

CSS → Cascading Style Sheets

31/22

CSS → Design (make up)

3 ways to apply CSS to HTML or Types of CSS.

* CSS :- Stands for Cascading Style Sheets.

→ CSS used to design HTML

→ Current Version of CSS is CSS3

→ Bootstrap is the framework of CSS, which is used to develop mobile friendly and responsive apps.

* Types of CSS :-

1) IN-LINE CSS

2) Internal CSS

3) External CSS.

* In-line-CSS :- [This is only applicable for only one element or one tag]

Syntax :- <P style="CSS:Value;"> </P>

for Ex:- <P style="color:red;"> Jspider1 </P>

O/P:- Jspider1 → red colour of big C

<!DOCTYPE html>

<html>

<head>

 css

 <title>HTML forms</title>

</head>

<body>

 <P style="color:red;"> Jspider1 </P>

 <P> Jspiders1 </P>

 <P style="color:red;"> Jspiders1 </P>

</body>

</html>

O/P:- Jspider1 (red)

Jspider1

Jspiders1 (red)

Dry code :- Could not repeat the same code

Internal CSS :- style tag is used $\langle \text{style} \rangle$ $\langle \text{style} \rangle$
↳ Selecting all the tags of same document
It is limited only to one document, cannot be used outside the document.

for Ex :- $\langle \text{!DOCTYPE Html} \rangle$

```
 $\langle \text{html} \rangle$ 
   $\langle \text{head} \rangle$ 
     $\langle \text{title} \rangle$  HTML FORMS  $\langle \text{/title} \rangle$ 
     $\langle \text{style} \rangle$ 
      P { color: red; }
     $\langle \text{/style} \rangle$ 
   $\langle \text{/head} \rangle$ 
   $\langle \text{body} \rangle$ 
     $\langle \text{P} \rangle$  Jspiders1  $\langle \text{/P} \rangle$ 
     $\langle \text{P} \rangle$  Jspiders1  $\langle \text{/P} \rangle$ 
     $\langle \text{P} \rangle$  Jspiders1  $\langle \text{/P} \rangle$ 
     $\langle \text{h1} \rangle$  Qspiders1  $\langle \text{/h1} \rangle$ 
   $\langle \text{/body} \rangle$ 
 $\langle \text{/html} \rangle$ 
```

Output :-
Jspiders1 (red)
Jspiders1 (red)
Jspiders1 (red)
Qspiders1

External CSS :- $\cdot \text{CSS}$ extension used
mandatory

Separate .css file is linked to HTML form.

$\langle \text{!DOCTYPE} \rangle$

$\langle \text{html} \rangle$

$\langle \text{head} \rangle$

$\langle \text{title} \rangle$ Jspiders $\langle \text{/title} \rangle$

$\langle \text{link rel="stylesheet" href=". /style.css" } \rangle$

$\langle \text{/head} \rangle$

```

<body>
  <P> Jspiders1 </P>
  <P> Jspiders1 </P>
  <P> Jspiders1 </P>
  <h1> Jspiders1 </h1>
</body>
</html>

```

Style.css

```

P{color:red;}

```

Output:

Jspiders1 (red)
Jspiders1 (red)
Jspiders1 (red)
Jspiders1

Program 1. HTML

- It is used in Industrial standards.
- Even though 'i' tag is present in the code, it is not displayed because it is overruled by 'b' tag.
- If we use external CSS, even though 'i' tag is present, it will be displayed.
- If we use inline CSS, then first preference given to 'b' tag.
- Text-decoration is used to remove underline over anchor tag.
- text-transform (text-transform) → used to change cases like upper to lower or lower to upper case.
- text-indent (text-indent) → used to change text shadow → accept - 4 values

Program:- Font properties

```

<!DOCTYPE HTML>
<html>
  <head>
    <title> CSS Properties </title>
  </head>
  <body>
    <p style="color: green;
      font-size: 100px;
      font-weight: bold;
      font-family: Blackadder ITC;
      font-style: italic;
      text-align: center;
      text-shadow: 5px 4px 8px red;"> Jspiders1 </p>

    <p style="text-align: justify;
      text-indent: 80px;">
      Jspider Java Training center
    </p>

```

```
<h1 style="text-decoration: underline;"> Jspiders </h1>
<h1 style="text-decoration: Overline;"> Jspiders </h1>
<h1 style="text-decoration: line-through;"> Jspiders </h1>
<a href="#" style="text-decoration: none;"> Jspiders </a>
Java training center </h1>
</body>
</html>
```

Overflow property is used to hide Content scroll value →
Scroll the hidden content. → Clip the Content or add the
scroll value when the element content is too big.

```
<!DOCTYPE HTML>
<HTML> <Head> </head>
<p style="background-color: green; width: 200px; height: 200px; overflow: auto; color: white;">
<ip>
</body>
</html>
```

Squash resolution using to change width height

```
<body>
<ul>
<li style="list-style-image: url('laser1.webp'); position: absolute; left: 0; top: 0; width: 100%; height: 100%; background-color: black; opacity: 0.5; z-index: -1;">
<li style="list-style-type: none; position: relative; z-index: 1; padding: 10px; border: 1px solid black; border-radius: 10px; background-color: white; width: fit-content; margin: auto;">
```

```
<li> 3spiders </li>
<li> 3spiders </li>
<li style="list-style-image: url(/race1.webp);
    list-style-position: inside;"> 3spiders </li>
```

```
<li> 3spider1 </li>
```

```
</ul>
```

```
<body>
```

```
</html>
```

→ save file as index.html → open file → open file → there is a file named index.html

style.css

→ will add some code in style.css file

→ click on file → open file → open file

→ click on file → open file → open file

→ create golden JEW -IS & save index.html.

& document + css + save.

→ How to apply border → inspect element

Right click on element (3spiders) → inspect element

Style → click on height & width, set (41px, 118px)

after put value in program

for design → solid

border is property used to set border with 3 values

solid, color, length.

border: 5px solid black;

border: 5px dashed black;

border: 5px groove black;

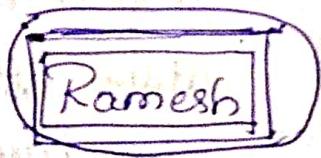
border: 5px dotted black;

border: 5px solid black;

for rounded border border-radius: 10px;

border-radius: 10px, 10px, 10px, 10px;

border-radius: 10px 10px 10px 10px;



Padding property → space b/w content & the border.

→ Padding: 10px 10px 0px 0px
left right Right left

top bottom
padding-top: 10px; } top & bottom not use for
padding-bottom: 0px; } border & radius.

outline property → after the border one line will come.

outline: 5px solid hotpink;

no outline top, bottom, border, radius properties.

→ Outline offset: 1px;

The space b/w & outline is Margin.

The space b/w & elements is Margin.

<body>, <h1> etc. are children of <html>.

<h1> is child of <h1>

<h1 style="margin: 50px"></h1>

<h1 style="margin-top: 50px; margin-bottom: 40px">

<h1 style="margin-top: 50px; margin-bottom: 40px">

HTML elements inherit margin-top & margin-bottom from its parent.

these all properties belongs to box model concept.

these all properties belongs to box model concept.

(google the box model concept) → proper design, layout

Interview question? what is box model CSS concept which is used to provide proper design

& layout of 4 properties → border, padding, outline offset, margin.

Outline offset → adds space b/w the outline & edge or border of an element.

* css box Model :-

In CSS, the term box model is used when talking about design & layout.

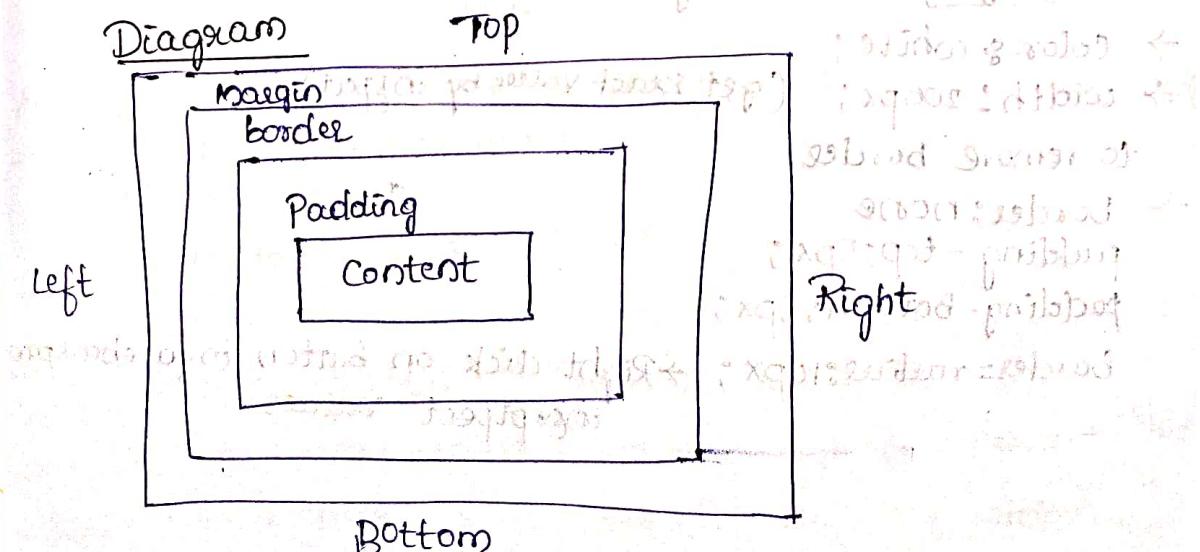
The CSS box model is essentially a box that wraps around every HTML element.

It consists of 4 parts - margin, border, padding and content.

b1

```
{ background-color: green;
height: 41px;
width: 118px;
color: white;
/* border: 2px solid red;
border: 5px groove orange;
border-top: 5px dotted orange;
border-left: 5px groove red;
border-radius: 10px 20px 10px 20px;
padding: 20px;
padding: 10px 5px 20px 20px;
padding-top: 20px;
padding-left: 10px;
outline: 5px solid hotpink;
outline-offset: 1px; }
```

Diagram



HTML part - program

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
      initial-scale=1.0">
    <title>css</title>
    <link rel="stylesheet" href=".style.css">
  </head>
  <body>
    <h1>Jspiders</h1>
    <h2 style="margin-top:50px; margin-bottom:-10px;">
      Jspiders</h2>
    <h1>Jspiders</h1>
  </body>
</html>

```

Facebook login Page Create

Create one button tag

button

{ background-color: blue;

}

→ login-fblue color bg.

→ color: white;

→ width: 200px; (get exact value by inspect)

to remove border

→ border: none

padding-top: 5px;

padding-bottom: 5px;

border-radius: 10px;

→ Right click on button in facebook page
inspect values.

Full program

button

```
{ background-color : blue ;  
Color : white ;  
width : 140px ;  
border : none ;  
padding - top : 5px ;  
padding - bottom : 5px ;  
border - radius : 5px ;
```

}

html part :-

```
<!DOCTYPE html>
```

<

<

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

```
<title> CSS </title>
```

```
<link rel = "stylesheet" href = ".\style.css" >
```

```
</head>
```

```
<body>
```

```
<button> Log in </button>
```

```
</body>
```

```
</html>
```

Display.html Display.css.

* why we use Display property

Display.html program

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset = "UTF-8" >
```

```
<meta http-equiv = "X-UA-Compatible" content = "IE=edge" >
```

```
<meta name = "viewport" content = "width=device-width,  
initial-scale=1.0" >
```

```
<title> CSS Display property </title>
```

```
<link rel="stylesheet" href=".display.css";
```

```
</head>
```

```
<body>
```

```
<P> JSP1</P>
```

```
<P> JSP2</P>
```

```
<P> JSP3</P>
```

```
<P> JSP4</P>
```

```
<h1> qsp1 </h1>
```

```
<h1> qsp2 </h1>
```

```
<h1> qsp3 </h1>
```

```
<h1> qsp4 </h1>
```

```
<a href="#"> psp1 </a>
```

```
<a href="#"> psp2 </a>
```

```
<a href="#"> psp3 </a>
```

```
<a href="#"> psp4 </a>
```

```
{ <span> spiders </span>  
  -||- moth  
  -||- butterfly  
<h1> -||- dragonfly
```

```
& 
```

```
& 
```

```

```

```
& 
```

Copy diff/
diff

img links

in diff, src
diff

diff

```
<P> JSP1</P>
```

```
<P> JSP2</P>
```

```
<P style="display:none;"> JSP3</P>
```

```
<P> JSP4</P>
```

```
<P style="visibility:hidden;"> JSP5</P>
```

```
<P> JSP6</P>
```

```
<P> JSP7</P>
```

```
<body>
```

```
</html>
```

Types of Elements

i) block level elements → CSS property applicable to whole element, Any ^{All} elements display in new line.

ii) Inline level → Any ^{(or) all} elements display in same line, CSS property apply to only content

Convert block level elements to inline level element
vice versa use display property.

→ Display property is used to change the default

→ Display = none property to hide the element

→ style-visibility: hidden

→ not occupy the space, hidden

→ it occupy the space.

→ display.css program

h1

{ border: 5px solid red;

display: inline;

h1 { border: 5px solid green; }

display: inline;

a { border: 5px solid red; }

display: block;

img { border: 5px solid green; }

height: 200px; width: 200px;

width: 200px; height: 200px;

display: block;

```
    } /* polyfill: placing 200px to element fixed height  
      and width of polyfill's elements off grid. Intrinsic width  
      P {  
        background-color: red;  
        height: 200px; /* placing 200px height  
        width: 200px;  
        /* margin: 60px; Intrinsic height should forward,  
        padding: 20px; */ /* polyfill's own margin size  
        display: inline-block;  
    }
```

y. *** flexbox.html, flexbox.css
In case of display property, more elements
div element is the parent, p is the children of the
div element.

→ display: flex

flex-direction → column or row by default it is
Column - reverse, row - reverse.

→ flex-wrap → It will bring the element to the
new line without space.

→ justify-content → property → center, start, end;
flex-end or end → represent horizontally

→ Space-around → space-between → space-evenly

flexbox.html program

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width,  
initial-scale=1.0">
```

```
<title> CSS Flex Box </title>
<link rel="stylesheet" href="style.css">
</head>
<body>
  <div>
    <p>JSP1</p>
    <p style="align-self: flex-start;">JSP2</p>
    <p style="align-self: center;">JSP3</p>
    <p>JSP4</p>
    <p>JSP5</p>
  </div>
</body>
</html>
```

flexbox.css program

```
div
{
  border: 2px solid purple;
  display: flex;
  /* flex-direction: column-reverse; */
  /* flex-direction: row-reverse; */
  flex-wrap: wrap;
  /* justify-content: center; */
  /* justify-content: start; */
  /* justify-content: end; */
  justify-content: space-around;
  /* justify-content: space-between; */
  /* justify-content: space-evenly; */
  height: 500px;
  /* align-items: center; */
  /* align-items: flex-start; */
```

```
align-items: flex-end; font-size: 30px; color: white; background-color: red; height: 200px; width: 200px; margin: 20px; }
```

Align Item

- align-items: center; want to represent vertically stretch
- style align-self: start; center;

these (this property used in p above program)

Create Facebook login Page

fblogin.html program

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Facebook Login</title>
<link rel="stylesheet" href=".\\fblogin.css">
</head>
<body>
<h1>facebook </h1>
<div>
<form action="">
<p>log in to facebook </p>
<input type="text" placeholder="Email address or phone number">
<input type="password" placeholder="password">
<button type="submit"> login </button>
</form>
</div>
</body>
</html>

```

fblogin.css program

```

body {
    background-color: #efefef;
}
h1 {
}

```

```
text-align: center;  
color: blue;
```

```
}
```

```
form
```

```
{
```

```
display: flex;
```

```
flex-direction: column;
```

```
justify-content: center;
```

```
width: 300px;
```

```
margin-left: 5px;
```

```
{
```

```
div
```

```
{
```

```
background-color: white;
```

```
height: 300px;
```

```
width: 350px;
```

```
margin: 50px auto;
```

```
display: flex;
```

```
flex-direction: column;
```

```
justify-content: space-around;
```

```
text-align: center;
```

```
border-radius: 5px;
```

```
{
```

```
input
```

```
{
```

```
border: 1px solid lightgray;
```

```
width: 100%;
```

```
margin: 4px;
```

```
padding: 15px;
```

```
border-radius: 9px;
```

```
outline: none;
```

```
{
```

```
button
```

```
{
```

```
padding: 21px;
```

position

background-color: blue;
 border: none;
 border-radius: 4px;
 color: white;
 font-size: 20px;
 font-weight: bold;

width: 336px;

margin-top: 14px;

}

14/1/22

Position property

Position: relative → from current position move left & right

top: 20px } it is moving

left:

create div in html and one is placed

<div> Parent
 <div> child </div> }
 add more children
 using inline css in div.

<div style="background-color: red; height: 10px; width: 10px;">

Parent <div style="position: absolute; top: -10px; left: -10px; background-color: blue; width: 20px; height: 20px;">

hello </div> <div style="position: absolute; top: 0px; left: 0px; background-color: blue; width: 20px; height: 20px;">

hello </div> <div style="position: absolute; top: 0px; left: 0px; background-color: blue; width: 20px; height: 20px;">

<div style="position: absolute; top: 0px; left: 0px; background-color: blue; width: 20px; height: 20px;">

<div style="position: absolute; top: 0px; left: 0px; background-color: blue; width: 20px; height: 20px;">

Position: relative: left;

→ Position absolute → from the parent div

move from parent current position

→ Parent must have position relative & child
 if we want to move child from body

child must have position absolute

- for selectors apply external.css.
- for navigation bar → position-sticky
→ position-fixed

Program position.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE-edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title> CSS position property </title>
    <link rel="stylesheet" href=". /position.css">
</head>
<body>
    <!-- <div>
        Jspiders
        </div> -->
    <P> Jspiders </P>
    <!-- <div style="background-color: orange; height: 60px;
position: sticky; top: 0px;"> I am Navigation bar </div> -->
    <div style="background-color: orange; height: 60px;
position: fixed; top: 0px; width: 100%; "> I am Navigation
bar </div>
```

Parent

hello {bold} **wildlife** criticism over farms in south
hello {bold} **criticism** over farms in south

```
<!-- <div style="background-color: green; width: 100px;  
height: 100px; color: white; position: relative; left: 200px;  
top: 40px; " > child </div > -->
```

```
<div style="background-color: green; width: 100px;  
height: 100px; color: white; position: absolute; top: 50px;  
left: 100px; " > child </div >
```

```
</div>
```

```
<p> Jspiders </p>
```

```
<p> Jspiders </p>
```

100 times

```
</body>
```

```
</html>
```

Position.CSS

```
/* div
```

```
{ background-color: red;  
height: 200px;  
width: 200px;  
position: relative;  
top: 20px;  
left: 200px;
```

```
} */
```

Example.CSS

→ pass one div tag

→ box-sizing → used to avoid ~~too~~ effect
the effect of Padding

→ padding affect the size

bootstrap
made
business
templates

elevanto
elements
1st link
↓
web templates

```
example.html program  
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta http-equiv="X-UA-Compatible" content="IE-edge">  
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
<title> CSS Examples </title>  
<link rel="stylesheet" href=". /example.css" type="text/css" media="all" />  
<head>  
<body>  
<!--<div>  
Jspiders Java training center  
</div>-->  
</body>  
</html>
```

example.css

```
div  
{  
background-color: red;  
height: 200px;  
width: 200px;  
padding: 50px;  
box-sizing: border-box;  
}
```

Web Design

Using Example program

- hover property → placing cursor (div:hover)
 - how to add video tag → youtube.com → tom & Jerry
 - Video → Copy control but video not supported.
 - without downloading video → give same URL to iframe → it gives msg like [] empty → bcz before key put forward slash / embed
 - i.e. "https://www.youtube.com/embed" remove some part & place this one adding
 - enable full screen → using tag & allowfullscreen
 - after link plat? & autoplay = 1 & mute = 1
 - autoplay is also used in normal video tag
- .html program

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE-edge">
  <meta name="viewport" content="width=device-width,
  initial scale=1.0">
<title> CSS Examples </title>
<link rel="stylesheet" href=".example.css">
</head>
<body>
<!-->
<div>
  Jspiders Java Training center
  </div> - ->
<iframe src="https://www.youtube.com/embed/
  xprpy1rxwMy?autoplay=1 & mute=1"
  height="400px" width="400px" allowfullscreen></iframe>
</body>
</html>

```

Example1.css

```

/* div
{ background-color:red;
height:200px;
width:200px;
padding:50px;
box-sizing:border-box;
}
div:hover
{
background-color:green;
color:white;
}
*/

```

SELECTORS

17/1/22

- Selecting the HTML (Elements) properties with the help of tag names like class, attributes like in multiways select.
- Element selector :- When we select only one type of element called Element selector
 - all paragraph, all H1, all a etc.
- Group selector :- When we select more than one element at a time.
 - P, div, a, h1 { color: red }

~~kkk~~

→ class selector → Indicate class, after pass class name. (anyname) (tagname.classname)

In .css) **h1.JSP**

color:red

In .HTML) **<h1 class="JSP> Qspider</h1>**

Generic class selector :-

#.JSP

```
{  
color:red  
}
```

}

Working is same as Generic.

ID Selector :- Working is same as Generic.

```
#dinga.  
{  
color:pink;  
}
```

}

Universal Selector :- It applies to all the tags in the program

*

```
{  
color:red;  
}
```

}

~~-----~~

```
<div> <span>Coloring</span> <br> -!>  
<div> <span>pink</span> <br> -!>  
<div> <span>blue</span> <br> -!>  
<div> <span>green</span> <br> -!>
```

```
'<div> <span>pink</span> <br> -!>  
<div> <span>blue</span> <br> -!>  
<div> <span>green</span> <br> -!>  
<div> <span>yellow</span> <br> -!>  
<div> <span>orange</span> <br> -!>  
<div> <span>red</span> <br> -!>  
<div> <span>purple</span> <br> -!>
```

multiple class -

Ex:- multiple class.html & .css

.btn-text-color → button {text-color: red;}

.btn-border

.btn-b-radius → button {border-radius: 50px;}

.btn-shadow

.btn-bg-color

→ button {background-color: blue;}

→ <div> Click me <div> anchor (button) </div>

Selectors, HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <title> CSS Selectors </title>
  <link rel="stylesheet" href=". /selectors.css" >
  <body>
    <!--<p id="dinga">Jspiders 1 </p>
      <p class="Jsp"> Jspiders 2 </p>
      <p> Jspiders 3 </p>
      <p> Jspiders 4 </p>
      <br>
      <h1 class="Jsp">Qspiders 1 </h1>
      <h1> Qspiders 2 </h1>
      <h1> class="Jsp"> Qspiders 3 </h1>
      <h1> class="Jsp"> Qspiders 4 </h1>
      <h1> class="Jsp" id="dinga"> Qspiders 5 </h1>
      <br>
      <div> pyspiders 1 </div>
      <div id="dinga"> pyspiders 2 </div>
      <div class="Jsp"> pyspiders 3 </div>
      <div> pyspiders 4 </div>
      <div> pyspiders 5 </div>
      <br>
      <h1 class="Jsp" id="dinga"> I am having both
        ID and Generic class Selector </h1> -->
```

```
<h1> outside div </h1>
<h1> outside div </h1>
<h1> outside div </h1>
<h1>
<div class="div-1">
  <h1> inside div 1 </h1>
  <h1> inside div 1 </h1>
  <h1> Hello Jspiders </p>
  <p> Inside div 1 </p>
  <h1> Inside div 1 </h1>
  <h1> Inside div 1 </h1>
</div>
```

```
<h1>
<div class="div-2">
  <h1> Inside div 2 </h1>
  <h1> Inside div 2 </h1>
  <h1> Hello Jspiders </p>
  <p> Inside div 2 </p>
  <h1> Inside div 2 </h1>
  <h1> Inside div 2 </h1>
</div>
```

```
<h1>
<h1> outside div </h1>
<body>
<html>
```

Selectors. CSS :-

/* Element Selector */

/* P

```
{ color:red;
} */
```

/* Group Selector */

/* p,h1,div

```
{ color:hotpink;
} */
```

/* class selector */

* h1.jsp

```
{ color:red;
} */
```

/* Generic class selector */

/* .JSP

```
{ color: red;  
  border-radius: 50px;  
}
```

/* ID Selector */

```
/* @ #dinga  
{  
  color green;  
  border: 2px solid red;  
}
```

/* ~~universal~~ Universal selector */

```
/* *  
{  
  color: red;  
}
```

/* Child selector */

```
/* div h1  
{  
  color: red;  
}
```

/* div > h1

```
{  
  color: red;  
}
```

MULTIPLE CLASSES .HTML

```
<!DOCTYPE HTML>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

```
  <meta name="viewport" content="width=device-width,  
  initial scale=1.0" />
```

```
<title> Applying multiple CSS classes </title>
```

```
<link rel="stylesheet" href="multiple classes.css" />
```

```
</head>
</body>
<button class="btn-txt-clr">Click </button>
<br>
<button class="btn-border"> Submit </button>
<br>
<button class="btn-bg-color"> Login </button>
<br>
<button class="btn-b-radius"> Signup </button>
<br>
<button class="btn-shadow"> Register </button>
<br>
<button class="btn-shadow btn-txt-clr btn-bg-clr
color btn-border btn-b-radius"> Subscribe
</button>
```

```
<br>
<button> Follow us </button>
```

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

Multiple classes. CSS

```
.btn-txt-clr
{
    color: white;
}

.btn-border
{
    border: 1px solid purple;
}

.btn-b-radius
{
    border-radius: 10px;
}

.btn-shadow
{
    box-shadow: 2px 2px 2px 2px red;
}

.btn-bg-color
{
    background-color: #49, 156, 149;
```

Cascading Style Sheets

CSS - Theoretical Part / Notes

- ⇒ It is used to design the web pages & to make web pages interactive.
 - ⇒ It is case insensitive language.
 - ⇒ There are 3 types of CSS or style sheets
- 1) Inline Style Sheets :-
 - 2) Internal style sheet (document level style sheet)
 - 3) External Style Sheets
- ⇒ With the help of Style tag we are able to define the CSS properties in web pages.

⇒ 1) Inline CSS :-

Syntax :- <tagname style="css property 1: value;
 css property 2: value2; ...> Content </tagname>

Ex:- <P style="color: red; background-color: pink;"
 Dinga</P>

- ⇒ Inside each & every tag if we will write the style property. It is known as Inline CSS.
- ⇒ It will consider the highest priority type.
- ⇒ ∵ it can override the values of internal as well as external also.

Ex:- if we wants to apply style to only one tag

<P style="color: red;">
 </P>

We can use separate style property to separate tag

2) Internal / Document level CSS :-

Syntax:-

<tagname{css property : value; ...}>

- ⇒ Inside the same html page if we will call the Style tag. It means inside the html page if we will apply CSS properties. It is known as internal CSS or Document level CSS or Embed CSS.

Syntax :-

<head>

<style>

element name tags {

// CSS code

}

</style>

</head>

ex:- h1 {

color: Aqua;

}

Ex :- <style>

P {
color: Red;

}

<style>

→ multiple element can be called
using one property.

ex:- P, h2 { color: red; }

}

; Semicolon is compulsory.
without ; this code will
not work.

Ex:- <head>

<style>

Tagname { css property: value; } forward of .

P, a { color: red; }

& <style> forward slash is also mandatory to

</head>

<body>

<P> Dinga </P>

<P> Dinga </P>

<P> JSP</P>

 QSP

</body>

3) External CSS: - file "22.css" = for CSS

→ If we wants to create separate page only for CSS

It will known as external CSS.

→ we have to save the page with the help of
.css extention.

Standard name → style.css

→ In html we should call link tag to connect
html with the CSS.

<link href="filename.css" rel="stylesheet">
(style.css)

→ In style.css no style tag is required.

Syntax

Tagname {
 css properties: value1;
 css properties: value2;
 css properties: value3;

css property n: value n;

```
{  
  color: green;  
}  
[saves style.css]
```

Inline style sheets :- whenever we are passing css properties using style attributes it becomes inline style sheets. It always overwrites the same properties of document level style sheets as well as external style sheets.

If properties are different then it inherits the properties from the external style sheets as well as from internal style sheets.

Ex:- <!DOCTYPE html>

<html>

<head>

<title> EST </title>

<link href = "alia.jpg" rel = "icon" />

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

</head>

<body>

<p> Ispider </p>

<p> Qspider </p>

 pypider

</body>

</html>

EST.html

```
P  
{  
  color: red;  
}  
[style.css]
```

Background Properties In CSS

- 1) Background - color : color code, color name
- 2) Background - image : URL (Image name)
- 3) Background - position : left (default), right, center
- 4) Background - Repeat : Repeat-x, Repeat-y, no repeat
- 5) Background - size : width, height, cover
- 6) Background - attachment : fixed, scrolled (default)

Document level :- whenever we are inserting CSS properties by using style tag then the style sheet becomes document level style sheets.
whenever we are passing CSS properties by creating separate file with .css extension then it becomes external style sheets.

External style sheets can be imported by using `<link>` tag. We can call (import) n number of style sheets in a single document.

Usage :-

```
<head>
<style>
<body>
    Background-color: pink;
    Background-image: url(images/.JPG);
    Background-position: center;
    Background-size: cover;
    Background-attachment: fixed;
    Background-repeat: no repeat;
    color: white;
</body>
</style> <h1> welcome </h1> - site 4
P { background-color: rgba(250,61,113,0.2);
    height: 500px;
    width: 500px;
}
</body> </head>
```

Font Properties In CSS

- 1) Font-style: Italic, Normal (Default)
- 2) Font-family: Any font family name.
- 3) Font-size: Px (for ex:- 80px)
- 4) Font-weight: Bold, lighter (default)
- 5) Font-variant: small-ops
- 6) Color: color code / name.

Usage:-

```
<body>
  <p style="font-size: italic"> Jspider </p>
  <p style="font-weight: bold"> Jspider </p>
  <p style="font-family: algerian"> Jspider </p>
  <p style="font-size: 100px"> Jspider </p>
</body>
```

Text Properties In CSS

- 1) Text-decoration: underline, Overline, line-through, none
- 2) Text-transform: uppercase, lowercase, Capitalize
- 3) Letter-spacing: Px
- 4) word-spacing: Px
- 5) line-height: px
- 6) Text-indent: px
- 7) Text-shadow: x-axis value, y-axis value, blur ratio, color
- 8) Color: color-name, colorcode
- 9) Text-align: left (default), Right, center

Usages:- {Style}

```
<body>
  <p style="color: rgba(255, 87, 99, 1)"> Jspider </p>
  <p style="text-indent: 8px"> Jspider </p>
  <p style="text-align: center"> Jspider </p>
  <p style="text-transform: uppercase"> Jspider </p>
  <p style="text-decoration: overline"> Dinga </p>
  <a href="#" style="text-decoration: none"> Dinga </p>
```

To remove the default underline provided by anchor tag.

HTML Lists

→ List is used to arrange the html elements in people order, there are 3 types of list.

1) List Ordered List (OL)

2) Unordered List (UL)

3) Definition List (DL)

OL Attributes

Type

Start

Parameters

1 (default), I, A, J, T

Any number

UL

Type

Disc (default), square, circle

Definition List:-

<dl> <dt> Javascript </dt>

<dd> Javascript is client side scripting language </dd>

<dt>

1) Programming Languages

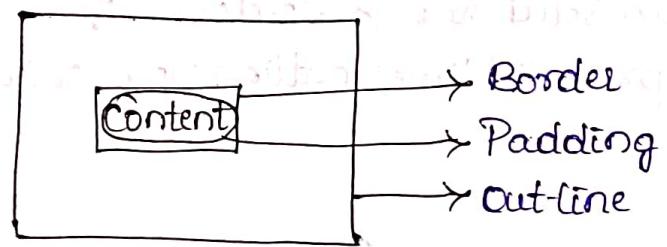
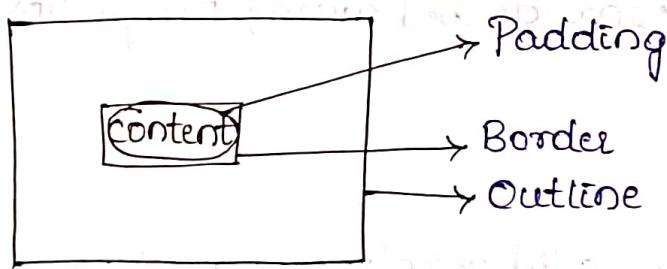
✓ Java ✓ JS ✓ HTML ✓ CSS ✓ SQL

2) Java Frameworks

✓ React JS ✓ Express JS

✓ Angular ✓ Vue JS ✓ Kime JS

BOX - MODEL:-



Box-Model consists of 4 parts

- 1) Border
- 2) Outline
- 3) Padding
- 4) Margin

Border Properties:-

- 1) Border-style: solid, double, dashed, dotted, groove, none
- 2) Border-color: color code / name
- 3) Border-width: px, %, auto
- 4) Border-radius: px (TLG, TRC, BRG, BLC)

short form

Border: width style color 1) Border-top-style: solid, groove
 2) Border-bottom-style
 3) Border-left-style
 4) Border-right-style

without using Border properties

Border: double 2px green

Border radius: 48px (Border radius is mandatory)

```
<P style="border-style: solid; border-color: red;
           border-width: 4px"> Jspiders </P>
<P style="border-style: dotted; border-bottom-color: blue;
           border-top-style: double"> Qspiders </P>
<P style="border-style: solid; border-radius: 30px 0px 30px
           0px"> PySpiders </P>
```

<P style="border: 2px dashed green"> Dinga </P>

</body>

Outline

```
<body>
<P style="border: 2px solid green"> Jspiders </P>
<P style="outline: 3px solid red"> Qspiders </P>
<P style="border: 2px solid lime; outline: 2px solid purple">
           PySpiders </P>
</body>
```

Outline Properties

- * Outline is created outside of a data property.
- 1) outline-style: solid, double, dashed, dotted
 - 2) outline-color: color code / name
 - 3) outline-width: px
 - 4) outline-radius: px
 - 5) outline-offset: px [To provide space b/w outline & border]
- Ex: outline-style: solid;
 outline-width: 2px;
 outline-color: green;
 outline-offset: 5px;
- outline: 3px solid green; outline: 3px solid green;
 outline-offset: 5px; outline-offset: 5px;

PADDING :- If we want to create gap b/w border to data we have to call padding properties.

- Properties :-
- 1) padding-top: px
 - 2) padding-bottom: px
 - 3) padding-left: px
 - 4) padding-right: px

```
<body>
<p style="border: 2px solid pink; background-color: lime; padding-top: 20px; padding-bottom: 10px; padding-left: 10px; padding-right: 10px;"> Ispiders </p>
<body>
```

MARGIN :- If we want to give gap b/w one border to another we use margin properties.

- 1) margin-top: px
- 2) margin-bottom: px
- 3) margin-left: px
- 4) margin-right: px

* Another use of Margin properties is to decrease the length of a border.

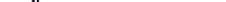
{Style}

```
img { border-radius: 100%; height: 200px; width: 200px; }  
button { background-color: lime;  
border: 2px solid red;  
outline: none;  
border-radius: 100%;  
padding: 10px; }
```

<{style}>

<thead>

{body}

 *limg src="1.jpg" > 'Lbal' >*

{button} click button{button}

31 body

{<html>}

→ To pass shadow to the boxes

box-shadow : x-axis y-axis Blue ratio color

Display Properties

1) Display Inline (Default), Block (default)

```
img { height: 200px; width: 200px; border: 4px solid red; border-radius: 100%; border-shadow: 4px 4px 3px lime; visibility: hidden }
```

Span hover { visibility: visible } 335 FC - 140px

<style>

<1head>

{body}

 Kajal Aggarwal

``

```
<body>
<p> JS <span style="color: red"> P </span> <br/>
</body>
```

span is a dummy tag which has no effect.

Span tag :- This is used to give different properties to different letters.

```
<link
<style>
```

```
P { position: relative;
```

```
top: 100px;
left: 350px;
```

input

visibility Property :- It is used to hide the content.

visibility: visible (default), hidden.

```
<body>
```

```
<p> welcome to Jspiders
```

```
welcome to Jspiders
```

```
<p>
```

```
<p style="visibility: hidden"> welcome to Jspiders
```

```
welcome to Jspiders </p>
```

```
</body>
```

```
</html>
```

Position Property :-

1) Position : state (default), Relative, sticky, fixed, absolute.

2) top: px

3) left: px

4) bottom: px

5) right: px

```
<head>
```

```
<style>
```

```
P { position: relative; left: 75px; top: 75px; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> Jspiders </p>
```

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

```
<style>
input { position: sticky;
        left: 160px; and make part property to be aware
        bottom: 0px; of its parent's height so it doesn't move
        border-radius: 50px; if parent's transform is
        outline: none; applied to it
    }
```

```
</style> <body>
<input type="text" placeholder="enter search element">
```

```
<p>
```

```
</p>
```

```
<body>
```

```
<html>
```

```
<style>
a { background-color: orange; height: 100px; }
b { background-color: green; height: 100px; }
img { position: relative; top: -200px; left: 250px; height: 100px;
      width: 100px; } 
```

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

```
<body>
```

```
<div class="a"></div>
```

```
<div class="b"></div>
```

```

```

```
</body>
```

Position Absolute:- It is always called inside the child

```
tag. <div class="main">
    <div id=" " ></div>
    <div id=" " ></div>
    <div id=" " ></div>
</div>
```

Inside child div we can use css property to call another div inside the parent div.

Position fixed:- When we scroll the page, all the things will scroll up. When we specify position: fixed; it is scroll the data upto the specified top data & it will stop to scroll up.

```
input {  
    border-radius: 30px;  
    position: sticky;  
    top: 10px;  
    left: 23px;  
}
```

SELECTORS:-

- Selectors are used to select the content we want to style.
- If we wants to make change inside the html properties we have to call selector.

Types of Selectors

- 1) Element Selector
- 2) Group Selector
- 3) class Selector
- 4) Generic Selector
- 5) ID Selector
- 6) Universal Selector.

1) Element Selector :- If we wants to apply some css properties for single element it will be known as Element Selector.

Syntax :- Tagname {css properties: values; }

Ex :- <head>

 <style>

 p {color: red;}

 </style>

</head>

<P> Dinga </P>

<P> Ding </P>

 Jsp

</body>

Element Selector selects the content based on Tagname.

2) Group Selector :- For multiple element if we wants to apply same css property. it will be known as Group Selector.

* we can call the elements with the help of ()

Ex:- P h3 h1 h2 {

color:red;
font-weight:bold; }
}

Syntax: - Tagname1, Tagname2 {css properties:values; } - }

Ex:- div, p, a { color:red; }

Group selector selects multiple tags at a time.

3) Class Selector: If we want to specify

→ For each and every tag if we wants to specify some particular CSS properties we have to call class selector.

→ we can create class with the help of (.) operator

→ we have to create the class with the help of element name.

→ we have to call the class property inside each & every element.

→ we can't call the same CSS property inside other elements.

Syntax :-

Element/tagname, class name {css properties:values; }

Ex:- {heads}

{style}

p.Dinga{color:red; }

{style}

{heads}

{body}

<p class="Dinga">Dinga</p>

<p> Dingi </p>

JSP

<div class="Dinga">&asp;</div>

{body}

→ Class selector selects the content based on tagname as well as class name.

→ Class selector can be called only inside particular tags.

4) Generic Selector :-

- Element specifying is not required.
- with the help of (.) operator we can create one class.
- we are able to call the class inside any tag.
- everytime it will return the data as obj.
- Syntax:-** Generic class selector begin with (.) It applied to any HTML element.
- .classname{css properties:values;...;}
- Generic selector Selects the content only based on classname. It can be called "Inside any tag".

Ex:- <head>

<style>

.Dingga{color:red;}

</style>

</head>

<body>

<p class="Dingga">Dingga</p>

<p>Dingga</p>

JSP

<div class="Dingga">ASP</div>

</body>

5) ID Selector :-

we can call the ID selector with the help of '#' operator.

- Each & every ID name should be unique.
- ID selector is the highest priority types in selector.
- Because it can override the value of class as well as generic.

Syntax:- # idname/classname{css properties:values;...;}

Ex:- <head>

<style>

#Dingga{color:red;}

</style>

</head>

<body>

<p id="Dingga">Dingga</p>

<p>Dingga</p>

JSP, <div id="Dingga">ASP</div>

→ Id selector selects the content based on id name whenever we are calling id selector and generic selector in a same tag then the preference will be given to id selector. id selector can be called inside any tag.

6) Universal Selector:-

→ Universal means all or everything.
→ If we wants to remove the default spaces from windows to data or if we wants to call universal selector.

→ Syntax:- * {css properties: values; }

Ex:- To remove default spaces

```
* {padding:0px;  
  Padding:0px;  
}
```

To universal selector

```
* {  
  padding:  
}
```

Ex:- <head>
<style>
* {color:red}
</style>

<head>
<body>
<p> Dinesh</p>
<p> Dinesh</p>
QSP
<div> HTML</div>
</body>

→ The CSS property is applicable to all the tags present in body.

Ex:- <!DOCTYPE html>

<html>

<head>

<title> ES </title>

<style>

* {color:red}

* {border: solid 2px lime}

* {background-color: pink}

```
<style>
</head>
<body>
<p class="b"> Dinga </p>
<p class="a b c"> Dingi </p>
<p> Jspiders </p>
<p> Aspiders </p>
<div> JSP </div>
<div class="a"> QSP </div>
<div> PSP </div>
<a href=""> HTML </a>
```

```
<style>
# a {color: red}
.b {background-color: pink}
#c {color: pink}
.d {color: green}

<style>
</head>
<body>
<p id="b"> Dinga </p>
<p id="a" class="b"> Dingi </p>
<p> Jspiders </p>
<p> Aspiders </p>
<div id="a"> JSP </div>
<div id="c" class="d"> QSP </div>
<div> PSP </div>
<a href="" id="a"> sheela </a>
<a href=""> leela </a>
<a href=""> laila </a>
</body>
```

```
<style>
.a {border: 2px solid red;}
.b {box-shadow: 5px 5px 3px lime;}
```

```
<style>
</head>
<body>
 br1
```

```
1 <p class="a"> Jspiders </p>
1 </body>
1 </html>
```

Web-site Development

18/1/22

- ✓ Bootstrap made → Templates
 - ✓ element elements: envato.com
 - ✓ shopify

✓ Bootstrap made → Business, Education
Empirical name

practise Bootstrap module

→ New folder create → Imperial name → open cmd → code → black screen displayed → create index.html → p In program title Imperial → link CSS → style.css → copy path & google → go to google search & pexabay search alphabets I → assign address in HTML → go to folder in HTML banner starts & ends write → create one div tag name as class: banner → add h1 tag → Imperial → h1 → welcome to Imperial studios → add Paragraph → to create working mobile apps → create & buttons → get started & our services → go to CSS file & select Universal selector → pass margin 0px → padding 0 → border box-sizing: border-box → font family: calibri → html, body → width 100% → overflow: x: hidden → create class for banner → .banner → height, bg color, → go pexabay image → in banner: place background image → Search buildings → in banner: place background image → no-background repeat → bground size 100% → background position: center → increase height width → how to apply black shade → for image linear gradient before URL → transparency → α value added. $\alpha = 1$ by default → Select black in RGB instead of RGB → pass min RGB values → change whole text color → want all text in center of banner → pass display flex-direction → flex direction columns → Justify Content center banner having 1 children → banner content → align-items: center → pass font styles in HTML class=heading 1 & 2 → class=para, class=btn → banner parent tag pass heading 1 → font family, size, blackadder etc apply → check in MS Word → same as second heading → h1 by select paragraph .banner para → & pass banner content in CSS → that center-align → h1

Select paragraph banner p → .banner
.css In that center-align → .center-align
Select button in banner class .banner button → .button
button class = "btn1, & btn2" → apply background color → apply
padding all 4 sides 10px → increase width 100px; Color of button
white - bold → font size 15px → font weight → include, banner.
btn2 → apply bg color for 2 buttons separately →
Placing cursor means hover → slowly display make past

- Using transition property: `bg-color 1s ease, 4sec;`
- At last apply margin to heading.
- hover is not a property it belongs to pseudocode concept

Imperial - Index.html

Imperial Style CSS

padding: space between border and content
margin: space between element and its container
border: border width, style, and color
font-family: font typeface

```
html, body
{width: 100%;  
overflow-x: hidden; /*
```

} /* Banner starts */

```
.banner
{height: 650px;  
background-image: linear-gradient(rgba(0,0,0,0.884),  
rgba(0,0,0,0.84)), url("https://bootstrapmade.com/  
demo/templates/Imperial/assets/img/hero-bg.jpg");  
background-repeat: no-repeat;  
background-size: 100% 100%;  
background-position: center;  
color: white;  
display: flex;  
flex-direction: column;  
justify-content: center;  
align-items: center;
```

}

```
.banner-content
{text-align: center;}
```

}

```
.banner .heading-1
{font-size: 85px;  
font-family: Segoe Script;}
```

}

```
.banner .heading-2
{font-size: 45px;  
margin-top: 20px;  
margin-bottom: 10px;}
```

}

```
.banner P
{font-size: 35px;  
font-weight: 200;}
```

}

```
.banner .btn1, .banner .btn2
{width: 161px;  
color: white;  
font-weight: 700;  
font-size: 16px;}
```

```
.border-radius: 3px;  
padding: 10px;  
transition: background-color 4s, border 4s;  
margin-top: 40px;}
```

}

```
.banner .btn1
{background-color: #007bff;  
background-color: transparent;  
border: none;}
```

}

```
.banner .btn1:hover
{background-color: transparent;  
border: 2px solid white;}
```

}

```
.banner .btn2
{background-color: transparent;  
border: 2px solid white;  
padding-top: 8px;  
padding-bottom: 8px;  
margin-left: 14px;}
```

}

```
.banner .btn2:hover
{background-color: #007bff;  
border: none;}
```

}

```
/* banner ends */
```

div class = "banner-btn"

19/1/22

Navigation bar → take "nav bar", starts & ends in ~~ul~~,
code → In HTML → inside nav tag pass ul tag →
~~ul~~ inside that ul pass li tag → take a tag for
all values name, about, services -

before UL tag pass one anchor tag for Imperial →

In .css → nav box starts & ends → take nav tag →
apply bg color → black → go to anchor tag in ~~ul~~ tag;
side nav bar → pass white color → text decoration → none
display: inline → letter spacing → font family →
margin-left → apply some margin to the UL tag →
hover → transition: 0.5s → let slice 200 pixels →

- Continue program after banner ends in .css

/* Navbar starts */

```
nav { background-color: rgba(0,0,0,0.8);  
      color: white;  
      text-decoration: none;  
      font-weight: bold;  
      transition: color 0.5s;  
      padding: 10px 0px;  
      width: 100%; }  
nav ul { list-style-type: none;  
        padding: 0px;  
        margin: 0px; }  
nav ul li { border-bottom: 1px solid black; }  
nav ul li a { display: inline-block; width: 100%; text-align: center; padding: 10px 0px; }  
nav ul li a:hover { color: red; }
```

Search google

for our font awesome
see cdn link
→ 1st click

In Load all styles,
copy link see
place in html
one more link

copied with .css
again search
fb icon & click
on that copy
HTML & replace

in .css in the
place of I tag

color: #222; font-size: 1em; font-weight: bold; }

```
nav ul { font-size: 1em; margin-left: 214px; } .brand-logo { font-size: 40px; color: white; text-decoration: none; font-family: Segoe Script; letter-spacing: 1px; margin-left: 143px; } /* Nav bar Ends */ /* aboutus starts */ .aboutus { background-color: pink; display: flex; flex-direction: column; justify-content: center; margin: 80px; } .aboutus .abouth1, .aboutus .aboutpara, .aboutus hr { text-align: center; } .aboutus .abouth1 { font-size: 40px; } .aboutus hr { width: 60px; height: 4px; background-color: cyan; border: none; margin: 20px auto; } .aboutus-child { /* border: 5px solid purple; */ display: flex; justify-content: space-between; margin-top: 20px; } .aboutus-img { /* border: 2px solid red; */ width: 50%; margin: 20px; } .aboutus-img img { height: auto; width: 100%; } .aboutus-content { /* border: 1px solid black; */ } .aboutus-content h2 { font-size: 1.5em; margin-bottom: 10px; } .aboutus-content p { font-size: 1em; margin-bottom: 10px; } .aboutus-content hr { width: 100px; height: 1px; background-color: cyan; border: none; margin: 20px auto; }
```

Imperial HTML for Navbae & About us

<!- Navbae starts ->

<nav>

 Imperial

 Home

 about

 services

 Portfolio

 Testimonials

 Team

 Dropdown

 Contact

<nav>

<!-- Nav Bar ends -->

<!-- Aboutus starts -->

<div class="about-us">

<h1 class="about-us-h1"> ABOUTUS </h1>

<h2>

<p> class="about-us-Para" - - - - - </p>

<div class="about-us-child">

<div class="about-us-img">

</div>

```
<div class="about-us-content">  
  <h1>WE - - - </h1>  
  <p>Loren<br/>Loren<br/>Loren<br/>Loren<br/></div>
```

```
</div>  
<!-- about-us ends -->  
Services part HTML program.
```

```
<!-- Services starts -->  
<div class="services">  
  <h1 class="services-h1">OUR SERVICES</h1>  
  <hr>  
  <p class="services-para">At - - - - </p>  
  <div class="services-child">  
    <div class="services-info">  
      <div class="service-icon">  
        <i class="far fa-briefcase"></i>  
      </div>  
      <div class="services-content">  
        <h1>LOREM IPSUM </h1>  
        <p> - - - - </p>  
      </div>  
    </div>
```

```
<div class="service-info">  
  <div class="services-icon">  
    <i class="far fa-clipboard-list-check"></i>  
  </div>
```

```
<div class="services-content">  
  <h1>DOLAR SISTEMA </h1>  
  <p> - - - - </p>  
</div>
```

```

<div class="Service-info">
    <div class="Service-icon">
        <i class="fas fa-chart-bar"></i>
    </div>
    <div class="Service-content">
        <h1>SED UT PERSPICIANTS</h1>
        <p>--<br/>--</p>
    </div>
</div>

<div class="Service-info">
    <div class="Service-icon">
        <i class="fas fa-binoculars"></i>
    </div>
    <div class="Service-content">
        <h1>MAGNI DOLORS</h1>
        <p>--<br/>--</p>
    </div>
</div>

<div class="Service-info">
    <div class="Service-icon">
        <i class="fas fa-lightbulb-on"></i>
    </div>
    <div class="Service-content">
        <h1>EIUMOD TEMPOR</h1>
        <p>--<br/>--</p>
    </div>
</div>

```

<!- Services ends ->

For Services.css program.

```

/* Services starts */
.Services {
    background-image: linear-gradient(rgba(255, 255, 255, 0.828), rgba(255, 255, 255, 0.849)), url("img/1");
    background-repeat: no-repeat;
    background-clip: 100% 100%;
    background-attachment: fixed;
}

```

```
display: flex;
flex-direction: column;
justify-content: center;

{
    services .Services-h1,
    services .Services-para,
    services hr
    {
        text-align: center;
    }

    services .Services-h1
    {
        font-size: 40px;
    }

    services hr
    {
        width: 60px;
        height: 4px;
        background-color: cyan;
        border: none;
        margin: 20px auto;
    }

    services -child
    {
        display: flex;
        flex-wrap: wrap;
        justify-content: space-around;
        margin-top: 20px;
        align-items: center;
    }

    .Services -info
    {
        width: 400px;
        padding: 20px;
        height: 250px;
        margin: 20px;
        display: flex;
        justify-content: space-around;
        align-items: flex-start;
        margin-top: -20px;
    }

    .Services -icon
    {
        margin: 5px;
        flex-basis: 150px;
        margin-top: 35px;
    }

    .Services -content
    {
        margin: 5px;
        margin-left: 10px;
    }

    .Services -icon-ly
    {
        height: 80px;
        width: 80px;
        background-color: #03C4EB;
        padding: 24px, 26px;
        border-radius: 100%;
        color: white;
        border: 2px solid #03C4EB;
        transition: all 0.5s;
        font-size: 25px;
    }

    .Services -info: hover
    .Services -icon-ly
    {
        background-color: transparent;
        border: 2px solid #03C4EB;
        color: #03C4EB;
    }

    .Services -content h1
    {
        margin-bottom: 20px;
        font-size: 25px;
    }

    .Services -content p
    {
        color: gray;
    }

/* Services ends */

```

for Subscribe

```

<!-- Subscribe starts -->
<div class="subscribe">
  <div class="subscribe-text">
    <h1>Subscribe for updates </h1>
    <p>--</p>
  </div>
  <div class="subscribe-btn">
    <button>Subscribe Now</button>
  </div>
</div>
<!-- Subscribe ends -->

```

.CSS

```

/* Subscribe starts */
.subscribe {
  height: 300px;
  background-image: linear-gradient(rgba(0,0,0,0.683),
    rgba(0,0,0,0.683)), url("");
  background-repeat: no-repeat;
  background-size: 100% 100%;
  background-attachment: fixed;
  color: white;
  align-items: center;
  justify-content: center;
}

.subscribe-text {
  margin: 80px;
  width: 650px;
}

.subscribe-btn {
  margin: 80px;
}

.subscribe-btn button {
  font-size: 18px;
  font-weight: 450;
}

```

HTML program

```

background: transparent;
color: white;
margin: 0;
border: 2px solid white;
padding: 7px 20px;
border-radius: 3px;
letter-spacing: 1px;
transition: all 0.5s;
width: 200px;
text-align: center;
}

.subscribe-btn button:hover {
  background-color: #03C4EB;
  border: 2px solid #03C4EB;
}

.subscribe-text h1 {
  margin-bottom: 20px;
}

.subscribe-text p {
  font-size: 20px;
}

/* Subscribe ends */

```

for Portfolio → HTML program

```
<!-- Portfolio starts -->  
<div class="Portfolio">  
  <h1> PORTFOLIO </h1>  
  <br>  
  <p> - - - - - </p>  
  <div class="Portfolio-child">  
    <div class="portfolio-image">  
        
      <div class="img-content">  
        APP1  
        </div>  
    </div>  
    <!-- portfolio is same for further 8 images but src  
    different after those -->  
  </div>  
<!-- Portfolio ends -->
```

.CSS program

```
* portfolio starts *!  
.Portfolio  
{  
  margin: 10px;  
}  
.Portfolio h1, .Portfolio p,  
.Portfolio hr  
{  
  text-align: center;  
}  
.Portfolio hr  
{  
  background-color: #034EB;  
  width: 80px;  
  height: 6px;  
  border: none;  
  margin: 20px auto;  
}
```

```
.Portfolio-child  
{  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: space-around;  
}  
.Portfolio-image img  
{  
  height: 100%;  
  width: 100%;  
}  
.img-content  
{  
  position: absolute;  
  bottom: 0px;  
  left: 0px;  
  height: 100px;  
  width: 100px;  
  background-color: rgba(0, 0, 0, 0.7);  
  font-weight: bold;  
  text-align: center;  
  visibility: hidden;  
  transition: all 0.7s;  
}
```

```
• portfolio-image:hover .img-content {  
  visibility: visible;  
}  
/*portfolio ends */
```

For Testimonials .HTML Program

```
<!-- Testimonials start -->  
<div class="Testimonials">  
  <h1> TESTIMONIALS </h1>  
  <hr>  
  <p> Endo <br>   
  
<div class="testimonial-child">  
  <div class="testimonial-content">  
    <div class="image">  
        
    <h2> Saul Goodman </h2>  
    <p> Lawless Inc </p>  
    <div>  
      <div class="Text">  
        <p><i class="fas fa-quote-left"></i> Prolo  
          - - - - - <i class="fas fa-quote-right"></i> </p>  
    </div>  
    <div>  
      <div class="testimonial-content">  
        <div class="text">  
          <p><i class="fas fa-quote-left"></i> sed ut  
            - - - - - <i class="fas fa-quote-right"></i> </p>  
        </div>  
        <div class="image">  
            
        <h2> Sarah Wilson </h2>  
        <p> Odeo Inc </p>  
      </div>  
    </div>  
  </div>  
<!-- Testimonials end -->
```

• CSS

```
* testimonials starts */
.Testimonials
{
    background-color: lightgray;
    padding: 20px 10px;
}

.testimonials h1, .testimonials P,
.testimonials hr
{
    text-align: center;
}

.testimonials hr
{
    background-color: #03C4EB;
    width: 80px;
    height: 6px;
    border: none;
    margin: 20px auto;
}

.testimonial-child
{
    margin: 80px;
    width: 1170px;
}

.testimonial-content
{
    display: flex;
    justify-content: space-around;
}

.testimonial-content .image img
{
    width: 100px;
    border-radius: 100px;
    border: 6px double white;
}

.testimonial-content .text
{
    padding: 40px;
    padding-top: 100px;
    font-size: 20px;
    background-color: white;
    margin: 40px; color: gray;
}

/* testimonial ends */
```

• testimonials - content .image

```
{ width: 350px;
```

• testimonial content p

```
{ margin-top: 48px;
```

• fa-quote-left, fa-quote-right

```
{ font-size: 30px;
    color: rgba(173, 216, 230, 0.609);
```

• testimonials - content .image

```
{ margin-top: 17px;
```

• testimonials - content .image h2

```
{ padding: 0px;
    color: rgb(0, 195, 255);
```

• testimonial-content .text

```
{ padding: 40px;
    padding-top: 100px;
    font-size: 20px;
    background-color: white;
    margin: 40px; color: gray;
```

for Contact us • HTML program

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

.css program :-

```
* Contact us starts */  
.contactus  
{ background-color: rgba  
(211, 211, 211, 0.205);  
padding: 20px 0px;  
}  
.contactus h1, .contactus p,  
.contactus hr  
{ text-align: center;  
}  
.contactus hr  
{ background-color: #03C4EB;  
width: 80px;  
height: 6px;  
border: none;  
margin: 20px auto;  
}  
.contactus-child  
{ margin-top: 40px;  
display: flex;  
justify-content: space-around;  
}  
.contactus-address  
{ width: 500px;  
display: flex;  
justify-content: end;  
}  
.contactus-form  
{ width: 600px;  
}  
.contactus-inputs  
{ width: 450px;  
display: flex;  
flex-direction: column;  
}
```

```
A. contactus-inputs div,  
• contactus - inputs input,  
• contactus - inputs textarea,  
• contactus - inputs button  
{ width: 100%;  
}  
.contactus-inputs input,  
.contactus-inputs textarea  
{ margin: 8px;  
padding: 8px 10px;  
border: 1px solid rgba(128,  
128, 128, 0.315);  
outline: none;  
}  
.contactus-inputs input: focus,  
.contactus-inputs textarea: focus  
{ border: 1px solid #03C4EB;  
}  
.contactus - inputs button  
{ width: 150px;  
padding: 10px 8px;  
background-color: #03C4EB;  
color: white;  
border: none;  
margin-left: 150px;  
}  
.contactus - address P  
{ font-size: 20px;  
}  
.fa-map-marker-alt,  
.fa-envelope, .fa-mobile  
{ color: #03C4EB;  
font-size: 35px;  
}  
::placeholder  
{ font-size: 20px;  
}  
/* Contactus - ends */
```

Assignment:- Medilab.

Medicio
Monitoring device
Control system
FEELY

Address: Singapore 223.

- Web Technology & It is the way the devices communicate with each other using markup language.
- Web Technologies allows companies to design a certain image & to attract and generate customer loyalty.

- * For-in-loop in JS → is used to loop through the properties of an object. syntax:

```
for (variable name in object){  
    statement or block of to execute  
}
```

In each repetition, one property from the object is associated with the variable name. The loop is continued till all the properties of the object are depleted.
 - * For-of-loop →

- What's a pool of browser objects & what's a global object?
- Javascript code can be embedded in a web page below <script language="Javascript"></script> tag.
- What's negative infinity? → It's a no. in JS, derived by dividing negative no. by zero.
- What's this keyword? → It refers to the current object.
- What's a way to append a value to an array?
arr[arr.length] = value;
- How do we assign object properties?
obj["age"] = 17 or obj.age = 17.
- Boolean operators support in JS + & , || and !
- JS is a client-side & server-side scripting language inserted into HTML pages & is understood by Web browsers. JS is also an Object-based PL.
- What's isNaN function? isNaN returns true if the argument is not a no.; otherwise it is false.
- ASP → Active Server Page Scripting condition
JS is faster than ASP.
- for single line comment //
multiline /* ... */.
- How can the style/class of an element be changed?
document.getElementById("myText").style.fontSize = "20";
- How to read & write a file using JS?
→ Using JS Extentions, using webpage & Active X-objects
- Variable typing in JS → is used to assign a no. to a variable. & same variable can be assigned to a string.
Ex i=10; i="string";
- parseInt() is used to convert string of any base to integer in JS.
- Null in JS → Null Value is used to represent no value or no object. It implies no object, or null string, no valid boolean value, no no. for array object.
- JS does not have Concept-level scope. The variable declared inside the func. has scope inside the func.
- Keywords are used to handle the exceptions.
 - Try, Catch, finally
- Load time error, & running time error, logical error