>
                        <li class="list-group-item">CSS</li>
                        <li class="list-group-item">JavaScript</li>
                        <li class="list-group-item">Boostrap</li>
                </ul>

        </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
```

```
	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">
			Courses:
			<ul class="list-group">
				<li class="list-group-item active">HTML</li>
				<li class="list-group-item">CSS</li>
				<li class="list-group-item">JavaScript</li>
				<li class="list-group-item disabled">Boostrap</li>
			</ul>

		</div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">
			Courses:
			<ul class="list-group list-group-flush">
				<li class="list-group-item ">HTML</li>
				<li class="list-group-item">CSS</li>
				<li class="list-group-item">JavaScript</li>
				<li class="list-group-item ">Boostrap</li>
			</ul>

		</div>
```

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

Bootstrap Badgets

Badges can be used as part of links or buttons to provide a counter.

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" >

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">

			<span class="badge bg-primary">2</span> Notification
			<span class="badge bg-info">5</span> Email
			<span class="badge bg-warning">1K</span> Subscribes
			<span class="badge bg-success">1M</span> Likes

		</div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" >

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
```

```
		<div class="container mt-5">

			<span class="badge bg-primary">Advanced</span> Java
			<span class="badge bg-info">Basics</span> Oracle
			<span class="badge bg-warning">Intermediate</span> ReactJS
			<span class="badge bg-success">Optional</span> Spring Boot

		</div>

</body>
</html>
```

Bootstrap alerts

A "alert" class is used to represent bootstrap alert.
We can use alert class along with contextual classes.
ex: alert-danger
alert-info
alert-success
alert-warning
and etc.

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">

			<div class="alert alert-danger">
				<strong>Sorry!</strong> Data is deleted
			</div>

			<div class="alert alert-success">
				<strong>Yahoo!</strong> Account is created
			</div>

			<div class="alert alert-warning">
				<strong>Listen!</strong> Don't learn fastly
			</div>
```

```
                        <div class="alert alert-info">
                                <strong>God!</strong> Very difficult
                        </div>

                </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
                <div class="container mt-5">

                        <div class="alert alert-danger collapse">
                                <strong>Sorry!</strong> Data is deleted
                        </div>

                        <div class="alert alert-success">
                                <strong>Yahoo!</strong> Account is created
                        </div>

                        <div class="alert alert-warning">
                                <strong>Listen!</strong> Don't learn fastly
                        </div>


                        <div class="alert alert-info">
                                <strong>God!</strong> Very difficult
                        </div>

                </div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" >

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">

			<div class="alert alert-danger alert-dismissible">
				<strong>Sorry!</strong> Data is deleted
				<span class="btn-close" data-bs-dismiss="alert"></span>
			</div>



		</div>

</body>
</html>
```

Bootstrap buttons

The .btn class is intended to be used in conjunction with our button variants,
or to serve as a basis for your own custom styles.
We need to use btn along with contextual classes
ex:

- btn-primary
- btn-info
- btn-danger
- and etc.

A .btn-outline-* ones to remove all background images and colors on any button.

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" >

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">

			<button class="btn btn-primary"> button1 </button>
			<button class="btn btn-secondary"> button2 </button>
			<button class="btn btn-info"> button3 </button>
			<button class="btn btn-danger"> button4 </button>
			<button class="btn btn-warning"> button5 </button>
			<button class="btn btn-success"> button6 </button>



		</div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" >

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">

			<button class="btn btn-outline-primary"> button1 </button>
			<button class="btn btn-outline-secondary"> button2 </button>
			<button class="btn btn-outline-info"> button3 </button>
			<button class="btn btn-outline-danger"> button4 </button>
			<button class="btn btn-outline-warning"> button5 </button>
			<button class="btn btn-outline-success"> button6 </button>
```

```
		</div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">

			<a href="" class="btn btn-outline-primary">clickME</a>



		</div>

</body>
</html>
```

ex:

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" >

	<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>
</head>
<body>
		<div class="container mt-5">
```

```
                    <button class="btn btn-link"> click </button>


                </div>

</body>
</html>
```

Bootstrap Design

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>

        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>

    <style type="text/css">
        body
        {
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
            background: linear-gradient(yellow,red);
        }
        .mystyle
        {
            box-shadow: 2px 2px 16px 6px #FFF;
        }
    </style>

    </head>
    <body>
                <div class="container">

                    <div class="mystyle">

                        <div class="row ">

                            <div class="col-3 border border-light">
```

```
                                        <img src="images/micky.png"
class="mt-5 w-75 h-75 d-block m-auto"/>
                                    </div>
                                    <div class="col-3 border border-light">
                                        <img src="images/micky.png"
class="mt-5 w-75 h-75 d-block m-auto"/>
                                    </div>
                                    <div class="col-3 border border-light">
                                        <img src="images/micky.png "
class="mt-5 w-75 h-75 d-block m-auto"/>
                                    </div>
                                    <div class="col-3 border border-light">
                                        <img src="images/micky.png"
class="mt-5 w-75 h-75 d-block m-auto"/>
                                    </div>

                                </div>


                            </div>


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

Bootstrap Glyphicons

Bootstrap includes 260 glyphicons from the Glyphicon Halflings set.

Glyphicons Halflings are normally not available for free, but their creator has made them available for Bootstrap free of cost.

We will see bootstrap glyphicons to display the icons in a web page.

```
<!DOCTYPE html>
<html>
    <head>
        <title>MyPage!</title>

        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>

    <!-- Bootstrap CDN LINK for  Icons -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
```

```
	</head>
	<body>

			<div class="container mt-5">

				<span class="bi bi-heart text-danger"></span>

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

ex:2

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>

	<!-- Bootstrap CDN LINK for  Icons -->
	<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

	</head>
	<body>

			<div class="container mt-5">

				<button class="bi bi-search btn btn-primary">Search</button>

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

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<meta charset="utf-8">
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>

        <!-- Bootstrap CDN LINK for  Icons -->
        <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">

        </head>
        <body>

                        <div class="container mt-5">

                                        <span class="bi bi-facebook text-primary"> </span>
                                        <span class="bi bi-twitter"> </span>
                                        <span class="bi bi-instagram"> </span>
                                        <span class="bi bi-youtube"> </span>
                                        <span class="bi bi-whatsapp text-success"> </span>

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

Bootstrap Cards

A card in Bootstrap 5 is a bordered box with some padding around its content. It includes options for headers, footers, content, colors, etc.

A basic card is created with the .card class, and content inside the card has a .card-body class:

The .card-header class adds a heading to the card and the .card-footer class adds a footer to the card.

Use .card-title to add card titles to any heading element.

The .card-text class is used to remove bottom margins for a <p> element if it is the last child (or the only one) inside .card-body.

Add .card-img-top to an <img> to place the image at the top inside the card.

Note that we have added the image outside of the .card-body to span the entire width:

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
                <div class="container mt-5">

                        <div class="card">
                                <div class="card-header">
                                        Heading
                                </div>
                                <div class="card-body">
                                        Body
                                </div>
                                <div class="card-footer">
                                        Footing
                                </div>
                        </div>

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

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
```

```
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
		<div class="container mt-5">

			<div class="card">
				<div class="card-header">
					Heading
				</div>
				<div class="card-body">
					<div class="card-title">Bootstrap</div>
					<div class="card-text">Bootstrap is a css framework</div>
				</div>
				<div class="card-footer">
					Footing
				</div>
			</div>

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

ex:3

```
<!DOCTYPE html>
<html>
	<head>
		<title>MyPage!</title>

		<meta charset="utf-8">

	<meta name="viewport" content="width=device-width, initial-scale=1">

	<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" ></script>

	<!-- Bootstrap CDN LINK for  Icons -->
	<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

	</head>
	<body>

			<div class="container mt-5">
```

```
                    <div class="row">

                        <div class="col-4">
                            <div class="card">
                            <img src="images/thumbnail1.jpg" class="card-
img img-fluid"/>
                            <div class="card-body">
                                <h2 class="card-title text-center">
Thumbnail-1</h2>
                                <p class="card-text text-center">
Illustration</p>
                            </div>
                            </div>
                        </div>

                        <div class="col-4">
                            <div class="card">
                            <img src="images/thumbnail2.jpg" class="card-
img img-fluid"/>
                            <div class="card-body">
                                <h2 class="card-title text-center">
Thumbnail-3</h2>
                                <p class="card-text text-center">
Illustration</p>
                            </div>
                            </div>
                        </div>

                        <div class="col-4">
                            <div class="card">
                            <img src="images/thumbnail3.jpg" class="card-
img img-fluid"/>
                            <div class="card-body">
                                <h2 class="card-title text-center">
Thumbnail-3</h2>
                                <p class="card-text text-center">
Illustration</p>
                            </div>
                            </div>
                        </div>

                    </div>

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

Bootstrap Collapse Plugin
==========
Collapsibles are useful when you want to hide and show large amount of content:

A data-bs-toggle="collapse" attribute is used to show and hide the content.

A data-bs-target attribute is used to connect a button with div tag.

A "collapse" class inside is used to hide/collapse the content for first time.

ex:1

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
		<div class="container mt-5">

			<button class="btn btn-primary" data-bs-toggle="collapse"
				data-bs-target="#myId"> Toggle </button>

			<h1 id="myId" class="collapse">This is heading tag</h1>

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

ex:2
```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>
```

```
        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link                                   href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

    <script                                  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

    <!-- Bootstrap Font Icon CSS -->
    <link          rel="stylesheet"          href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
            <div class="container mt-5">

                <a href="#myId" data-bs-toggle="collapse"> Toggle </a>

                <h1 id="myId" class="collapse">This is heading tag</h1>

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

Bootstrap collapse plugin with cards

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link                                   href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

    <script                                  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

    <!-- Bootstrap Font Icon CSS -->
    <link          rel="stylesheet"          href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
```

```
</head>
<body>
		<div class="container mt-5">

			<div class="card">

				<div class="card-header">
					<button class="btn btn-outline-primary"
							data-bs-toggle="collapse"
							data-bs-target="#myId">		Toggle
</button>
				</div>
				<div class="card-body collapse" id="myId">
					<div class="row">
						<div class="col-4">
							<img	src="images/rock.jpg"	class="w-
100 h-100"/>
						</div>
						<div class="col-4">
							<img	src="images/rock.jpg"	class="w-
100 h-100"/>
						</div>
						<div class="col-4">
							<img	src="images/rock.jpg"	class="w-
100 h-100"/>
						</div>
					</div>
				</div>

			</div>

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

Bootstrap responsive navbar

Navbars require a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl|-xxl} for responsive collapsing and color scheme classes.
Navbars and their contents are fluid by default.
Change the container to limit their horizontal width in different ways.
Use our spacing and flex utility classes for controlling spacing and alignment within navbars.
Navbars are responsive by default, but you can easily modify them to change that.
Responsive behavior depends on our Collapse JavaScript plugin.
.navbar-brand for your company, product, or project name.
.navbar-nav for a full-height and lightweight navigation (including support for dropdowns).
.navbar-toggler for use with our collapse plugin and other navigation toggling behaviors.
Flex and spacing utilities for any form controls and actions.
.navbar-text for adding vertically centered strings of text.
.collapse.navbar-collapse for grouping and hiding navbar contents by a parent breakpoint.

Add an optional .navbar-scroll to set a max-height and scroll expanded navbar content.

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

  <!-- Bootstrap Font Icon CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
		<nav class="navbar navbar-expand-lg navbar-light bg-light">
			<div class="container-fluid">
				<a class="navbar-brand" href="#">IHUBTALENT</a>

				<button class="navbar-toggler"
				    data-bs-toggle="collapse"
				    data-bs-target="#navbarId">

					<span class="navbar-toggler-icon"></span>
				</button>
				<div class="collapse navbar-collapse" id="navbarId">
					<ul class="navbar-nav ">
				<li class="nav-item">
					<a class="nav-link active" href="#">Home</a>
				</li>
				<li class="nav-item">
					<a class="nav-link active" href="#">About</a>
				</li>
				<li class="nav-item">
					<a class="nav-link active" href="#">Service</a>
				</li>
				<li class="nav-item">
					<a class="nav-link active" href="#">Portfolio</a>
				</li>
				<li class="nav-item">
					<a class="nav-link active" href="#">Contact</a>
```

```
                        </li>
                    </ul>
                        </div>
                    </div>
                </nav>
</body>
</html>
```

ex:2

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
                <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
                    <div class="container-fluid">
                        <a class="navbar-brand" href="#">IHUBTALENT</a>

                        <button class="navbar-toggler"
                            data-bs-toggle="collapse"
                            data-bs-target="#navbarId">

                            <span class="navbar-toggler-icon"></span>
                        </button>
                        <div class="collapse navbar-collapse" id="navbarId">
                            <ul class="navbar-nav ">
                        <li class="nav-item">
                            <a class="nav-link active" href="#">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" href="#">About</a>
                        </li>
                        <li class="nav-item">
```

```
                                <a class="nav-link active" href="#">Service</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link active" href="#">Portfolio</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link active" href="#">Contact</a>
                            </li>
                        </ul>
                            </div>
                        </div>
                </nav>
</body>
</html>
```

ex:3

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">

</head>
<body>
                <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
                        <div class="container-fluid">
                                <a class="navbar-brand" href="#">IHUBTALENT</a>

                                <button class="navbar-toggler"
                                    data-bs-toggle="collapse"
                                    data-bs-target="#navbarId">

                                        <span class="navbar-toggler-icon"></span>
                                </button>
                                <div class="collapse navbar-collapse" id="navbarId">
                                        <ul class="navbar-nav ">
```

```
                        <li class="nav-item">
                            <a class="nav-link active" href="#">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" href="#">About</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" href="#">Service</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" href="#">Portfolio</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" href="#">Contact</a>
                        </li>
                    </ul>
                        </div>
                    </div>
                </nav>

                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
                <h1>This is heading tag </h1>
</body>
</html>
```

Bootstrap Scrollspy

The scrollspy plugin automatically highlights the navigation links based on the scroll position to indicate where the user is currently on the page.

```
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

    <!-- Bootstrap Font Icon CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

    <style>
        .section1
        {
            width: 100%;
            height: 560px;
            background-color: #C4E538;
        }
        .section2
        {
            width: 100%;
            height: 560px;
            background-color: #ED4C67;
        }
        .section3
        {
            width: 100%;
            height: 560px;
            background-color: #9980FA;
        }
        .section4
        {
            width: 100%;
            height: 560px;
            background-color: #12CBC4;
        }
        .section5
        {
            width: 100%;
            height: 560px;
            background-color: #833471;
        }


    </style>

</head>
<body>
            <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
                <div class="container-fluid">
```

```
                    <a class="navbar-brand" href="#">IHUBTALENT</a>

                    <button class="navbar-toggler"
                        data-bs-toggle="collapse"
                        data-bs-target="#navbarId">

                            <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarId">
                            <ul class="navbar-nav ">
                    <li class="nav-item">
                            <a class="nav-link active" href="#homeId">Home</a>
                    </li>
                    <li class="nav-item">
                            <a class="nav-link active" href="#aboutId">About</a>
                    </li>
                    <li class="nav-item">
                            <a class="nav-link active"
href="#serviceId">Service</a>
                    </li>
                    <li class="nav-item">
                            <a class="nav-link active"
href="#portfolioId">Portfolio</a>
                    </li>
                    <li class="nav-item">
                            <a class="nav-link active"
href="#contactId">Contact</a>
                    </li>
                        </ul>
                            </div>
                        </div>
                </nav>

                <section class="section1"  id="homeId">
                        <h1 class="pt-5 text-center">
                            Home Page
                        </h1>
                </section>
                <section class="section2"  id="aboutId">
                        <h1 class="pt-5 text-center">
                            About Page
                        </h1>
                </section>
                <section class="section3"  id="serviceId">
                        <h1 class="pt-5 text-center">
                            Service Page
                        </h1>
                </section>
                <section class="section4"  id="portfolioId">
                        <h1 class="pt-5 text-center">
```

```
                    Portfolio Page
                </h1>
            </section>
            <section class="section5"  id="contactId">
                <h1 class="pt-5 text-center">
                    Contact Page
                </h1>
            </section>

</body>
</html>
```

Bootstrap5 forms

Bootstrap 5 supports two types of form layouts.

1) Stacked Forms
2) Inline forms

1) Stacked Forms

All textual <input> and <textarea> elements with class .form-control get proper form styling. we add a .form-label class to each label element to ensure correct padding.

Checkboxes have different markup. They are wrapped around a container element with .form-check, and labels have a class of .form-check-label, while checkboxes and radio buttons use .form-check-input.

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>

    <!-- add bootstrap 5 plugins -->
    <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

  <!-- Bootstrap Font Icon CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
            <div class="container mt-5">

                <!-- stack form -->
                <form action="">

                    <label class="my-3">UserName:</label>
```

```
                        <input type="text" name="t1" class="form-control" placeholder="username" autocomplete="off" />

                        <label class="my-3">Password:</label>
                        <input type="password" name="t1" class="form-control" placeholder="password" />

                        <input type="submit" value="submit" class="btn btn-primary my-5 w-100"/>

                    </form>



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

ex:2

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>

    <!-- add bootstrap 5 plugins -->
    <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

  <!-- Bootstrap Font Icon CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
            <div class="container mt-5">

                <div class="w-50">
                <!-- stack form -->
                <form action="">

                    <label class="my-3">UserName:</label>
                    <input type="text" name="t1" class="form-control" placeholder="username" autocomplete="off" />
```

```
                        <label class="my-3">Password:</label>
                        <input type="password" name="t1" class="form-control" placeholder="password" />

                        <input type="submit" value="submit" class="btn btn-primary my-5 w-100"/>

                    </form>
                    </div>


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

ex:3

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>

    <!-- add bootstrap 5 plugins -->
    <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

  <!-- Bootstrap Font Icon CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
            <div class="container mt-5">


                    <form action="">
                        <label>Address:</label>
                        <textarea name="t1" rows="5" cols="10" class="form-control"></textarea>
                    </form>

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

ex:4

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
		<div class="container mt-5">


			<form action="">

				<div class="form-check">
		<input type="radio" name="t1" value="male" class="form-check-input"/>
		<label class="form-check-label">MALE</label>
				</div>

				<div class="form-check">
		<input type="radio" name="t1" value="female" class="form-check-input"/>
		<label class="form-check-label">FEMALE</label>
				</div>

			</form>

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

ex:5

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
```

```
        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

    <script                                    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

    <!-- Bootstrap Font Icon CSS -->
    <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
                <div class="container mt-5">


                        <form action="">

                                <div class="form-check">
                <input   type="checkbox"   name="t1"   value="married"   class="form-check-
input"/>
                <label class="form-check-label">MARRIED</label>
                                </div>

                                <div class="form-check">
                <input   type="checkbox"   name="t1"   value="single"   class="form-check-
input"/>
                <label class="form-check-label">SINGLE</label>
                                </div>

                        </form>

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

ex:6
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
                <div class="container mt-5">


                        <form action="">

                                <label>Select Country:</label>
                                <select name="t1" class="form-select">
                                        <option value="">none</option>
                                        <option value="india">India</option>
                                        <option value="america">America</option>
                                        <option value="australia">Australia</option>
                                        <option value="canada">Canada</option>
                                </select>

                        </form>

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

ex:7
<!DOCTYPE html>
<html>
<head>
        <title>IHUB TALENT</title>

        <!-- add bootstrap 5 plugins -->
        <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link                                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script                                   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
```

```
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
		<div class="container mt-5">


			<form action="">

				<label>Performance Rating :</label>
				<input type="range" name="t1" class="form-range"/>

			</form>

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

ex:8

```
<!DOCTYPE html>
<html>
<head>
	<title>IHUB TALENT</title>

	<!-- add bootstrap 5 plugins -->
	<meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

   <!-- Bootstrap Font Icon CSS -->
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
		<div class="container mt-5">


			<form action="">

				<div class="input-group">
					<label class="input-group-text">www.</label>
					<input type="text" name="t1"/>
					<label class="input-group-text">.com</label>
```

```
                    </div>

                </form>

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

2. Inline Forms

If you want your form elements to appear side by side, use .row and .col.

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>

    <!-- add bootstrap 5 plugins -->
    <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

  <!-- Bootstrap Font Icon CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
            <div class="container mt-5">


                <form action="">

                <div class="row">

                    <div class="col-2">
                        <label>UserName:</label>
                    </div>
                    <div class="col-3">
                        <input type="text" name="t1"/>
                    </div>
                    <div class="col-2">
                        <label>Password:</label>
                    </div>
                    <div class="col-3">
                        <input type="password" name="t1"/>
```

```
                    </div>
                    <div class="col-2">
                        <input type="submit" value="submit"/>
                    </div>
                </div>

                </form>

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

Bootstrap Carousel
A .carousel class creates a carousel
A .slide class adds a CSS transition and animation effect when sliding from one item to the next. Remove this class if you do not want this effect.
A .carousel-inner class adds slides to the carousel.
A .carousel-item class specifies the content of each slide.
Add elements inside within each to create a caption for each slide.
A .carousel-indicators class adds indicators for the carousel.
These are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing).
A .carousel-control-prev class adds a left (previous) button to the carousel, which allows the user to go back between the slides.
A .carousel-control-next class adds a right (next) button to the carousel, which allows the user to go forward between the slides.
A .carousel-control-prev-icon class used together with .carousel-control-prev to create a "previous" button.
A .carousel-control-next-icon class used together with .carousel-control-next to create a "next" button.

```
<!DOCTYPE html>
<html>
<head>
    <title>IHUB TALENT</title>

    <!-- add bootstrap 5 plugins -->
    <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" >

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" ></script>

  <!-- Bootstrap Font Icon CSS -->
```

```
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
        <div class="container mt-5">


    <div id="carouselId" class="carousel slide" data-bs-ride="carousel">

    <div class="carousel-indicators">

        <button type="button" data-bs-target="#carouselId" data-bs-slide-to="0" class="active"></button>

        <button type="button" data-bs-target="#carouselId" data-bs-slide-to="1" ></button>

        <button type="button" data-bs-target="#carouselId" data-bs-slide-to="2" ></button>
    </div>
                <div class="carousel-inner">

                    <div class="carousel-item active">

                        <img src="images/thumbnail1.jpg" class="w-100 mx-auto d-block"/>

                        <div class="carousel-caption">
                            <h1>Washington DC</h1>
                            <p>Our Team Is Our Strength</p>
                        </div>

                    </div>

                    <div class="carousel-item">

                        <img src="images/thumbnail2.jpg" class="w-100 mx-auto d-block"/>

                        <div class="carousel-caption">
                            <h1>Los Angeles</h1>
                            <p>Our Work Is Our Identity</p>
                        </div>

                    </div>

                    <div class="carousel-item">
```

```
                                                    <img src="images/thumbnail1.jpg" class="w-100 mx-auto d-block"/>

                                                    <div class="carousel-caption">
                                                        <h1>New Jersey</h1>
                                                        <p>Our Service are Best Practice</p>
                                                    </div>


                                                </div>

                                    </div>


<button class="carousel-control-prev" type="button" data-bs-target="#carouselId" data-bs-slide="prev">
    <span class="carousel-control-prev-icon"></span>
    <span class="visually-hidden">Previous</span>
  </button>

  <button class="carousel-control-next" type="button" data-bs-target="#carouselId" data-bs-slide="next">
    <span class="carousel-control-next-icon"></span>
    <span class="visually-hidden">Next</span>
  </button>


                            </div>



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

React/ReactJS

It is a declarative, efficient and flexible javascript frontend library used to develop frontend applications and user interfaces i.e UI.
It is an open source, component based javascript frontend library responsible only for view layer of the application.
It is developed by Jordon Walke who was a software engineer at Facebook.
It is initially introduced by Facebook and later it is used in their own products like Instagram and facebook.
It was released to the public in the month of May, 2013.
The official website of reactjs is https://react.dev.
The latest version of reactjs is v18.2.2.
The main objective to introduced reactjs is to develop reusable components.
A component is a building block of react application.

Advantages of ReactJS

1. It is easy to learn and easy to use.
2. It is used to develop reusable components.
3. It supports one way data binding.
4. It supports virtual DOM.
5. Good documentation and Community support.
6. It supported by all major browsers.

Q) What is the difference between Angular and ReactJS ?

Angular	ReactJS
It is a product of Google.	It is a product of Facebook.
It was released in Oct,2015.	It was released in May,2013.
It is a javascript framework which is used to develop web and mobile applications.	It is a javascript open source library responsible only for view layer of the application.
It supports traditional DOM.	It supports virtual DOM.
It supports two way data binding.	It supports one way data binding.
It is used to develop rich featured Application.	It is used to develop Single Page Applications. (SPA)
Typescript language is used.	JSX language is used.
Jasmine and Karma is used as a testing framework.	Jest and Enzyme is used as a testing framework.
The default port number is 4200.	The default port number is 3000.
Angular used by Google, Mc'Donalds, Nike and etc.	React used by Facebook , Instagram, youtube airbnb ,skype and etc.

Pre-Requisition to Learn ReactJS

1) Good Knowledge on HTML,CSS and JavaScript.
2) Idea on JSX
3) Basic of ES6.
4) npm commands

How ReactJS works?

ReactJS internally uses virtual DOM.
Virtual DOM is also a Tree Node Structure.

Virtual DOM will find a effective way to make the changes in traditional/real DOM. Hence react applications will execute fastly.

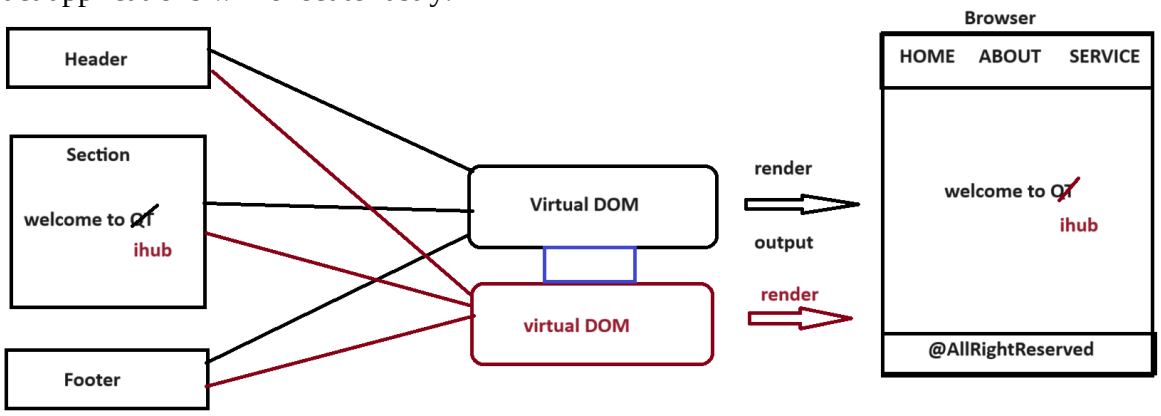

Diagram: react1.1

JSX

JSX stands for JavaScript XML.
JSX allows us to write HTML code in Javascript.
JSX element contains tags, attributes and childrens.
JSX is a not a neccessity to create react application instead we can use Babel.
JSX makes our program simpler and elegant.
JSX ultimately transpile to pure javascript which is understand by a browser window.

JSX elements

JSX elements allows us to write HTML code without using createElement() method or appendChild() method.

ex:1 JSX

```
<h1> Heading Tag </h1>
```

Babel

```
React.createElement('h1',null,'Heading Tag');
Here
'h1' is a tag name
'null' is a optional attribute
'Heading Tag' is a children
```

ex:2 JSX

```
<div> <h1>Heading Tag</h1> </div>
```

Babel

```
React.createElement('div',null,
        React.createElement('h1',null,'Heading Tag'));
```

ex:

JSX

```
<h1 id="myId"> Heading Tag </h1>
```

Babel

```
React.createElement('h1',{id:'myId'},'Heading Tag');
```

ex:

JSX

```
<h1 className="myClass"> Heading Tag </h1>
```

Babel

```
React.createElement('h1',{class:'myClass'},'Heading Tag');
```

ex: JSX

```
<h1 id="myId" className="myClass"> Heading Tag </h1>
```

Babel

```
React.createElement('h1',{id:'myId',class:'myClass'},'Heading Tag');
```

JSX Expression

JSX expression is used to represent expression in curly brace i.e. {}.
JSX expression can be a variables, constants and any valid javascript expressions.

```
ex:1    var a=10;
        <h1>{a}</h1>
ex:2    <h1>{10 + 20 }</h1>
ex:3    <h1>{Math.random()}</h1>
```

npm

npm stands for Node Package Manager.
It is a integrated tool for nodejs.
npm is used to install node dependencies/packages/libraries.
We can install node dependencies as follow.

```
ex:     npm install -g  dependency_name/library/package_name
```

All the dependencies will be installed in "node_modules" folder.

Steps to work with npm

```
step1:  Download and Install nodejs software.
        ex:     https://nodejs.org/en
step2:  Copy nodejs directory.
        ex:     C:\Program Files\nodejs
step3:  Paste nodejs directory in environmental variables.
        ex:     right click to my pc --> properties --> advanced system settings
                ---> environmental variables

                user variables --> click to new button -->

                variable name : path
                variable value :
                C:\Program Files\nodejs;C:\Users\Dell\AppData\Roaming\npm;
step4:  Check the environmental setup done perfectly or not.
        ex:     cmd> node  -v
                cmd> npm -v
step5:  Install npm by using below command.
        ex:     cmd> npm install -g npm
                        or
                cmd> npm install -g create-react-app
```

Steps to develop First React application

```
step1:  Make sure Nodejs setup is done perfectly.
step2:  Download and Install VSC (Visual Code Editor) editor.
        ex:     https://code.visualstudio.com/
step3:  Create a "ReactProjects" folder inside "E" drive.
step4:  Open the command prompt from "Reactprojects" folder.
step5:  Open VSC editor from "Reactprojects" folder.
        ex:     Reactprojects> code  .
```

step6: Create a react application/project i.e myapp1.
ex: Reactprojects>npx create-react-app myapp1
step7: Switch to the react project.
ex: Reactprojects> cd myapp1
step8: Run the react project by using below command.
ex: Reactprojects/myapp1> npm start
step9: Test the application by using below request url.
ex: http://localhost:3000
*Note: React application runs in a light weight development server with default 3000 port no.

Q) How to create react project /application ?
npx create-react-app myapp1

Q) How to move to the project in react?
cd myapp1

Q) How to run react application/ project?
npm start

<u>React Project Structure and work flow</u>

```
myapp1
|
|---node_modules
|
|---public
        |
        |---favicon.ico
        |---index.html
        |---manifest.json
|
|-----src
        |
        |---index.js
        |---index.css
        |
        |---App.js
        |---App.css
        |
        |---App.test.js
|
|------package.json
|------README.md
```

A "node_modules" contains all dependencies and libraries installed.
A "favicon.ico" is a favrouite icon of a react application.
A "index.html" is a main tempate of react application.
A "manifest.json " file contains metadata which is used when we install application on client mobile or computer.
A "index.js" file is a entry point.

A "index.css" file is related to index.js and it is global.
A "App.js" is a parent component.
A "App.css" file is related to App.js and it is global.
A "App.test.js" file is releted to unit testing.
A "package.json" file contains dependencies along with versions.

Note: index.html - main template
index.js - entry point
App.js - parent component
package.json - dependencies with versions

Work flow

code load to render to output
App.js ----------> index.js ----------> index.html ----------> Browser

Steps to develop second application in react

step1: create a react application i.e myapp2.
ex: ReactProjects> npx create-react-app myapp2
step2: Open the VSC editor.
ex: ReactProjects> code .
step3: Switch to myapp2 project.
ex: ReactProjects> cd myapp2
step4: Run the react application.
ex: ReactProjects/myapp2> npm start
step5: Test the application by using below request url.
ex: http://localhost:3000
step6: Write below code in App.js file.

Approach1

App.js

```
function App()
{
  return(
    <h1>I Love ReactJS </h1>
  )
}
export default App;
```

Approach2

App.js

```
var App=function(){

  return(
    <h1>I Love ReactJS Programming</h1>
  )
}
export default App;
```

Approach3

App.js

```
var App=()=>{

  return(
    <h1>I Love ReactJS Programming and Development</h1>
  )
}
export default App;
```

*React is mainly used to develop reusable components.

App.js

```
function App() {
  return (
    <h1>
      React Example for Reusability
    </h1>
  )
}
export default App
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
    <App />
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

React Fragment

Fragment is used to group of list of childrens without adding extra nodes of the DOM.

In general, We can return only one element at a time but we can't return more then one element directly.

To return more then one element we need to use React Fragment.

syntax <React.Fragment>

-

-

```
        </React.Fragment>
```
or
```
        <>
                -
                -
        </>
```

Examples

App.js

```
function App
{
   return (
      //return react element
      return  <h1>IHUB Talent</h1>
            <h2>React Tutorial For Freshers</h2>
   );
}
//export React component
export default App
```

o/p: Filed to compile

To overcome above problem we can use tag and inside that tag we can declare any child tags.

ex:

App.js

```
function App
{
   return (
      //return react element
      return
                <div>
                        <h1>IHUB Talent</h1>
                <h2>React Tutorial For Freshers</h2>
                </div>
   );
}
//export React component
export default App
```

Note: In above program " " tag is a unused tag.
To remove unused/unnecessary tags we can use React Fragment.

approach1

App.js

```
import React from "react";

function App()
```

```
{
   return (
         <React.Fragment>
         <h1>IHUB React Tutorial</h1>
         <h1>React Classes for Freshers</h1>
         </React.Fragment>
   );

}
export default App;
```

approach2

App.js

```
import React from "react";
import {Fragment} from 'react';
function App()
{
   return (
         <Fragment>
         <h1>IHUB React Tutorial</h1>
         <h1>React Classes for Freshers</h1>
         </Fragment>
   );

}
export default App;
```

approach3

App.js

```
import React from "react";

function App()
{
   return (
         <>
         <h1>IHUB React Tutorial</h1>
         <h1>React Classes for Freshers</h1>
         </>
   );

}
export default App;
```

React Components

A component is a building block of react application.

Components allows us to split our UI into independent reusable pieces.

ex: <Header> , <Footer>, <Section>, <Table>, <Form> and etc.

React components are like javascript functions because they accept arbitary inputs like props and return react element describing what should appear on the screen.

React component name always starts with uppercase letter.
There are two ways to declare react components.

1) Function Component / Functional Component
2) Class Component

1) Function Component

Function component is a javascript function which takes props as a argument along with inputs.
Functoin component is also known stateless component because it does not hold state.

syntax:1

```
function App()
{
	return
	(
		<h1> Named Function </h1>
	)
}
export default App;
```

syntax:2

```
var App=function()
{
	return
	(
		<h1> Anonymous Function </h1>
	)
}
export default App;
```

syntax:3

```
var App=()=>
{
	return
	(
		<h1> Arrow Function </h1>
	)
}
export default App;
```

Project structure

```
myapp3
|
|---node_modules
|
|-----public
	|
	|---manifest.json
	|---index.html
	|---favicon.ico
```

```
|
|------src
        |
        |---index.js
        |
        |---App.js

|------package.json
|------README.md
```

step1: create a react application i.e myapp3.
ex: Reactprojects> npx create-react-app myapp3
step2: Open VSC editor from Reactprojects folder.
ex: Reactprojects> code .
step3: Jump/Switch to myapp3 project.
ex: Reactprojects> cd myapp3
step4: Run the react application.
ex: Reactprojects/myapp3> npm start
step5: Test the react application by using below request url.
ex: http://localhost:3000
step6: Declare below code inside App.js file.

App.js

```
var App=()=>{
return (
        <h1>Arrow Function component</h1>
)
}
export default App
```

Function component with props

In order to use props in a component We need to perform following changes in react "myapp3" project.

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App name="Alan" rollno="101"/>
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
```

```
reportWebVitals();
```

App.js

```
var App=(props)=>{
  return (
    <>
    <h1>Name : {props.name}</h1>
    <h1>RollNo : {props.rollno}</h1>
    </>
  )
}
export default App
```

2) Class component

A class Component requires to extends from React Component.
The class must implements a render() method function which returns A react Element to be render.This is Similar to return value of a functional component.
In a class based component props are accessible via this.props.
The class component is also known as a stateful component because they can hold or manage local state.

Project structure

```
myapp4
|
|---node_modules
|
|-----public
          |
          |---manifest.json
          |---index.html
          |---favicon.ico
|
|------src
          |
          |---index.js
          |
          |---App.js

|------package.json
|------README.md
```

step1: create a react application i.e myapp4.
ex: Reactprojects> npx create-react-app myapp4
step2: Open VSC editor from Reactprojects folder.
ex: Reactprojects> code .
step3: Jump/Switch to myapp4 project.
ex: Reactprojects> cd myapp4
step4: Run the react application.
ex: Reactprojects/myapp4> npm start
step5: Test the react application by using below request url.

ex: http://localhost:3000

step6: Declare below code inside App.js file.

ex:

App.js

```
import {Component} from 'react';
class App extends Component
{
        render()
        {
        return(
        <h1>Class Component</h1>
        )
        }
}
export default App
```

Class component with props

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App name="Jose" rollno="501"/>
  </React.StrictMode>
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

App.js

```
import {Component} from 'react';
class App extends Component
{
   render()
   {
      return(
        <>
          <h1>Name :{this.props.name}</h1>
          <h1>RollNo :{this.props.rollno}</h1>
        </>
      )
   }
}
export default App
```

Composing Components in React

A component can refer to other components in their output is called composing component. Let us use some component abstraction for any level of details.

Project structure

```
myapp4
|
|-----node_modules
|
|-----public
|         |
          |---index.html (main template)
          |---favicon.ico (favicon)
          |---manifest.json (metadata)
|
|-----src
|         |
          |---index.js (entry point)
          |
          |
          |---App.js  (parent component)
          |
          |---Student.js (custom component)

|
|-----package.json
|-----README.md
```

step1: Create a React Application.
ex: ReactProjects>npx create-react-app myapp4
step2: Start Visual Studio Code (VSC) Editor.
ex: ReactProjects> code .
step3: Delete all the files from "src" folder.
step4: Create "index.js" file inside "src" folder.

index.js

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";

const root=ReactDOM.createRoot(document.getElementById('root'));

root.render(
   <React.StrictMode>
      <App/>
   </React.StrictMode>
)
```

step5: Create App.js file inside "src" folder.

App.js

```
import Student from './Student';
```

```
function App()
{
   return (
      <Student/>
   )
}
export default App;
```

step6: Create Student.js file inside "src" folder.

Student.js

```
function Student()
{
   return (
      <h1>Student Component</h1>
   )
}
export default Student;
```

step7: Move to myapp4.
ex: ReactProjects> cd myapp4

step8: Run the react application.
ex: ReactProjects/myapp4> npm start

step9: Check the output by using below url.
ex: http://localhost:3000

composing components using props

index.js

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";

const root=ReactDOM.createRoot(document.getElementById('root'));

root.render(
   <React.StrictMode>
      <App course="React"/>
   </React.StrictMode>
)
```

App.js

```
import Student from './Student';

function App(props)
{
   return (
     <Student crs={props.course}/>
   )
}
export default App;
```

Student.js

```
function Student(props)
{
   return (
      <h1>My Course Name : {props.crs}</h1>
   )
}
export default Student;
```

React CSS

CSS in React is used to style the React App or Component.

There are two ways available to add styling to your React App or Component with CSS.

1) Inline Styling
2) CSS Stylesheet

1)Inline CSS

Inline CSS represent by "style" attribute in React application.

The inline styles are specified with a JavaScript object in camelCase version of the style name.

ex:

App.js

```
import Student from "./Student";
function App()
{
 return <>
       <h1 style={{color:"green"}}>React Inline CSS</h1>
       <h1 style={{backgroundColor:"yellow"}}>React Inline CSS</h1>
        </>
}
export default App;
```

The inline styling also allows us to create an object with styling information and refer it in the style attribute.

App.js

```
import Student from "./Student";

function App()
{
 const mystyle = {
   color: "white",
   backgroundColor: "DodgerBlue",
   padding: "10px",
   fontFamily: "Arial"
 };
 return <>
       <h1 style={mystyle}>React Inline CSS</h1>
       <h1 style={{backgroundColor:"yellow"}}>React Inline CSS</h1>
        </>
}
export default App;
```

2) CSS Stylesheet

We can write styling in a separate file for your React application, and save the file with a .css extension.

Later we can import .css file in our required application.

ex:1

App.js

```
import Student from "./Student";
import './App.css';
function App()
{

  return <>
        <h1>React CSS styles</h1>
        <h1>React CSS styles</h1>
         </>
}
export default App;
```

App.css

```
body{
  background-color: yellow;
}
h1
{
  color:blue;
}
```

ex:2

App.js

```
import Student from "./Student";
import './App.css';
function App()
{

  return <>
        <h1 id="myId">React CSS styles</h1>
        <h1 className="myClass">React CSS styles</h1>
         </>
}
export default App;
```

App.css

```
body{
  background-color: yellow;
}
#myId
{
  color:blue;
}
.myClass
```

```
{
  color:red;
}
```

State

State is similar to props but it is a private and fully controlled by the component.
we can create a state only in class component but not in functional component.
It is possible to update the state or modify the state , where as props only for read only.
There are two ways to initialize the state in React component.

1)Directly inside class
2)Inside the Constructor

1. Directly inside class

```
class Student extends Component
{
        //define state
        state={
                name: "Anna Julie",
                prop1: this.props.prop1
        }
        render()
        {
                -
        }
}
```

Note: The "state" property is refered as state.
"this" is a class instance property

example

Project_structure

```
myapp6
|
|------node-modules
|
|------public
|       |
|       |------favicon.ico
|       |------index.html
|       |------manifest.json
|
|------src
        |
        |------index.js
        |------App.js
|
|------package.json
|------README.md
```

step1: Develop React Application.

ex: E:/ReactProjects>npx create-react-app myapp6

step2: Open VSC editor from Reactprojects.
ex: E:/Reactprojects>code .
step3: Install "ES7 React " Plugin/Extension from Visual Studio Code
for shortcuts to create React Applications.
ex:
imr +tab
imrc + tab
imrd + tab
imp + tab
rcc - class component
rcfe - named function component
rafce - anonymous function component
conlg+ tab
step4: Add below code inside "App.js" file.

App.js

```
import React, { Component } from 'react'

export default class App extends Component {
        state={
                name:"Alan"
        }
  render() {
        return (
          <h1>Hello {this.state.name}</h1>
        )
  }
}
```

step5: move to myapp5
ex: E:/BUI-2pm/ReactProjects> cd myapp6
step6: Run the application.
ex: DE:/BUI-2pm/ReactProjects/myapp6>npm start
step7: Test the React Application.
ex: http://localhost:3000

ex:2

App.js

```
import React, { Component } from 'react'
export default class App extends Component {
        state={
                name:"Alan",
                roll:this.props.rollno
        }
  render() {
        return (
          <div>
                <h1>Name: {this.state.name}</h1>
                <h1>RollNo: {this.state.roll}</h1>
           </div>
        )
```

```
  }
}
```

index.js

```
import App from './App';
import ReactDOM from 'react-dom/client';
import React from 'react';

const root=ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <React.StrictMode>
        <App rollno={501} />
    </React.StrictMode>
)
```

*Note: Here props property we are storing into a state.

2. Inside the Constructor

```
class App extends Component
{
        //constructor
        //props is optional
        constructor(props)
        {
                //it is required to call the parent class constructor
                super(props);

                //state
                this.state={
                        name:"alan",
                        prop1: this.props.prop1
                }
        }
        render()
        {
                -
        }
}
```

When the component class is created, The constructor is the first called so it is right place to add state.
The class instance has already been created in memory. So we can use "this" to set properties on it.
When we write a constructor ,make sure to call parent class constructor by using super(props) keyword.
When we call super with props ,React will make props avaiable accross/access the component through this.props.

Project structure

```
myapp7
|
```

```
|------node-modules
|
|------public
|          |
|          |------favicon.ico
|          |------index.html
|          |------manifest.json
|
|------src
          |
          |------index.js
          |------App.js
|
|------package.json
|------README.md
```

step1: Develop React Application.
ex: E:/ReactProjects>npx create-react-app myapp7
step2: Open VSC code editor.
ex: ReactProjects> code .
step3: Write below code inside "App.js" file in "src " folder (rcc).

Student.js

```
import React, { Component } from 'react'

export default class App extends Component {

          constructor()
          {
                    super();

                    this.state={
                              name: "Alan",
                              roll: 101
                    }
          }
  render() {
          return (
            <div>
                    <h1>Name: {this.state.name}</h1>
                    <h1>RollNo: {this.state.roll}</h1>
             </div>
          )
  }
}
```

step4: move to myapp7
ex: E:/BUI-2pm/ReactProjects> cd myapp7
step5: Run the application.
ex: DE:/BUI-2pm/ReactProjects/myapp7>npm start

step6: Test the React Application.
ex: http://localhost:3000

ex:2
App.js
```
import React, { Component } from 'react'
export default class App extends Component {

        constructor(props)
        {
                super(props);
                this.state={
                        name: "Alan",
                        roll: this.props.rollno
                }
        }
  render() {
        return (
          <div>
                <h1>Name: {this.state.name}</h1>
                <h1>RollNo: {this.state.roll}</h1>
           </div>
        )
  }
}
```

index.js
```
import App from './App';
import ReactDOM from 'react-dom/client';
import React from 'react';

const root=ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <React.StrictMode>
         <App rollno={501} />
    </React.StrictMode>
)
```

Event Handling in React
Event: Action to which a javascript can respond is called event.
ex: clicking on button
hovering of an element
and etc.
Handling events on react Elements are same like handling events on DOM elements.
ex:
Javascript
```
<button  onclick="f1()">clickMe</button
```
React
```
<button onClick={handleClick}>clickMe</button>  --> Function component
<button onClick={this.handleClick}>clickMe</button>  --> Class component
```

Event Handling using Function component

Project structure

```
myapp8
|
|----node_modules
|
|----public
          |
          |---index.html
          |---favicon.ico
          |---manifest.json

|
|-----src
|         |
          |---index.js
          |---index.css

          |
          |---App.js
          |---App.css
          |---App.test.js
|
|----package.json
|
```

step1: create a react project/application.
ex: ReactProjects>npx create-react-app myapp8

step2: Starts VSC code editor.
ex: ReactProjects> code .

step3: Move to the project.
ex: ReactProjects> cd myapp8

step4: Run the react application/project.
ex: ReactProjects/myapp8> npm start

ex:1

App.js

```
function App()
{

  function handleClick()
  {
    console.log("Button is clicked");
  }
 return (
    <button onClick={handleClick}>clickMe</button>
 )
}
export default App;
```

ex:2

App.js

```
function App()
{

   const handleClick=()=>
   {
      console.log("Button is clicked");
   }
  return (
      <button onClick={handleClick}>clickMe</button>
  )
}
export default App;
```

ex:3

```
import React from 'react'

function App() {

  function handleClick(e)
  {
     e.preventDefault();
     console.log("You have clicked");
  }

  return (
    <div>
       <a href="http://www.google.com" onClick={handleClick}> click </a>
    </div>
  )
}

export default App
```

Eventing Handling using class component

Project structure

```
myapp9
|
|----node_modules
|
|----public
          |
          |---index.html
          |---favicon.ico
          |---manifest.json

|
|-----src
|         |
```

```
        |---index.js
        |---index.css

        |
        |---App.js
        |---App.css
        |---App.test.js
|
|----package.json
|
```

step1: create a react project/application.
ex: ReactProjects> create-react-app myapp9
step2: Starts VSC code editor.
ex: ReactProjects> code .
step3: Move to the project.
ex: ReactProjects> cd myapp9
step4: Run the react application/project.
ex: ReactProjects/myapp9> npm start

ex:1

App.js

```
import {Component} from "react";
export default class App extends Component
{
   handleClick=()=>
   {
      console.log("Button is clicked",this);
   }

   render()
   {
      return(
        <button  onClick={this.handleClick}>clickMe</button>
      )
   }
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
```

```
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

update state

Using setState() method is used to update states.

ex:

```
this.state={
        name:"Alan"
}
this.setState({name:"Kelvin"});
```

ex:1

App.js

```
import {Component} from "react";
export default class App extends Component
{
    state={
       name : "Nancy",
       rollno: 101
    }

    handleClick=()=>
    {
      this.setState({name:"Lisa",rollno:501});
    }
  render()
  {
     return(
       <>
          <h1>Name : {this.state.name}</h1>
          <h1>RollNo : {this.state.rollno}</h1>
          <button  onClick={this.handleClick}>Change state</button>
       </>
     )
  }
}
```

Interview Questions

Q) Difference between function component vs class component?

function component	class component
It is also known as stateless component.	It is a statefull component.
In a function component we will use return keyword.	In a class component we will use render() method.
It supports hooks.	It does not support hooks.
Constructor is not used.	Constructor is used.

Q) Difference between real dom vs virtual dom ?

Real dom	virtual dom
It updates slow.	It updates faster.
Can directly updates HTML.	Can't directly updates HTML.
Creates a new dom if element updates.	Update the jsx if element updates.
DOM manipulation is very expensive.	DOM manipulation is very easy.
Too much of memory wastage.	No memory wastage.

Q) Difference between props and state ?

props	state
Props are read-only.	States are updatable.
Props are immutable.	State is mutable.
Props allow us to pass data from one component to other components as an argument.	State holds information about the components.
Props can be accessed by the child component.	State cannot be accessed by child components because it is private.
Stateless component can have Props.	Statefull components can have state.

Phases of components in ReactJS

There are four Phases of components in ReactJS.

1)Mounting
2)Updating
3)Error Handling
4)Unmounting

1. Mounting

Mounting is a process of creating an element and inserting it in a DOM tree.

2. Updating

Updating is a process of changing state or props of a component and update changes to nodes already existing in the DOM.

3. Error Handling

Error Handling used when there is a error during rendering, in lifecycle method or in the constructor of any child component.

4. Unmounting

Unmounting is a process of removing elements from the DOM tree.
In general it will clear the reserved memory.

Q) Explain life cycle methods of mounting ?

Mounting phase contains four methods.

1) constructor()
2) getDerivedStateFromProps()
3) render()
4) componentDidMount()

Q) Explain life cycle methods of unmounting?
Unmounting phase contains one method.
1) componentWillUnmount()

Q) Explain life cycle methods of updating?
updating phase contains five methods.
1) getDerivedStateFromProps()
2) shouldComponentUpdate()
3) render()
4) getSnapshotBeforeUpdate()
4) ComponentDidUpdate()

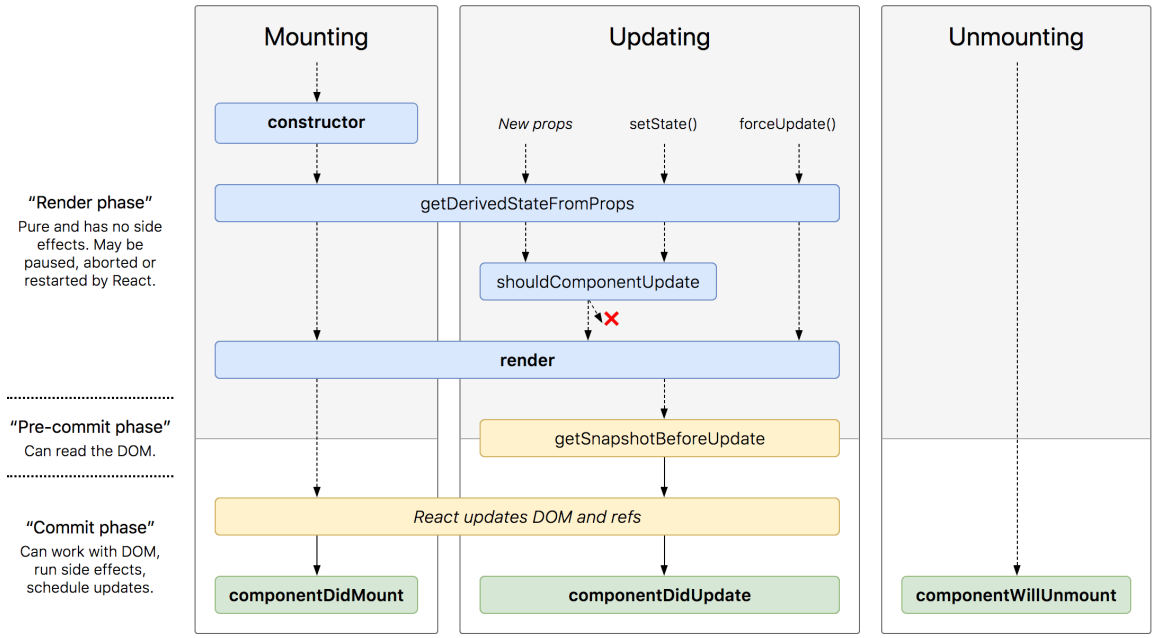

Diagram: react6.1

In react , all life cycle methods we can declare inside class component.

App.js

```
import React, { Component } from 'react'

export default class App extends Component {

  constructor()
  {
    console.log('constructor');
     super();

      this.state={
        name:"Alan"
      }
  }

  static getDerivedStateFromProps(props,state)
  {
    console.log('getDerivedStateFromProps')
```

```
  }

  render() {
    console.log('render');
    return (
      <>
        <h1>Name : {this.state.name}</h1>
      </>
    )
  }

  componentDidMount()
  {
    console.log('componentDidMount')
  }
}
```

Hooks

Hooks allow us to "hook" into React features such as state and lifecycle methods.
Hooks allow function components to have access to state , lifecycle methods and other React features.
Hooks allow us to use React without classes.It means you can use state and other React features without writing a class.
React provides a few built-In hooks like useState,useEffect and etc.
Hooks are new addition in React 16.8.

When use Hooks

If you write a function component and relize you need to add some state to it.

Rules of Hooks

There are 3 rules for hooks:

1)Hooks can only be called inside React function components.
2)Hooks can only be called at the top level of a component.
3)Hooks cannot be conditional

*Note: Hooks will not work in React class components.

Declaring State

A useState() is a Hook that allows us to add React state to function components.
We call it inside a function component to add some local state to it.
A useState() returns a pair - the current state value and a function that let us update it.
React will preserve this state between re-renders.
We can call this function from an event handler or somewhere else.

Project structure

```
myapp10
|
|---node_modules
|
|---public
        |
        |--favicon.ico
        |--index.html
```

```
        |--manifest.json
|
|------src
        |
        |---App.js
        |
        |---index.js

|-----package.json
|-----README.md
```

step1: create a react project.
ex: Reactprojects> npx create-react-app myapp10
step2: Open the VSC editor.
ex: Reactprojects> code .
step3: Move/Jump to myapp8 project.
ex: Reactprojects> cd myapp10
step4: Run the myapp8 project.
ex: Reactprojects/myapp10> npm start
step5: Test the application by using below request url.
ex: http://localhost:3000

App.js

```
import { useState } from "react";
function App()
{
   const [name,setName]=useState("Alan");

   const handleClick=()=>
   {
        setName("Kelvin");
   }
   return (
      <div>
         <h1>Name : {name}</h1>
         <button onClick={handleClick}>clickMe</button>
      </div>
   )
}
export default App;
```

index.js

```
import Student from './Student';
import ReactDOM from 'react-dom/client';
import React from 'react';
import App from './App';

const root=ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <React.StrictMode>
```

```
        <App />
    </React.StrictMode>
)
```

Effect Hooks

The Effect Hook let us to perform side effects in function components.

Data fetching, setting up a subscription, and manually changing the DOM in React components are all examples of side effects.

useEffect()

A useEffect is a hook for encapsulating code that has "side effects".if we are familiar with React class life cycle methods. We can thing of useEffect Hooks as componentDidMount, compnoentDidUpdate and componentWillUnmount combined.

useEffect =componentDidMount+ componentDidUpdate +componentWillUnmount

ex:

```
import React,{useEffect} from "react";
useEffect(Function)
```

or

```
useEffect(Function ,Array)
```

The function passes to useEffect will run after the render is committedto the screen.

Second argument to useEffect that is the array of values that the effect depends on.(It is for condition purpose).

*Note: We can call useEffect as many times we required.

ex:

```
useEffect(()=>
{
        console.log("Hello useeffect");
});
```

ex:

```
useEffect(()=>
{
        console.log("Hello useEffect");
},[count]);
```

What does useEffect do?

By using this Hook,we can tell react that your component needs to do something after render.

React remember the function we passed and call it later after performing the DOM updates.

In this effect, we set the document title,we could also perform data fetching or call some other imperative API.

*Note: useEffect runs after the first render and after every update.

<u>Project structure</u>

```
myapp11
|
|---node_modules
|
|---public
        |
```

```
        |--favicon.ico
        |--index.html
        |--manifest.json
|
|------src
        |
        |---App.js
        |
        |---index.js

|-----package.json
|-----README.md
```

step1: create a react project.
ex: Reactprojects> npx create-react-app myapp11
step2: Open the VSC editor.
ex: Reactprojects> code .
step3: Move/Jump to myapp9 project.
ex: Reactprojects> cd myapp11
step4: Run the myapp9 project.
ex: Reactprojects/myapp11> npm start
step5: Test the application by using below request url.
ex: http://localhost:3000

App.js

```
import { useState, useEffect } from "react";
function App()
{
   const [count,setCount]=useState(0);

   const handleClick=()=>
   {
        setCount(count+1);
   }

   useEffect(() => {
      // Update the document title using the browser API
      document.title = `you have click for ${count} times`;
    });
   return (
      <div>
         <h1>You clicked  {count} Times</h1>
         <button onClick={handleClick}>clickMe</button>
      </div>
   )
}
export default App;
```

index.js

```
import Student from './Student';
```

```
import ReactDOM from 'react-dom/client';
import React from 'react';
import App from './App';

const root=ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <React.StrictMode>
        <App />
    </React.StrictMode>
)
```

React useContext Hook (Context API)

Context provides a way to pass the data through the component tree without passing props down manually at several level.

To do this without Context, we will need to pass the state(useState) as "props" through each nested component. This is called "props drilling".

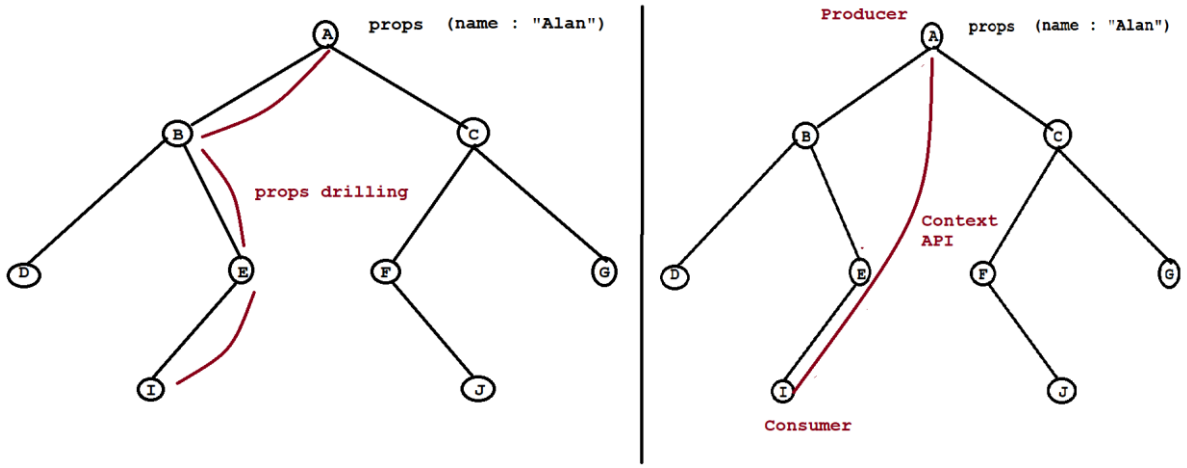

Diagram: react7.1

Project structure

```
myapp12
|
|------node-modules
|
|------public
|         |
|         |------favicon.ico
|         |------index.html
|         |------manifest.json
|
|------src
          |
          |------index.js
          |------App.js
          |------Acomponent.js
          |------Bcomponent.js
          |------Ccomponent.js
|
|------package.json
```

|------README.md

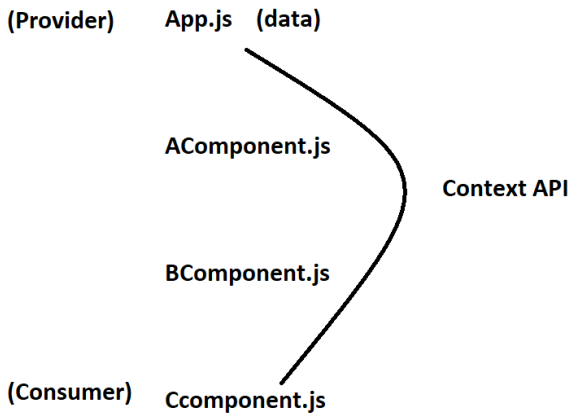

Diagram: react7.2

step1: create a react project.
ex: Reactprojects> npx create-react-app myapp12
step2: Open the VSC editor.
ex: Reactprojects> code .
step3: Move/Jump to myapp10 project.
ex: Reactprojects> cd myapp12
step4: Run the myapp10 project.
ex: Reactprojects/myapp12> npm start

App.js

```
import React from 'react';
import Acomponent from "./Acomponent";
export const UseContext=React.createContext();
function App()
{

   return (
      <div>
         <UseContext.Provider value={'IHUB'}>
         <Acomponent/>
         </UseContext.Provider>

      </div>
   )
}
export default App;
```

Acomponent.js

```
import Bcomponent from "./Bcomponent";
function Acomponent()
{
   return (
      <Bcomponent/>
   )
}
export default  Acomponent;
```

Bcomponent.js

```
import Ccomponent from "./Ccomponent";
function Bcomponent()
{
   return (
     <Ccomponent/>
   )
}
export default  Bcomponent;
```

Ccomponent.js

```
import {UseContext} from "./App";
function Ccomponent()
{
   return (
   <div>
    <UseContext.Consumer>
       {
          user => {
             return <div>The value is : {user} </div>
          }
       }
    </UseContext.Consumer>
   </div>
   )
}
export default  Ccomponent;
```

index.js

```
import Student from './Student';
import ReactDOM from 'react-dom/client';
import React from 'react';
import App from './App';

const root=ReactDOM.createRoot(document.getElementById('root'));
root.render(
     <React.StrictMode>
          <App />
     </React.StrictMode>
)
```

Custom Hooks

Hooks which are created by the user based on the application requirement are called custom hooks.

ex: myCustomHook()
customHook()
ihubHook()
myCustomCounter()

Project Structure

```
myapp11
|
|----node_modules
|
|----public
|          |
           |----favicon.ico
           |----index.html
           |----manifest.json
|
|-----src
|          |
           |----index.js
           |----App.js
           |----CustomHook.js
|
|-----package.json
|-----README.md
```

step1: Create a react project or application.
ex: Reactprojects> npx create-react-app myapp11
step2: Open VSC editor.
ex: Reactprojects> code .
step3: Move or Jump to myapp11 project.
ex: Reactprojects> cd myapp11
step4: Run the react application.
ex: Reactprojects/myapp11> npm start
step5: Test the react application.
ex: http://localhost:3000
step6: Create "CustomHook.js" file inside "src" folder.

ex:1

CustomHook.js

```
import React from 'react'
import {useState} from 'react'
function CustomHook()
{

  const [count,setCount]=useState(0);

  const handleClick=()=>
  {
    setCount(count+1);
  }

    return(
        {
        count,
        handleClick
```

```
      })
}
export default CustomHook
```

App.js

```
import React from 'react'
import customHook from './CustomHook';
function App() {
  const data=customHook();

  return (
    <div>
      <h1>Count : {data.count}</h1>
      <button onClick={data.handleClick}>Increment</button>
    </div>
  )
}
export default App
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Images/Assets in ReactJS

We can set images/Asset in ReactJS using two ways.

1. Inside public Folder.
2. Inside src folder.

1. Inside public folder

If we put a file into a public folder,It will not be processed by webpack. Instead it will be copied into the build folder untouched.

To reference assets in the public folder, we need to use a special variable called PUBLIC_URL. Only files inside the public folder will be accessible by %PUBLIC_URL% prefix.

How to use image

1)

myapp13

|

|---public

|

|---rock.jpg

index.html

```
<img src="%PUBLIC_URL%rock.jpg" alt="mypic"/>
```

2)

myapp

|

|---public

|

|---images

|

|--rock.jpg

index.html

```
<img src="%PUBLIC_URL%/images/rock.jpg" alt="mypic"/>
```

If we want to use Image in Javascript file.

App.js

```
<img src={process.env.PUBLIC_URL +"/rock.jpg" } />
<img src={process.env.PUBLIC_URL +"/images/rock.jpg" } />
```

ex:1

index.html

```
-
-
-
 <div id="root"></div>
<img src="%PUBLIC_URL%/team1.jpeg" alt="mypic"/>
-
-
-
```

*Note: Mostly of the time we are displaying images in Component only.

ex:

App.js

```
import React, { Component } from 'react'

export default class App extends Component {
      render() {
            return (
                  <div>
<img src={process.env.PUBLIC_URL+"team1.jpeg"} alt="mypic"></img>
                  </div>
            )
      }
```

```
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from "./App";

//render the component to index.html
ReactDOM.render(<App />,document.getElementById("root"));
```

2) Inside src folder

we can import a file right in a Javascript module.This tell webpack to include that file in the bundle.

How to use

1)

```
myapp
|
|---src
        |
        |---rock.jpg
```

App.js

```
import pic from "./rock.jpg";
<img src={pic} alt="mypic" />
```

This ensures that when the project is built.Webpack wil correctly movethe images into the build folder and provide us with correct paths.

ex:

App.js

```
import React, { Component } from 'react'
import pic from "./team1.jpeg";

export default class App extends Component {
    render() {
        return (
            <div>
                <img src={pic} alt="mypic"></img>
            </div>
        )
    }
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from "./App";

//render the component to index.html
ReactDOM.render(<App />,document.getElementById("root"));
```

React Router

Routing is a process in which a user is directed to different pages based on their actions or requests.

ReactJS Router is mainly used for developing Single Page Web Applications.

React Router is used to define multiple routes in the application.

When a user types a specific URL into the browser, and if this URL path matches any 'route' inside the router file, the user will be redirected to that particular route.

React Router is a standard library system built on top of the React and used to create routing in the React application using React Router Package.

React contains three different packages for routing.

1) react-router:

It provides the core routing components and functions for the React Router applications.

2) react-router-native:

It is used for mobile applications.

3) react-router-dom:

It is used for web applications design.

*Note:

It is not possible to install react-router directly in your application.

To use react routing, first, you need to install react-router-dom modules in your application.

We have two types of router components.

1. <BrowserRouter>:
   It is used for handling the dynamic URL.
2. <HashRouter>:
   It is used for handling the static request.

Project structure

```
myapp12
|
|------node-modules
|
|------public
|       |
|       |------favicon.ico
|       |------index.html
|       |------manifest.json
|
|------src
        |
        |------index.js
        |------App.js  (Routing File)
        |------Home.js
        |------About.js
        |------Contact.js
        |------Error.js
|
|------package.json
|------README.md
```

step1: create react "myapp12" project in VSC.
ex: projects>npx create-react-app myapp12
step2: Move to myapp12 project.
ex: project> cd myapp12
step3: install react router dom.
ex: project/myapp12>npm install --save react-router-dom
step4: Restart the application .
ex: myapp14> npm start
step5: create App.js,Home.js,About.js ,Contact.js and Error.js component inside "src" folder.

App.js

```
import Home from './Home';
import Contact from './Contact';
import About from './About';
import Error from './Error'
import { BrowserRouter, Routes, Route } from "react-router-dom";

function App() {
  return (
    <div>
      <BrowserRouter>
     <Routes>
       <Route exact path="/" element={<Home />}/>
       <Route path="/about" element={<About />}/>
       <Route path="/contact" element={<Contact />}/>
       <Route path="*" element={<Error />}/>
     </Routes>
     </BrowserRouter>
    </div>
  );
}
export default App;
```

Home.js

```
function Home()
{
   return (
      <div>
        <h1>Home</h1>
      </div>
   )
}
export default Home;
```

About.js

```
function About()
{
   return (
      <div>
        <h1>About</h1>
```

```
      </div>
   )
}
export default About;
```

Contact.js

```
function Contact()
{
   return (
      <div>
        <h1>Contact</h1>
      </div>
   )
}
export default Contact;
```

Error.js

```
function Error()
{
   return(
      <div>
         <h1>OOPS! 404 Error </h1>
      </div>
   )
}
export default Error;
```

step6: create index.js component to render the output inside "src" folder.

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>

    <App/>
  </React.StrictMode>
);
```

step7: Test the application by using below url's.

ex: http://localhost:3000/
http://localhost:3000/home
http://localhost:3000/about
http://localhost:3000/contact
http://localhost:3000/gallery
http://localhost:3000/services

Adding Navigation using Link component

A Link component is used to create links which allow to navigate on different URLs and render its content without reloading the webpage.

ex:2

App.js

```
import Home from './Home';
import Contact from './Contact';
import About from './About';
import Error from './Error'

import {Link, Routes,Route,BrowserRouter } from 'react-router-dom'
function App() {
  return (
    <div>
      <BrowserRouter>

      <nav >
        <Link style={{display:"block"}} to="/">Home</Link>
        <Link style={{display:"block"}} to="/about">About Us</Link>
        <Link style={{display:"block"}} to="/contact">Contact US</Link>
      </nav>
      <Routes>
        <Route exact path="/" element={<Home />}/>
        <Route path="/about" element={<About />}/>
        <Route path="/contact" element={<Contact />}/>
        <Route path="*" element={<Error />}/>
      </Routes>
      </BrowserRouter>
    </div>
  );
}
export default App;
```

Home.js

```
function Home()
{
    return (
        <div>
          <h1>Home</h1>
        </div>
    )
}
export default Home;
```

About.js

```
function About()
{
    return (
        <div>
```

```
        <h1>About</h1>
      </div>
   )
}
export default About;
```

Contact.js

```
function Contact()
{
   return (
      <div>
        <h1>Contact</h1>
      </div>
   )
}
export default Contact;
```

Error.js

```
function Error()
{
   return(
      <div>
         <h1>OOPS! 404 Error </h1>
      </div>
   )
}
export default Error;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>

    <App/>

  </React.StrictMode>
);
```

Bootstrap in React

A Single-page applications gaining popularity over the last few years, so many front-end frameworks have introduced such as Angular, Vue, Ember, etc. As a result, jQuery is not a necessary requirement for building web apps.

Currently, React is mostly used JavaScript library for building web applications, and Bootstrap become the most popular CSS framework.

Let see how to use bootstrap in react applications.

Project structure

```
myapp14
|
|----node_modules
|
|----public
          |
          |---favicon.ico
          |---index.html
          |---manifest.json
|
|------src
|         |
          |---index.js
          |
|         |---App.js
|
|------package.json
|
|------README.md
```

step1: create a react project i.e myapp14.
ex: Reactprojects> npx create-react-app myapp14
step2: Open the VSC code editor.
ex: Reactprojects> code .
step3: Move/Switch to myapp14 project.
ex: Reactprojects> cd myapp14
step4:
Install Bootstrap package.
ex: Reactprojects/myapp13> npm install bootstrap
step5: Run the react application.
ex: Reactprojecs/myapp14> npm start
step6: Create a App.js file inside "src" folder.

App.js

```
function App()
{
  return(
     <div className="container mt-5">
        <button className="btn btn-outline-primary">clickMe</button>
     </div>
  )
}
export default App;
```

step7: Import bootstrap package inside "index.js" file.

index.js

```
import React from 'react';
```

```
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import '../node_modules/bootstrap/dist/css/bootstrap.css';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

reportWebVitals();
```

step8: Test the application by using below request url.
ex: http://localhost:3000

React Forms

Forms are an integral part of any modern web application.
It allows the users to interact with the application as well as gather information from the users.
Forms can perform many tasks that depend on the nature of your business requirements and logic such as authentication of the user, adding user, searching, filtering, booking, ordering, etc.
A form can contain text fields, buttons, checkbox, radio button, etc.

Creating Form

React offers a stateful, reactive approach to build a form.
The component rather than the DOM usually handles the React form.
In React, the form is usually implemented by using controlled components.

Controlled component

In the controlled component, the input form element is handled by the component rather than the DOM. Here, the mutable state is kept in the state property and will be updated only with setState() method.

Controlled components have functions that govern the data passing into them on every onChange event, rather than grabbing the data only once, e.g., when you click a submit button.This data is then
saved to state and updated with setState() method. This makes component have better control over the form elements and data.

Project structure

```
myapp15
|
|----node_modules
|
|----public
        |
        |---favicon.ico
```

```
        |---index.html
        |---manifest.json

|------src
        |
        |---index.js
        |
        |---App.js

|
|-------package.json
|
|-------README.md
```

step1: create a react project i.e myapp15.
ex: Reactprojects> npx create-react-app myapp15
step2: Open the VSC code Editor.
ex: Reactprojects> code .
step3: Switch/Move to myapp15 project.
ex: Reactprojects> cd myapp15
step4: Install bootstrap package.
ex: Reactprojects/myapp14> npm install bootstrap
step5: Run the react application.
ex: Reactprojects/myapp14> npm start
step6: Import Bootstrap package inside "index.js" file.

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import '../node_modules/bootstrap/dist/css/bootstrap.css';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

step7: Create App.js file inside "src" folder.

App.js

```
import {useState} from 'react';
```

```
function App()
{
  const [userRegistration,setUserRegistration]=useState({
    username:"",
    password:"",
    date:"",
    category:""
})

const handleClick=(e)=>
{
  const name=e.target.name;
  const value=e.target.value;
  //set to state
  setUserRegistration({... userRegistration,[name]:value});
}

const handleSubmit=(e)=>
{
    e.preventDefault();
    setUserRegistration({username:"",password:"",date:"",category:""});
}


  return(

    <div className="container mt-4">

       <form onSubmit={handleSubmit}>
       <div className="row w-50">
       <h1 className="text-center" ><u>React Form </u></h1>
        <label htmlFor="username" className="my-3">UserName:</label>
        <input type="text" name="username" autocomplete="off"
             className="form-control"
             value={userRegistration.username}
             onChange={handleClick}/>

        <label htmlFor="password" className="my-3">Password:</label>
        <input type="password" name="password" autocomplete="off"
              className="form-control"
              value={userRegistration.password}
              onChange={handleClick}/>

        <label htmlFor="date" className="my-3">Date:</label>
        <input type="date" name="date" autocomplete="off"
              className="form-control"
              value={userRegistration.date}
              onChange={handleClick}/>

        <label htmlFor="category" className="my-3">Category</label>
```

```
      <select name="category" className="form-control"
            value={userRegistration.category}
            onChange={handleClick}>
          <option value="">none</option>
          <option value="entertainment">Entertainment</option>
          <option value="drama">Drama</option>
          <option value="action">Action</option>
      </select>

      <button className="btn btn-primary mt-4 w-100"> submit </button>
      </div>

      </form>
   </div>

  )

}
export default App;
```

step8: Test the application by using below request url.
ex: http://localhost:3000

Lists in ReactJs

Lists are used to display data in an ordered format and mainly used to display menus on websites. In React, Lists can be created in a similar way as we create lists in JavaScript. Let us see how we transform Lists in regular JavaScript.
The map() function is used for traversing the lists.
ex:
Project structure

```
myapp16
|
|----node_modules
|
|-----public
|         |
          |---favicon.ico
          |---index.html
          |---manifest.json
|
|-----src
          |
          |---index.js
          |---App.js
|
|-----package.json
|-----README.md
```

step1: create a react project i.e myapp16.
ex: Reactprojects> npx create-react-app myapp16

step2: Open the VSC code editor.
ex:
Reactprojects> code .
step3: Move/Switch to myapp16 project.
ex: Reactprojects> cd myapp16
step4: Run the react application.
ex; Reactprojects/myapp16> npm start
step5: Create App.js file inside "src" folder.

App.js

```
import React, { Component } from 'react'
export default class App extends Component {
  render() {

       var arr=[10,20,30,40];

       var newArr=arr.map((element)=>
       {
         return <li>{element}</li>
       })

    return (
      <ul>
         {newArr}
      </ul>
    )
  }
}
```

step6: Test the application by using below request url.
ex: http://localhost:3000

ex:2

App.js

```
import React, { Component } from 'react'
export default class App extends Component {
  state={
    users:[
      {pid:101,pname:"LG",pprice:10000},
      {pid:102,pname:"LAVA",pprice:20000},
      {pid:103,pname:"MI",pprice:30000},
      {pid:104,pname:"SAMSUNG",pprice:40000}
    ]
  }

  render() {

        var newArr=this.state.users.map(user=>
        {
          return <h1>Id: {user.pid} Name: {user.pname} Price: {user.pprice}</h1>
```

```
        })

    return (
       <div>
         {newArr}
       </div>
    )
  }
}
```

ex:3

App.js

```
import React, { Component } from 'react'
export default class App extends Component {
  state={
    users:[
      {pid:101,pname:"LG",pprice:10000},
      {pid:102,pname:"LAVA",pprice:20000},
      {pid:103,pname:"MI",pprice:30000},
      {pid:104,pname:"SAMSUNG",pprice:40000}
    ]
  }

  render() {

        var newArr=this.state.users.map(user=>
        {
          return <tr><td>{user.pid}</td> <td> {user.pname}</td> <td>{user.pprice}</td></tr>
        })

    return (
       <div>
         <table border={1} width="100%">
           <thead>
              <tr>
                <th>ID</th>
                <th>NAME</th>
                <th>PRICE</th>
              </tr>
           </thead>
           <tbody>
              {newArr}
           </tbody>
         </table>
       </div>
    )
  }
}
```

Key in ReactJS

A key is a special string attribute you need to include when creating lists of elements.
Keys help react identify which items have changed are added or are removed.
ex:

App.js

```
import React, { Component } from 'react'
export default class App extends Component {
  state={
    users:[
      {pid:101,pname:"LG",pprice:10000},
      {pid:102,pname:"LAVA",pprice:20000},
      {pid:103,pname:"MI",pprice:30000},
      {pid:104,pname:"SAMSUNG",pprice:40000}
    ]
  }
  render() {
        var newArr=this.state.users.map(user=>
        {
         return <tr key={user.pid}><td>{user.pid}</td> <td> {user.pname}</td>
<td>{user.pprice}</td></tr>
        })

    return (

        <table border={1} width="100%">
          <thead>
            <tr>
              <th>ID</th>
              <th>NAME</th>
              <th>PRICE</th>
            </tr>
          </thead>
          <tbody>
            {newArr}
          </tbody>
        </table>

    )
  }
}
```

Axios

Axios is used to make HTTP request (GET,POST,PUT,DELETE).
Using axios we can give the request to Rest API's.
We can install axios by using below command.

```
ex:     reactprojects> npm  install axios
                       or
        reactprojects> yarn add axios
```

Project structure

```
myapp17
|
|----node_modules
|
|-----public
|          |
           |---favicon.ico
           |---index.html
           |---manifest.json
|
|-----src
           |
           |---index.js
           |---App.js
           |---FetchApi.js
|
|-----package.json
|-----README.md
```

step1: create a react project i.e myapp17.
ex: Reactprojects> npx create-react-app myapp17
step2: Open the VSC code editor.
ex: Reactprojects> code .
step3: Move/Switch to myapp17 project.
ex: Reactprojects> cd myapp17
step4: Install axios in myapp17 project.
ex: Reactprojects/myapp17> npm install axios
step5: Run the react application.
ex: Reactprojects/myapp17> npm start
step6: Create App.js file inside "src" folder.

App.js

```
import FetchApi from "./FetchApi";
function App()
{
  return (
     <FetchApi/>
  )
}
export default App;
```

step7: Arange one REST API for fetching the data.
ex: https://jsonplaceholder.typicode.com/users
step8: Create FetchApi.js file inside "src" folder.

FetchApi.js

```
import {useState} from 'react';
import axios from 'axios';
function FetchApi()
```

```
{
   const [data,setData]=useState([])
  const handleClick=()=>
  {
    axios.get("https://jsonplaceholder.typicode.com/users")
    .then(response=>
      {
        setData(response.data)
      })
      .catch(error=>
        {
          this.setData(error);
        })
  }
  return (
    <div>
      <center>
        <button onClick={handleClick}>Fetch API </button>
      </center>
      <table border={1} width="100%">
        <thead>
          <tr>
              <th>ID</th>
              <th>NAME</th>
              <th>USERNAME</th>
              <th>EMAIL</th>
          </tr>
        </thead>
        <tbody>
           {
            data.map(data=>
              {
                return <tr>
                        <td>{data.id}</td>
                        <td>{data.name}</td>
                        <td>{data.username}</td>
                        <td>{data.email}</td>
                       </tr>
              })
           }
        </tbody>
      </table>
    </div>
  )
}
export default FetchApi;
```

step9: Test the application by using below request url.

ex: http://localhost:3000