* Types of CSS

1. Inline CSS :- An inline style may be used to apply a unique style for a single element.

de:-
```
<!DOCTYPE html>
<html>
<body>
<h1 style = "color : blue; text-align : center ;">
    This is a heading </h1>

<p style = " color : red; "> This is a paragraph </p>

</body>
</html>
```

2. Internal CSS :- An internal style sheet may used if one single HTML page has a unique style.

e:
```
<!DOCTYPE html>
<html>
<head>
<style>
        body {
            background-color : linen;
        }
    h1 {
        color : maroon;
        margin-left : 40px;
}
</style>
</head>
<body>
```

<h1> This is a heading </h1>
 This is a paragraph

</body>
</html>

(3) External CSS :- with an external style sheets you can change the look of an entire website by changing just one file!

Code:- <! DOCTYPE html>
<html>
<head>
<link rel = "stylesheet" href = "mystyle.css" >
</head>
<body>
<h1> This is a heading </h1>
 This is a paragraph

</body>
</html>

mystyle.css

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```

* CSS selectors

1. Element selectors :- The element selector selects html element based on the element name.

Code :-

```
<html>
<head>
<style>
        p{
            text-align : center;
            color : green;
        }
</style>
</head>
<body>

<p> Every Paragraph will be affected by the style </p>

<p id = "para1" > me too ! </p>

<p> And me ! </p>

</body>
</html>
```

2. Id selector :- The id selector uses the id attribute of an HTML element to select a specific e

```
<html>
<head>
<style>

#para1 {
        text-align : center;
```

```
        color: green;
    }
</style>
</head>
<body>

<p id = "para1"> Hello world! </p>

</body>
</html>
```

(3) class selector :- The class selector selects HTML elements with a specific class attribute.

code :-

```
<html>
<head>
<style>

.center
    text-align: center;
    color : red;
}
</style>
</head>
<body>

<h1 class = "center"> Red and center-aligned heading. </p>

</body>
</html>
```

4. Universal selector:- The Universal (*) selects all HTML element on the page:

Code:-

```
<html>
<head>
<style>

*{
    text-align : center;
    color : blue;
}
</style>
</head>
<body>
```

 Every element on the page will be affected the style.

```
<p id = "para"> me too! </p>
<p> And me! </p>
</body>
```

5. Grouping selector:- The grouping selector selects all HTML elements with the same style definitions

Code:-

```
<html>
<head>
<style>
        h1 , h2 , p {
            text-align : center;
            color : red;
        }
</style>
</head>
```

<h1> Hello world! </h1>
<h2> smaller heading! </h2>
 this is a paragraph

</body>
</html>

* Grey layout code:-

-> <html>
<head>
<style>
#R1{
background-color: grey;
height: 100%; text-align: center;
width: 60 px;
padding-top: 20 px;
Padding-bottom: 70px;
}

#R2, #R3{
background-color: grey;
height: 33px; width: 100%.
text-align: center;
margin-bottom: 10px;}

#R3{ background-color: grey;
height: 33 px; width: 100%;
text-align: center;
margin-bottom: 10 px;}

#R4{ height: 600px; width: 100%;
margin-bottom: 10 px;}

```
#R1C1 { width : 28% ; margin - right : 2% ; }
#R1C2 { width : 70% }

#R1 div { float : left ; height : 320 px ;
          Putting - top : 280 px ;
          text - align : center ;
          background - color : gray ; }
```

```
<style>
<head>
<body>
```

```
<div id = "R1" > Logo </div>
<div id = "R2" > Navigation </div>
<div id = "R3" > Header / Banner </div>
<div id = "R4" >
    <div id = "R1C1" > sidebar </div>
    <div id = "R1C2" > Body area </div>
</div>
```

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

2.

```
<! Doctype html>
<html>
<head>
<style type = "text/css" >

#R1 { back ground - color : gray
      font - weight : bold ; height : 120 px ;
      Putting - top : 30 px ; width : 95% ;
      Putting - left : 5% ;
      margin - bottom : 10 px ; }
```

```
# R2, #R3, # R5 { background-color: gray;
                  height: 33%; width: 100%;
                  text-align: center;
                  padding-top: 1%;
                  margin bottom; 10px; }

#R4 { height: 600px; width: 100%;
      margin: bottom; 10px; }

# RUC1 { width: 32%; margin-right: 1%;

# RUC2 { width: 32%; margin-right: 1%;
         margin-left: 1%; }

# RUC3 { width: 32%; margin-right: 1%; }

# R4 div { float: left; height: 320 px;
           padding-top: 10px;
           text-align: center;
           background-color: gray; }
```

```
<style>
<head>
<body>

  <div id="R1"> logo </div>
  <div id="R2"> Navigation </div>
  <div id="R3"> header Banner </div>
  <div id="R4">
  <div id="RUC1"> sidebar </div>

  .

  <div id="RUC2"> Body-area </div>

  .

  <div id="RUC3"> Body text </div>
```

</body>
</html>

* video positioning effect *

-> <!doctype html>
<html>
<head>
<meta name = "viewport"
content = "width = device - width, initial - scale = 1">
<style>
* {
box-sizing : border - box;
}

body {
margin : 0;
font - family : Arial;
font - size : 17 px;
}

#myvideo {
position : fixed;
right : 0px;
bottom : 0px;
min-width : 100%;
min - height : 100%;
}

.content {
position : fixed;
bottom : 0px;
background : rgba(0,0,0,0.5);

```
    color : #f1f1f1
    width : 100%;
    padding : 20px;
}

#myBtn {
    width : 100px;
    font-size : 18px;
    padding : 10px;
    border : none;
    background : #000;
    color : pink;
    cursor : pointer;
}

#myBtn : hover {
        background : #ddd;
        color : black;
}
</style>
</head>
<body>

<video autoplay muted loop id ="myVideo (mp4" >
<source src ="sun.mp4" >
    your browser does not support HTML5 video.
</video>

<div class ="content" >
    <h1> heading </h1>
<p> this is a sivas ook university </p>
<button id ="myBtn"
    onclick ="myFunction()" pause </button>
</div>
```

```
<script>
Var video=
    document.getElementById("myvideo");
Var btn=..
    document.getElementById("myBtn");

function myfunction(){
    if(video.paused){
        video.play();
        btn.innerHTML="pause";
    }
else{
    video.pause();
    btn.innerHTML="play";
}
}
</script>
</body>
</html>
```

* media query

```
-> <!Doctype html>
   <html>
   <head>
   <meta name="viewport"
      content="width=device-width, initial-scale=1.0">
</style>
   *{
   .wrapper{overflow:auto;}

   #main{margin-left:10px;}
```

```
#left-sidebar {
    float: none;
    width: auto;
}

#menu-list {
    margin: 0;
    padding: 0;
}

.menu-item {
    background: #CDEDFC;
    border: 1px solid black;
    border-radius: 4px;
    list-style-type: none;
    margin: 4px;
    padding: 2px;
}

@media screen and (min-width: 480px)
{
#left-sidebar {width: 200px; float: left;}

#main {margin-left: 216px;}
}
</style>
</head>
<body>

<div class="wrapper">
<div id="left-sidebar">
<ul id="menu-list">
  <li class="menu-item">menu-item
</li>
```

```
<li class="menu item"> menu - item 2 </li>
<li class="menu item"> menu - item 3 </li>
<li class="menu item"> menu - item 4 </li>
<li class="menu item"> menu - item 5 </li>
</ul>
</div>

<div id="main">
    <h1> Resize the browser window to see the effect! </h1>

<p> this example shows a menu that will float to the left of the page if the viewport is 1150 pixels wide or wider </p>

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

* Image Hover Effect

->
```
<!DOCTYPE html>
<html>
<head>
<meta name="view port"
  content="width=device-width, initial-scale=1">
</style>
```

. Container
```
    position: relative;
    width: 50%;
}
```

```
.image {
    display: block;
    height: auto;
    width: 100%;
}

.overlay {
    position: absolute;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    height: 100%;
    width: 100%;
    opacity: 0;
    transition: 0.5s ease;
    background-color: #008CBA;
}

.container:hover .overlay {
    opacity: 1;
}

.text {
    color: white;
    font-size: 20px;
    position: relative;
    top: 50%;
    left: 50%;
    -webkit-transform: translate(-50%, -50%);
    -ms-transform: translate(-50%, -50%);
    transform: translate(-50%, -50%);
    text-align: center;
}
```

</style>
</head>
<body>

<h2> fade in overlay </h2>
 Hover over the image to see the effect.

<img src = "img_avtar.png" alt = "avatar" class="ima

 Hello world

</body>
</html>

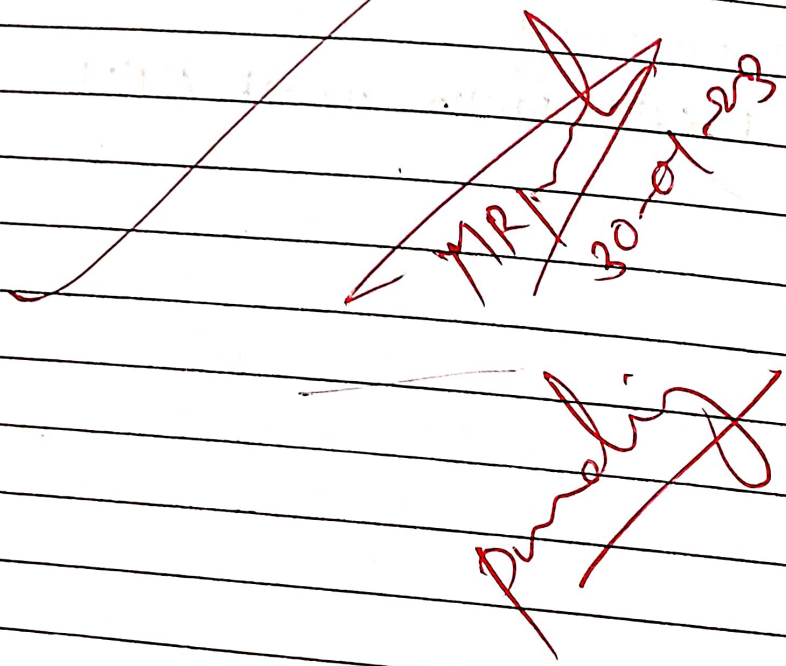
* font awesome.

<!DOCTYPE html>
<html>
<head>
<title> font awesome I cons </title>
<meta name = "view Port"
content = "width = device-width, initial-scale = 1" >

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

</head>
</body>

```
<i class = "fa fa-car"></i>
<i class = "fa fa-car" style = "font-size:48px;"></i>
<i class = "fa fa-car" style = "font-size:60px;
    color:red;"></i>
</body>
</html>
```



Bootstrap

Date
Page

★ What is Bootstrap?

=> Bootstrap is a free front end framework for faster and easier web development.

-> Bootstrap includes HTML and CSS based design templates for typography, forms, button, tables, navigation, modals, image carouses and many other, as well as optional JavaScript plugins.

-> Bootstrap also gives you the ability to easily create responsive designs.

example:

```
<html>
<head>
<title> Bootstrap Example </title>
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@
  5.2.3 /dist/css/bootstrap.min.css#" rel = "stylesheet">

<script src = "https://cdn.jsdelivr.net/npm/bootstrap@ 5.2.3
  dist/js/bootstrap.bundle.min.js"></script>

</head>
<body>

<div class = "container-fluid p-5 bg-primary text-white
  text-center">

<h1> my first Bootstrap Page </h1>
<p> Resize this responsive page to see the effect!</p>
</div>
```

```
<div class = "container mt-5">
    <div class = "row">
    <div class = "col-sm-4">
    <h3> column 1 </h3>
<p> this is the sivas ooh university and we use
    in the socca </p>

<div class = "col-sm-4">
<p> this is the sivas ooh university and we use
    in the socca </p>

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

★ why use Bootstrap ?

Advantages of Bootstrap.

-> Easy to use :- Any body with just basic knowledge of HTML and CSS can start using Bootstrap.

-> Responsive features :- Bootstrap responsive css. adjust. to phones tables and desktops.

-> Mobile - first approach :- In Bootstrap. mobile-first styles are part of the core framework throughout.

→ Browser compatibility :- Bootstrap 5 is compatible with all modern browsers. Note that if you need support for IE 11 and down, you must use either BS4 or BS3.

★ Bootstrap 5 CDN

If you don't want to download and host BS 5 yourself, you can include it from a CDN (Content Delivery network).

Jsdelivr provides CDN support for Bootstrap CSS and javascript.

Maxcdn :-

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>

Create your first web page with Bootstrap 5

1. Add the HTML 5 doctype
Bootstrap 5 uses elements and css properties that require the HTML 5 doctype

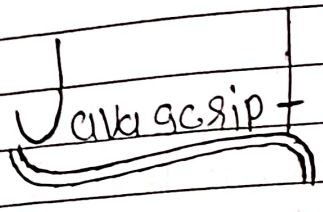
```
<html>
<head>
<title> Bootstrap 5 example </title>
</head>
</html>
```

★ container
Bootstrap 5 also requires a containing element to wrap site contents.

There are two containers class to choose from:

1. the container class provides a responsive fixed width container.

2. the .container-fluid class provides a full width container, spanning the entire of the viewport.



★ why study Javascript?

-> Javascript is one of the 3 langages on web developers must learn.

-> Html to define the content of web pages.

-> css to specify the layout of web pages.

-> Javascript to program the behavior of wel pages

-> Javascript can change HTML content

-> one of many javascript methods is getElement ById().

Date
Page

```
<html>
<body>
<h2> what can JS do?. </h2>

<p id = "demo"> JS can change Html content. </p>

<button type = "button" onclick = 'document.getElement
By Id ("demo") innerHtml = " Hello Javascript !" > click me!
</button>

</body>
</html>
```

-> Javascript can change Html Attribute values.

-> In this example Javascript changes the value of the src (source) attribute of an <img> tag.

```
<html>
<body>
<h2> what can JS do? </h2>
<p> In this case Javascript changes the value of
the src attribute of an image. </p>

<button onclick = " document.getElement By Id ('my Image
src = 'pic_bulbon.gif' > Turn on the light </button>

<img id = " myImage" src = "pic_bulboff.gif" style = "width:
100px" >

<button onclick = " document.getElement By Id ('my Image').
src = " pic_bulboff.gif" > Turn of the light </button>

</body>
</html>
```

Date
Page

-> ~~the~~ ~~what is can do? </h2>~~
JavaScript can change Html styles (css).

<html>
<body>
<h2> what is can do? </h2>

 is can change the style of an Html element

<button type = "button" onclick = "document.getElement
BY Id ('demo') style. font size = '35 px'"> click me! </button

</body>
</html>

-> JavaScript can ~~hide~~ Html elements.

-> Hiding Html elements can be done by changing the display style;

<html>
<body>
<h2> what can JS do? </h2>
 JavaScript can hide Html elements

<button type = "button" onclick = "document". get
Element By Id ('demo'). style display = 'none'> click m
</button>

</body>
</html>

-> and use the display = "block"; to show the Html element.

Date
Page

Javascript where to

-> The <script> tag

```
<html>
<body>
 <h2> JS in body </h2>

<p id = "demo" </p>

<script> document.getElementBYId ("demo").innerHtml = "
 my first Javascript" : </script>
</body>
</html>
```

Javascript functions and events.

-> A Javascript functions is a block of Javascript code that can be executed when "called" for.

For example a funtion can be called when an event occurs. like when the uses clicks a button.

Javascript in Head.

```
<html>
<head>
<script>
   function myFuntion () {document.getElement By Id
     ("demo"). innerHtml = "Paragraph changed" }

</script>
</head>
<body>
```

<h2> demo JS in Head </h2>
 A paragraph.
<button type = "button" onclick = "my function c()">
Try it </button>

</body>
</html>

=> External JavaScript.

Script can also be placed in external files.

-> JavaScript files have the file extension .js.

-> To use an external script. put the name of the script file in the src attribute of a <script> tag.

<html>
<body>
<h2> demo External JS </h2>

 A paragraph

<button type = "button" onclick = "myfunction c()"> Try it
</button>

 this example links to "myscript.js"
 my function is stored in "myscript.js"
<script src = "myscript.js"> </script>
</body>
</html>

Javascript output

-> JS display possibilities.

-> JS can "display" data in different ways.

-> writing into a HTML element using innerHTML
-> writing into the output using document.write().
-> writing into an alert box, using window.alert().
-> writing into the browsers console, using console.log().

-> using innerHTML

-> To access an HTML element, JavaScript can use the document.getElementById(id) method.

```
-> <html>
   <body>
   <h1> my first web page </h1>

   <p id = "demo"> </p>
   <script>
      document.getElementById("demo").innerHTML = 5+6;
   </script>
   </body>
   </html>
```

-> using document.write()

```
<html>
<body>
<h1> my first web page </h1>
<script>
    document.write(5+6);
</script>
</body> </html>
```

using document.write() after an Html document is loaded. will delete all existing Html.

=> <html>
<body>
<h1> my first web page </h1>
 my first paragraph.

<button type="button" onclick="document.write(5+6)"> Try it </button>

</body>
</html>

=> using window.alert()

You can use an alert box to display data;

<html>
<body>
<h1> my first web page </h1>
<script>
window.alert(5+6);
</script>
</body>
</html>

=> using console.log()

-> for debugging purposes, you can call the console.log() method in the browser to display data. You will learn more about debgging in a later chapter.

→ <html>
<body>
<script>
console.log (5+6);
</script>
</body>
</html>

→ Javascript print.

→ JS doesn't not have any print object or print method.

→ You can't access output devies from Javascript.

→ The only exception is that you can call the window.print() method in the browser to print the the content of the current window.

→ <html>
<body>

<button onclick="window.print()">print this page</button>

</body>
</html>

* Javascript statements

<html>
<body>
<h2> Javascript statements </h2>

{p} A {b} Javascript program {/b} is a list of {b} statements {/b} to be executed by a computer {/p}

{p id = "demo"} {/p}

{script}
let x, y, z;
x = 5;
y = 6;
z = x+y;

document.getElementById("demo").innerHTML =
"The value of z is " + z + "."; {/script}

{/body}
{/html}

★ Javascript Programs

-> A computer program is a list of "instructions" to be "executed" by a computer.

-> In a programming language, these programming instructions are called statements.

-> A Javascript program is a list of programming statements.

Javascript statements.

-> Javascript statement are composed of;

-> values, operators, Expressions, keywords, and comment.

Date
Page

-> This statement tells the browser to write "Hello Dolly" inside an Html element with id = "demo";

<html>
<body>
<h2> JavaScript statments </h2>

 In Html, JS statements are executed by the browser.

<script>
document.getElementById("demo").innerHTML = "Hello Dolly"; </script>

</body>
</html>

semicolons;

-> semicoms separate JS statments

-> Add a semicolon at the end of each executable statement;

<html>
<body>
<h2> JavaScript statement </h2>

 JS statements are separated by semicolons

<script>
let a, b, c;
a = 5;
b = 6;
c = a+b;

document.getElement BY Id ("demo"); inner HTML
</script>
</body>
</html>

JS line length and line Breaks

<html>
<body>
<h2> JS statment </h2>

 the best place to break a code line is after
an operator or a comma.

<script>
document.getElement BY Id ("demo"), inner
HTML = " Hello Dolly !";
</script>
</body>
</html>

JS code Blocks

-> JS statements can be grouped together in code blocks
inside curly brackets {...}.

-> The purpose of code blocks is to define statements
to be executed together.

<html>
<body>
<h2> JavaScript statements </h2>

Date
Page

<P> JS code blocks are written between <and> </P>

<button type="button" onclick="myFunction()">
Click me! </button>

<script>
function myFunction() {
document.getElementById("demo2").innerHTML="
How are you?"; }

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