220403010065U

DATE
PAGE

SYLLABUS

Sr. No	Topic
1	About W3.Org and its web page validator
2	CSS - Cascading Styling sheet : Types of CSS & its preferences
3	Working with CSS selectors
4	Gray layouts and box-model.
5	Video positioning effect
6	Image hover effect
7	Media Query
8	Font Awesome
9	Parallax effect
	CSS - Cascading Stylesheet. Revision
10	Bootstrap : Getting started with responsive designing
11	Bootstrap grid.
12	Bootstrap form
13	Bootstrap readymade code plugs - navbars, modal popup, carousel
	Bootstrap Revision
14	One page web profile designing with wow.js and animate.css
15	Working with Javascript : Getting started with DOM.
16	Variable declaration in JS.
17	Type conversion in JS.
18	Loops in JS.
19	Confiditional statements in JS.
20	Functions in JS (in built & user define)
21	Events and Dynamic DOM manipulation in JS.
22	Getting started with Jquery and its plugin.
23	Getting started with $ and document object.
24	ready ( ) function.
25	function chaining & function callback in jQuery.
26	Event and val parameters in jQuery function.
27	Working with jQuery events.
28	Keypress, blur and click event validation in jQuery with reg ex.
29	jQuery UI Interactions
30	jQuery UI Widgets
	Javascript, jQuery, jQuery UI Revision

DATE
PAGE

2204030100654

Programs

1. Types of css and its preferences.
   - Inline CSS
   - Internal CSS
   - External CSS

(i) Inline CSS

```
<html>
<head>
    <title> CSS types </title>
</head>
<body>
<div style="background-color: red; height: 100px; width: 100px; color: white; font-weight: bold; font-size: 15px; text-align: center; justify-content: center; display: flex; padding: 15px;">
    Inline Stylesheet
</div> <hr/>

<div style="background-color: cornflowerblue; height: 100px; width: 100px; color: black; font-weight: bold; font-size: 15px; text-align: center; align-items: center; justify-content: center; display: flex; padding: 15px;"> Inline Stylesheet
</div> <hr/>

<div style="background-color: cyan; height: 100px; width: 100px; color: black; font-weight: bold; font-size: 15px; text-align: center; align-items: flex-end; justify-content: center; display: flex; padding: 15px;"> Inline Stylesheet
</div> <hr/>
</body>
</html>
```

220403010065 4

(ii) Internal CSS

```
<html>
<head>
<title> CSS types </title>
<style>
.divCls {
background-color : red; height : 100px; width: 100px;
color : white; font-weight : bold; font-size : 15px;
align-items : top; justify-content : center;
display : flex; padding : 15px; text-align : center;
}
.divCls1 {
background-color : cornflower; height : 100px; weight :
100px; color : black; font-weight : bold; font-size :
15px; text-align : center; padding : 15px;
}
.divCls2 {
background-color : cyan; height : 100px; weight : 100px
color : black; font-weight : bold; font-size : 15px;
padding : 15px; text-align : center; align-items :
flex-end; justify-content : center; display : flex;
}
</style>
<body>
<div class = "divCls"> Internal Stylesheet
</div> <hr/>

<div class = "divCls1"> Internal stylesheet
</div> <hr/>

<div class = "divCls2"> Internal stylesheet
</div> <hr/>
</body>
</html>
```

DATE
PAGE

2204030100654

(iii) External CSS

```
<html>
 <head>
  <title> CSS types </title>
  <link rel="stylesheet" href="external.css">
 </head>
 <body>
 <div id="divCls"> External Stylesheet
 </div> <hr/>

 <div id="divCls1"> External Stylesheet
 </div> <hr/>

 <div id="divCls2"> External Stylesheet
 </div> <hr/>
 </body> </html>
```

External.css -

```
#divCls {
 background-color: red; height: 100px; width: 100px;
 color: white; font-weight: bold; font-size: 15px;
 align-items: top; justify-content: center;
 display: flex; padding: 15px; text-align: center;
 }

#divCls1 {
 background-color: cornflowerblue; height: 100px;
 width: 100px; color: black; font-weight: bold;
 font-size: 15px; padding: 15px; text-align: center;
 align-items: center; justify-content: center;
 display: flex; }

#divCls2 {
 background-color: cyan; height: 100px; width: 100px;
 color: black; font-weight: bold; font-size: 15px;
 padding: 15px; align-items: flex-end; text-align:
 center; display: flex; justify-content: center;
 }
```

2. CSS selectors - Universal selector
Id selector
Class selector
Grouping selector
Element selector

(i) Universal selector (*)
It selects all HTML elements on the page.

```
<html>
 <head>
  <title> CSS selectors </title>
  <style>
   * {
    text-align: center;
    color: blue;
    background-color: bisque;
   }
  </style>
  </head>
  <body>
  <h1> Hello Everyone ! </h1>
  <p> Every element on the page will be
     affected by style. </p>
  </body>
  </html>
```

(ii) Id Selector (#)
It selects id attribute of an HTML element.

```
<html>
 <head>
  <title> CSS selectors </title>
 <style>
 # para {
    text-align: center;
    color: red;
 }
```

```
</style>
</head>
<body>
<p id="para"> Hello Everyone </p>
<p> This paragraph is not affected by the
 style. </p>
</body>
</html>
```

(iii) Class selector (.)

It selects HTML elements with a specific class attribute

```
<html>
<head>
<title> CSS selectors </title>
<style>
.center {
 text-align: center;
 color: pink;
}
</style>
</head>
<body>
<h1 class="center"> Red and center-aligned
 heading </h1>
<p class="center"> Red and center-aligned heading
 </h1>
</body>
</html>
```

(iv) Grouping selector (h1, p, h3)

It selects all the HTML elements with the same style definitions.

```
<html>
<head>
<title> CSS selectors </title>
```

2204030100654

```
<style>
h1, h2, p {
  text-align: center;
  color: orange;
}
</style>
</head>
<body>
<h1> Hello Everyone </h1>
<h2> Smaller Heading! </h2>
<p> This is a paragraph. <p>
</body>
</html>
```

(v) Element selector (p), (h1), (h2)

It selects HTML elements based on element name.

```
<html>
<head>
<title> CSS selectors </title>
<style>
p {
  text-align: center;
  color: red;
  font-weight: bold;
}
</style>
</head>
<body>
<p> Every paragraph of the body will be
    affected by the style </p>
<p id="para1"> This is another paragraph </p>
</body>
</html>
```

3. Gray Layouts -

GRAY LAYOUT - 1

```
<!DOCTYPE html>
<html>
 <head>
    <title> Gray layout -1 </title>
    <style>
      * { font-size : 20px ; font-weight : bold; }
    #R1 {
    background-color : antiquewhite ; width : auto; height : 50p
    padding : 5px; align-items : center; justify-content :
    center; margin-bottom : 10px; padding-left : 2%;
    padding-top : 30px; }
    #R2, #R5 {
    background-color : darkgray ; width : auto; height : 50px;
    padding : 5px; align-items : center ; text-align :
    center; display : flex ; justify-content : center ;
    margin-bottom : 10px; }
    #R3 {
    background-color : darkseagreen; width : auto; padding :
    5px; height : 60px; align-items : center; text-
    align : center; display : flex; justify-content :
    center; margin-bottom : 20 px; }
    #R4 {
    height : 500px; width : auto ; margin-bottom : 20px; }
    #R4C1 {
     width : 29% ; margin-right : 1% ;
     background-color : cadetblue ; margin-bottom : 10px; }
    #R4C2 {
     width : 70% ; background-color : darkturquoise ;
     margin-bottom : 10 px ; }
    #R4 div {
     float : left ; height : 300 px ; padding-top : 280px ;
     text-align : center ; }
    #R5 { margin-top : 40px ; }
```

```
</style>
</head>
<body>
<div>
<div id="R1"> Logo </div>
<div id="R2"> Navigation </div>
<div id="R3"> Header / Banner </div>
<div id="R4">
<div id="R4C1"> Side Bar </div>
<div id="R4C2"> Body Area </div>
</div>
<div id="R5"> Footer </div>
</div>
</body>
</html>
```

GRAY LAYOUT - 2

```
<!DOCTYPE html>
<html>
<head>
<title> Gray layouts-2 </title>
<style>
*{ font-size: 22px; font-weight: bold; }

#R1{
background-color: antiquewhite; height: 50px; padding-top: 30px; width: 95%; padding-left: 4%;
margin-bottom: 10px; display: flex; }

#R2{
background-color: gray; height: 33px; width: 100%;
padding-top: 7px; text-align: center; margin-bottom: 10px; }
```

```
#R3{
background-color: darkseagreen; height: 55px; width: 100%;
padding-top: 7px; text-align: center; margin-bottom:
10px; align-items: center; justify-content: center;
display: flex; }

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

#R4C1{
width: 24%; margin-right: 1%;
background-color: aquamarine; }

#R4C2{
width: 23%; margin-right: 1%; margin-left: 1%;
background-color: lightgreen; }

#R4C3{
width: 23%; margin-right: 1%; margin-left: 1%;
background-color: rgb(116, 210, 177); }

#R4C4{
width: 24%; margin-left: 1%;
background-color: rgb(108, 162, 185); }

#R4 div{
float: left; height: 320 px; padding-top: 280 px;
text-align: center; }

#R5{
height: 10px; width: 100%; margin-bottom: 10px;}

#R5C1{
width: 49%; margin-right: 1%;
background-color: violet; }
```

```
#R5C2 {
width: 49%; margin-left: 1%;
background-color: blueviolet; }

#R5 div {
float: left; height: 40px; padding-top: 10px;
text-align: center; margin-bottom: 10px; }
</style>
</head>
<body>
<div>
<div id="R1"> Logo </div>
<div id="R2"> Navigation </div>
<div id="R3"> Header/Banner </div>
<div id="R4">
<div id="R4C1"> Side Bar </div>
<div id="R4C2"> Body Area </div>
<div id="R4C3"> Body Text </div>
<div id="R4C4"> Body Text </div>
</div>
<div id="R5">
<div id="R5C1"> Side Bar </div>
<div id="R5C2"> Body Area </div>
</div>
</div>
</body>
</html>
```

220403010065 4

The 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: margins, borders, padding and the actual content.

Explanation of different parts:

- Content: The content of the box, where text and images appear.
- Padding: Clears an area around the content. The padding is transparent.
- Border: A border that goes around the padding & content.
- Margin: Clears an area outside the border. The margin is transparent.

2204030100654

4. IMAGE HOVER EFFECT

HTML →

```
<!DOCTYPE html>
<html>
    <head>
    <title> Image Hover Text effect with HTML & CSS </title>
    <link rel = "stylesheet" href = "image.css">
    </head>
   <body>
    <div class="container">
    <img src = "girl.jpeg" alt = "Avatar" class = "image 1">
    <div class = "middle">
    <div class = "text"> Hello Everyone </div>
    </div>
    </div>
   </body>
   </html>
```

CSS →

```
. container {
position: relative; width : 50%; }

. image 1 {
opacity: 1; display : flex; width : 100%; height : 100%;
transition : .5s ease; backface-visibility : hidden; }

. middle {
transition : .5s ease; opacity : 0; position : absolute;
top : 50%; left : 50%; transform : translate (-50%, -50%);
-ms-transform : translate (-50%, -50%); text-align : center; }

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

. container : hover . image 1 { opacity : 0.3; }

. text { background-color : #04AA6D; color : white;
font-size : 16px; padding : 16px 32px; }
```

220403010065 54

5. FONT AWESOME

```
<!DOCTYPE html>
    <html>
     <head>
      <title> font Awesome </title>
      <link rel = "stylesheet" href = " https : // cdnjs .
       cloudflare . com / ajax / libs / font - awesome / 4.7.0 /
       css / font - awesome . min . css ">
     <style>
      ul li {
     color : darkviolet ; font - size : 30px ; list - style :
      none ; }

       ul li i { color : cadetblue ; }
     </style>
     </head>
    <body>
      <ul>
      <li><i class = "fa fa-coffee"> </i> </li> < hr>
      <li><i class = "fa fa-child"> </i> </li> <hr>
      <li><i class = "fa fa-database"> </i> </li> <hr>
      <li><i class = "fa fa-paper-plane"> </i> </li> <hr>
      <li><i class = "fa fa-paint-brush"> </i> </li> <hr>
      <li><i class = "fa fa-envelope-o"> </i> </li> <hr>
       <li><i class = "fa fa-play-circle-o"> </i> </li> <hr>
       <li><i class = "fa fa-github"> </i> </li> <hr>
      </ul>
    </body>
    </html>
```

220403010065 4

6. PARALLAX EFFECT

HTML →

```
<!DOCTYPE html>
<html>
<head>
<title> Parallax Effect </title>
<link rel = "stylesheet" type = "text/css href = "parallax.css">
</head>
<body>
<div class = "box1" > </div>
<div class = "box2" > </div>
<div class = "box3" > </div>
</body>
</html>
```

CSS →

```
body {
margin : 0; padding : 0; }

.box1 {
width : 100 %; height : 100vh; background-image : url ( );
background-size : cover; background-position : center;
background-attachment : fixed; }

.box2 {
width : 100%; height : 100vh; background-image : url ( );
background-size : cover; background-position : center;
background-attachment : fixed; }

.box3 {
width : 100%; height : 100vh; background-image : url ( );
background-size : cover; background-position : center;
background-attachment : fixed; }
```

DATE
PAGE

220403010065S

7. 2D TRANSFORMS / 3D TRANSFORMS

HTML → (i) Translate

```
<!DOCTYPE html>
<html>
<head>
<title> Animation </title>
<link rel="stylesheet" href="animation.css">
</head>
<body>
<div class="translate"> Translate </div>
<hr>
<div class="rotate"> Rotate </div>
<hr>
<div class="text-shadow"> Text-shadow </div>
<hr>
</body>
</html>
```

CSS →

```
.translate {
text-align: center; font-size: 20px; color: burlywood;
background-color: black; border: 1px; width: 120 px;
height: 120px; transform: translate (30px) ; }

.translate:hover {
transition: 2s; color: black; background-color: gray;
Transform: translate (100px, 100px) ; }

.rotate {
text-align: center; font-size: 20px; color: crimson;
background-color: burlywood; border: 1px; width: 120px;
height: 120px; }

.rotate:hover {
transition: 2s; color: black; background-color: gray;
```

transform: rotate(180deg); }

.text-shadow {
text-align: center; font-size: 35px; color: black;
margin: 10px; }

.text-shadow:hover {
text-align: left; font-size: 35px; transition: 2s;
text-shadow: 0 0 5px blue, 2px 5px 5px pink,
5px 10px 5px violet; }

* ANIMATION with KEYFRAMES.

```
<!DOCTYPE html>
<html>
<head>
<title> Animation-1 </title>
<style>
.div {
width: 100px; height: 100px; background-color: cadetblue;
position: relative; animation-name: example;
animation-duration: 15s; }

@keyframes example {
0% { background-color: orange; transform: translate(50px,
100px); box-shadow: 2px 2px 5px green, 2px 2px 5px
yellow; left: 0px; top: 0px; }

25% { background-color: yellow; transform: rotate(150deg);
border-radius: 10px; left: 300px; top: 250px; }

50% { background-color: blue; transform: rotate(360deg);
left: 400px; top: 400px; }
```

```
75% { background-color: green; border-radius: 20px;
      left: 500px; top: 200px; }

100% { background-color: cornflowerblue; left: 700px;
       top: 0px; }

</style>
</head>
<body>
<h1> CSS Animation </h1>
<div> </div>
</body>
</html>
```

8. VIDEO POSITIONING EFFECT

HTML →

```
<!DOCTYPE html>
<html>
<head>
<title> Video Positioning Effect </title>
<link rel="stylesheet" type="text/css" href="video.css">
</head>
<body>
<div class="container">
<video autoplay muted loop display>
<source src="video.mp4" type="video/mp4">
</video>
<div class="text-box">
<h1> SOCCA </h1>
</div>
</div>
</body>
</html>
```

DATE
PAGE

220403010065y

CSS →

```
* {
    margin : 0 ;
    padding : 0;
    font- family : 'Segoe UI', Tahoma, Geneva, Verdana, sans-
    serif ; box - sizing : border - box ; }

. container {
    width : 100% ; height : 100vh; position : relative ; }

. container video {
    width : 100% ; height : 100% ; }

. text - box {
    position : absolute ; top : 0 ; left : 0 ; width : 100% ;
    height : 100% ; color : transparent ; display : flex ;
    align - items : center ; justify - content : center ;
    font - size : 40px ; transition : 0.5 s ; }

. text - box : hover {
    -webkit - text - stroke : 2px #fff ;
    color : darkviolet ;
}
```

MRI
30-01-2023.

Good Girl.

220403010065 4

DATE
PAGE

BOOTSTRAP

```
<!DOCTYPE html>
<html lang = "en" >
<head>
<title> Bootstrap Example </title>
<meta charset = "utf-8" >
<meta name = "viewport" content = " width - device - width
  initial - scale = 1 " >
<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>
<style>
.row {
    margin - bottom : 10px ;
}
</style>
</head>
<body>
<div class = " container fluid mt-3">
<h1> Responsive Columns </h1>
<div class = " Row " >
<div class = "col - sm - 1 p-3 bg - primary text - white"> . col </div>
<div class = " col - sm - 1 p-3 bg - dark text - white " > . col </div>
<div class = " col - sm - 1 p-3 bg - primary text - white"> . col </div>
<div class = " col - sm - 1 p-3 bg - dark text - white"> . col </div>
<div class = " col - sm - 1 p-3 bg - primary text - white"> . col </div>
<div class = " col - sm - 1 p-3 bg - dark text - white"> . col </div>
<div class = " col - sm - 1 p-3 bg - primary text - white">. col </div>
<div class = " col - sm - 1 p-3 bg - dark text - white" > . col </div>
<div class = " col - sm - 1 p-3 bg - primary text - white " > . col </div>
<div class = " col - sm - 1 p-3 bg - dark text - white " > . col </div>
<div class = " col - sm - 1 p-3 bg - primary text - white"> . col </div>
<div class = " col - sm - 1 p-3 bg - dark text - white"> . col </div>
</div>
```

220403010065Y

```
<div class = "row" >
<div class = "col - sm-4 p-3 bg-primary text-white"> .col </div
<div class = "col - sm-4 p-3 bg-dark text -white"> .col </div
<div class = "col - sm-4 p-3 bg-primary text -white"> .col </div
</div>
<div class = "row" >
<div class = "col - sm-4 p-3 bg- primary text- white"> .col </div
<div class = "col - sm-8 p-3 bg-dark text -white" > .col </div
</div>
<div class = "row" >
<div class = "col - sm-6 p-3 bg -primary text -white"> .col </div
<div class = "col - sm-6 p-3 bg -primary text -white"> .col </div
</div>
<div class = "row" >
<div class = "col- sm-12 p-3 bg-primary text -white"> .col
</div>
</div>
</div>
</body>
</html>
```

220403010065 4

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Example </title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width"
initial-scale=1">
<link rel="stylesheet" href="https://cdn.jsdelivr.net
/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@
3.6.3/dist/jquery.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@
1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap
@4.6.2/dist/JS/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class="container">
  <h2> Stacked form </h2>
  <form action="/action_page.php">
  <div class="form-group">
  <label for="email">Email:</label>
  <input type="email" class="form-control" id="email"
    placeholder="Enter email" name="email">
  </div>
  <div class="form-group">
  <label for="pwd">password:</label>
  <input type="pwd" class="form-control" id="pass-
    word" placeholder="Enter password" name="pwd">
  </div>
  <div class="form-group-form-check">
  <label class="form-check-label">
  <input class="form-check-inpwt" type="checkbox"
  name="remember">
```

DATE
PAGE

2204030100654

```
remember me
</label>
</div>
<button type="submit" class="btn btn-primary">
  Submit </button>
</form>
</div>
</body>
</html>
```

220403010065S4

* Javascript can change HTML content

→ One of many javascript HTML method is getElement By Id ( ).

The example below "finds" an HTML element (with id = "demo"), and changes the element content to "Hello Javascript":

```
<html>
<body>
<h2> What can Javascript Do? </h2>
<p id = "demo"> Javascript can change HTML content.
</p>
<button type = " button" onclick = " document. getElement
By Id ("demo") .inner HTML = " 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 Javascript Do? </h2>
<p> Javascript can change HTML attribute values. </p>
<p> In this case Javascript changes the values
  of the src attribute of an image. </p>
<button onclick = " document. get Element By Id ('Image')
. src = ' pic. bulbon. gif "> Turn on the Light </button>

<img id = " Image" src = " pic. bulboff. gif"
```

DATE
PAGE

style = "width: 100px" >

<button onclick = " document . get Element By Id ('Image' .src = 'pic. bulboff. gif' " > Turn off the light
</button >
</body >
</html >

* Javascript can change HTML styles (css)

→ Changing the style of an HTML element, is a variant of changing an HTML attribute :

<html >
<body >
<h2 > What can Javascript Do? </h2 >
 Javascript can change the style of an HTML element.

<button type = " button" onclick = " document . get Element By Id ('demo') . style font size = '35 px' " >
Click Me!
</button >
</body >
</html >

* HTML Elements

→ Hiding HTML elements can be done by changing the display style :

< html >
< body >
<h2 > What can Javascript Do? </h2 >
 id = "demo" > Javascript can hide HTML elements

DATE
PAGE

```
</p>
<button type = "button" onclick = "document.getElement
By Id ('demo') . style display = 'none'" > Click Me!
</button>
</body>
</html>
```

* Show HTML Elements

→ Showing hidden HTML elements can also be done by changing the display style:

```
<html>
<body>
<h2> What can Javascript Do? </h2>
<p> Javascript can show hidden HTML elements . </p>
<p id = "demo" style = " display : none " > Hello
Javascript ! </p>
<button type = "button" onclick = "document.getElement
By Id ('demo') . style . display = 'block'" > Click Me!
</button>
</body>
</html>
```

Old Javascript examples may use a type attribute:

```
<script type = "text / Javascript" >
```

The type attribute is not required. Javascript is the default scripting language in HTML.

DATE
PAGE

220403010065Y

* Javascript functions and Events

→ A Javascript function is a block of javascript code, that can be executed when "called" for.

For example,
A function can be called when an event occurs, like when the user clicks a button.

* Javascript in <head> or <body>

→ You can place any number of scripts in an HTML document.

Scripts can be placed in the <body> or in the <head> section of an HTML page, or in both.

* Javascript in <head>

In this example, a javascript function is placed in the <head> section of an HTML page.

The function is invoked when a button is clicked.

```
<! DOCTYPE html>
<html>
<head>
<script>
function myfunction() {
  document.getElementById("demo").innerHTML =
"paragraph changed.";
}
</script>
</head>
```

```
<body>
<h2> Demo Javascript in Head </h2>
<p id = "demo"> A paragraph </p>
<button type = "button" onclick = "myfunction()">
 Try it </button>
</body>
</html>
```

* Javascript in <body>

In this example, a Javascript function is placed in the <body> section of an HTML page.

The function is invoked when a button is clicked:

```
<!DOCTYPE html>
<html>
<body>
<h2> Demo Javascript in Body </h2>
<p id = "demo"> A paragraph </p>
<button type = "button" onclick = "myfunction()">
 Try it </button>
<script>
function myfunction() {
 document.getElementById("demo").innerHTML =
"paragraph changed";
}
</script>
</body>
</html>
```

2204030100654

* External Javascript

Scripts can also be placed in external files:

```
Function myfunction() {
document.getElementById("demo").innerHTML
="paragraph changed.";
}
```

External scripts are practical when the same code is used in many different web pages.

Javascript file 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:

Example:

```
<html>
<body>
<h2> Demo External Javascript </h2>
<p id="demo"> A paragraph </p>
<button type="button" onclick="myfunction()">
Try it </button>
<p> This example links to "myscript.js". </p>
<p> (myfunction is stored in "myscript.js") </p
<script src="myscript.js"> </script>
</body>
</html>
```

You can place an external script reference in <head> or <body> as you like.

The script will behave as if it was located exactly

where the <script> tag is located.

External scripts cannot contain <script> tags.

* Javascript display possibilities.

- Writing into an HTML element, using innerHTML.
- Writing into the HTML output using document.write()
- Writing into an alert box, using window.alert()
- Writing into an browser console, using console.log().

* Using innerHTML.

```
<html>
<body>
<h1> My first web page </h1>
<p> My first paragraph </p>

<p id = "demo" > </p>

< script > document.getElement By Id ("demo")
.innerHTML = 5+6 ;
</ script >
</ body >
</ html >
```

* Using document.write()

```
<html >
<body >
<h1> My first webpage </h1>
```

```
<p> My first paragraph </p>
<script> document.write (5+6);
</script>
</body>
</html>
```

* Delete all existing HTML

```
<!DOCTYPE html>
<html>
<body>
<h1> My first webpage </h1>
<p> My first paragraph </p>
<button type = "button" onclick = "document.getElement By Id write (5+6) "> Try it
</button>
</body>
</html>
```

* Using window.alert ()

```
<html>
<body>
<h1> My first web page </h1>
<p> My first paragraph </p>
<script> window.alert (5+6); </script>
</body>
</html>
```

You can skip the window Keyword.

In Javascript, the window object is the global scope object. This means that variables, properties and methods by default belong to the window

DATE
PAGE

object.

```
<!DOCTYPE html>
<html>
<body>
<h1> My first webpage </h1>
<p> My first paragraph </p>
<script> alert (5+6); </script>
</body>
</html>
```

* Using console.log()

```
<!DOCTYPE html>
<html>
<body>
<script> console.log (5+6); </script>
</body>
</html>
```

=> Javascript print

Javascript does not have any print object or print methods.

You cannot access output devices from javascript.

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

```
<!DOCTYPE html>
<html>
<body>
<button > onclick = " window.print() ">
```

print this page </button>
</body>
</html>

* Javascript statements

Values, Operators, Expressions, Keywords and comments.

```
<html>
<body>
<h2> Javascript statements </h2>
<p> In HTML, Javascript statements are
    executed by the browser. </p>
<p id = "demo" > </p>
<script>
 document.getElementById("demo").innerHTML
 = "Hello Everyone";
</script>
</body>
</html>
```

* Variables

In a programming language, variables are used to store data values.

Javascript uses the keywords var, let and const to declare variables.

An equal sign is used to assign values to variables.

```
<html>
<body>
<h2> Javascript variables </h2>
<p> In this example, x is defined as a variable.
Then, x is assigned the value of 6: </p>
<p id = "demo" > </p>
<script >
let x ;
x = 6 ;
document. get Element By Id ("demo"). inner HTML = x;
</script >
</body>
</html >
```

* Single Line comments

```
<html >
<head>
<body >
<h1 id = "myH" > </h1 >
<p id = "myP"> </p >
<script >
// change heading :
document. get Element By Id ("myH"). inner HTML =
" Javascript comments" ;
// change paragraph :
document. get Element By Id ("myP"). inner HTML =
" My first paragraph" ;
</ script >
</body >
</html >
```

* Multi Line comments

```
<html>
<body>
<h2> Javascript comment </h2>
<h1 id = "myH" > </h1>
<p id = "myP" > </p>
<script>
// document. getElement By Id ("myH"). inner HTML
= " My first page" ;
document. get Element By Id ("myp"). inner HTML =
" My first paragraph" ;
</script>
<p> The line starting with // is not executed
</p>
</body>
</html>
```

* Javascript Variables.

- 4 ways to declare a javascript variable.

(i) Using var
(ii) using let
(iii) using const
(iv) using nothing

(i)
```
<! DOCTYPE html >
<html>
<body>
<h1> Javascript variables </h1>
<p> In this example, x, y and z are
variables. </p>
```

```
<p id = "demo" > </p>
<script >
var x = 5 ;
var y = 6 ;
var z = x+y ;
document. get Element By Id ("demo"). inner HTML =
" The value of z is : " + z ;
</script >
</body >
</html >
```

(ii)

```
<! DOCTYPE html>
<html>
< body >
<h1 > Javascript Variables </h1 >
<p> In this example x, y and z are variables. </p>
<p id = "demo" > </p>
<script >
let x = 5 ;
let x = 6 ;
let z = x+y ;
document. get Element By Id ("demo") . inner HTML =
" The value of z is : " + z ;
</ script >
</ body >
</html >
```

(iii)

```
<! DOCTYPE html >
<html >
< body >
<h1 > Javascript Variables </h1 >
<p> In this example , price 1 , price 2 and total are
    variables . </p>
<p id = "demo" > </p>
```

```
<script>
const price1 = 5;
const price2 = 6;
let total = price1 + price2;
document.getElement By Id ("demo") .innerHTML =
"The total is :" + total;
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h1> Javascript Variables </h1>
<p> In this example, x, y and z are
    undeclared variables. </p>
<p id="demo"> </p>
<script>
x = 5;
y = 6;
z = x + y;
document.getElement By Id ("demo").innerHTML =
"The value of z is : " + z;
</script>
</body>
</html>
```

* Types of Javascript Operators

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Logical Operators
- Conditional Operators
- Type Operators

DATE
PAGE

```
<html>
<body>
<p id="demo"></p>
<script>
let a = 3;
let x = (100+50) * a;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h1> Javascript Variables </h1>
<p id="demo"> </p>
<script>
let text1 = "What a very";
text1 += " nice day";
document.getElementById("demo").innerHTML = text1;
</script>
</body>
</html>
```

* Adding Strings ~~to~~ and numbers

```
<html>
<body>
<h1> Javascript Operators </h1>
<p id="demo"> </p>
<script>
let x = 5+5;
let y = "5" + 5;
let z = "Hello" + 5;
document.getElementById("demo").innerHTML =
```

```
x + "<br>" + y + "<br>" + z ;
</script>
</body>
</html>
```

* Loops

```
<!DOCTYPE html>
<html>
<body>
<h2> Javascript for Loop </h2>
<p id="demo"> </p>
<script>
const cars = ["BMW", "Volvo", "Saab", "Ford",
              "Audi"];
let text = "";
for (let i=0 ; i< cars.length ; i++) {
    text += cars[i] + "<br>" ;
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

* For loop

```
<html>
<body>
<h2> Javascript for loop </h2>
<p id="demo"> </p>
<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i, len, text;
```

```
for ( i=0 ; len = cars. length , text = " " ; i<len; i++){
 text += cars [i] + "<br>";
 }
document. get Element By Id ("demo" ). inner HTML = text ;
</ script >
</ body >
</ html >
```

* The for in loop

```
<! DOCTYPE html>
< html >
< body >
< h2 > Javascript for In loop </ h2 >
< p id = "demo" > </ p >
< script >
const person = { fname : "John" , lname : "Doe", age : 25 }
 let text = " ";
 for ( let x in person ) {
 text += person [x] ;
 }
document. get Element By Id ("demo") . inner HTML = text ;
 </ script >
</ body >
 </ html >
```

→ The for in loop iterates over a person object.

→ Each iteration returns a Key (x)

→ The key is used to access the value of the key.

→ The value of key is person [x].

220403010065 4

* For In Over Arrays

```
<! DOCTYPE html>
<html>
<body>
<h2> Javascript for In </h2>
<p id = "demo" > </p>
<script>
const numbers = [45, 4, 9, 16, 25];
let text = " ";
for (let x in numbers) {
 text += numbers [x] + "<br>";
}
document. get Element By Id ("demo"). inner HTML =
 text;
</script>
</body>
</html>
```

* Array. for Each ()

```
<! DOCTYPE html>
<html>
<body>
<h2> Javascript Array. for Each () </h2>
<p id = "demo" > </p>
<script>
const numbers = [45, 4, 9, 16, 25];
let text = " ";
 numbers. for Each (myfunction);
document. get Element By Id ("demo"). inner HTML =
 text;
function myfunction (value, index, array) {
 text += value + "<br>";
}
```

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

* The for of loop

```
<html>
<body>
<h2> Javascript for of Loop </h2>
<p id="demo"> </p>
<script>
const cars = ["BMW", "Volvo", "Mini"];
let text = "";
for (let x of cars) {
  text += cars x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

* Looping Over an Array

```
<html>
<body>
<h2> Javascript for of Loop </h2>
<p id="demo"> </p>
<script>
const cars = ["BMW", "Volvo", "Mini"];
let text = "";
for (let x of cars) {
text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
```

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

* The while loop

```
<html>
<body>
<h2> Javascript while loop </h2>
<p id="demo"> </p>
<script>
 let text = " ";
 let i = 0;
 while (i < 10) {
   text += "<br> The number is " + i;
   i++;
 }
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

* The Do While loop

```
<html>
<body>
<h2> Javascript Do While Loop </h2>
<p id="demo"> </p>
<script>
 let text = " ";
 let i = 0;
 do {
  text += "<br> The number is" + i;
  i++;
 }
while (i<10);
```

```
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

* Comparing for and While

```
<html>
<body>
<p id = "demo"> </p>
<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i = 0;
let text = "";
for (; cars[i];) {
  text += cars[i] + "<br>";
  i++;
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

* TYPE Conversion

Converting Strings to Numbers
Converting Numbers to Strings
Converting Dates to Numbers
Converting Numbers to Dates
Converting Booleans to Numbers
Converting Numbers to Booleans.

2204030100654

* Converting Strings to Numbers

```
<html>
<body>
<h1> Numbers </h1>
<h2> The Number ( ) Method </h2>
<p id= "demo" > </p>
< script >
document. get Element By Id ("demo") .inner HTML =
Number ("3.14") + "<br>" +
Number (Math. PI) + "<br>" +
Number ("  ") + "<br>" +
Number ("") + "<br>" +
Number ("99  88") + "<br>" +
Number ("John") + "<br>" ;
</script>
</body>
</html>
```

* Converting Numbers to Strings

```
<htmL>
<body>
<h2> Javascript String ( ) Method </h2>
<p id= "demo" > </p>
< script>
let x = 123;
document. get Element By Id ("demo") . inner HTML =
string (x) + "<br>" +
string (123) + "<br>" +
string (100 + 23);
</script>
</body>
</html>
```

DATE
PAGE

* tostring

```
<html>
<body>
<h2> Number Method </h2>
<p id = "demo" > </p>
<script>
 let x = 123;
document.getElement By Id ("demo").innerHTML =
 x.tostring ( ) + "<br>" +
(123).tostring ( ) + "<br>" +
(100+23).tostring ( );
 </script>
 </body>
 </html>
```

* Converting Dates to Strings

```
<html>
<body>
<h2> Javascript Dates </h2>
<p id = "demo" > </p>
<script>
 let x = 25/05/2023;
document.getElement By Id ("demo").innerHTML =
String (Date) + "<br>" +
string (x) + "<br>" +
String (100+35);
</script>
</body>
</html>
```

DATE
PAGE

* Automatic Type Conversion

```
<html>
<body>
<h2> Automatic Type Conversion </h2>
<p id = "demo" > </p>
<script>
document.getElement By Id ("demo"). innerHTML =
(5 + null) + "<br>" +
("5" + null) + "<br>" +
("5" + 2) + "<br>" +
("5" - 2) + "<br>" +
("5" * 2) + "<br>" +
("5" / 2) + "<br>"
</script>
</body>
</html>
```

* Conditional Statements [Break, Continue]

(i) Break : It "jumps out" of a loop.

```
<! DOCTYPE html>
<html>
<body>
<h2> Javascript loops </h2>
<p id = "demo" > </p>
<script>
let text = " ";
for (let i = 0; i < 10; i++) {
if (i === 3) { break; }
text += "The number is" + i + "<br>";
}
```

```
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

(ii) Continue — It "jumps over" one iteration in a loop.

```
<!DOCTYPE html>
<html>
<body>
<h2> Javascript loops </h2>
<p id = "demo"> </p>
<script>
let text = "";
for (let i = 0; i < 10; i++) {
  if (i===3) { continue; }
  text += "The number is" + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

* Javascript functions

- A Javascript function is a block of code designed to perform a particular task.
- It is executed when something 'calls it'.

```
<!DOCTYPE html>
<html>
<body>
<h2> Javascript functions </h2>
<p> This example calls a function which perforems a
    calculation, and returns the result: </p>
```

```
<p id = "demo" > </p>
<script>
function myfunction (p1, p2) {
  return p1 * p2;
}
document.getElementById("demo").innerHTML = myfunction
(4, 3);
</script>
</body>
</html>
```

→ Function Return

- When Javascript reaches a return statement, the function will stop executing.

```
<!DOCTYPE html>
<html>
<body>
<h3> Javascript functions </h3>
<p id = "demo" > </p>
<script>
var x = myfunction (4, 3);
document.getElementById("demo").innerHTML = x;

function myfunction (a, b) {
return a * b;
}
</script>
</body>
</html>
```

DATE
PAGE

```
<! DOCTYPE html >
<html >
<body >
<p > This example calls a function to convert from
   Fahrenheit to Celsius : </p>
<p id = "demo" > </p>
<script >
function to Celsius (f) {
return (5/9) * (f-32);
}
document. get Element By Id ("demo"). inner HTML =
   to Celsiuss (77) ;
</ script >
<hr >

< p id = "demo1" > </p>
< script >
document. get Element By Id ("demo1" ). inner HTML =
" The temperature is" + to Celsius (77) + " Celsius";

function to Celsius (fahrenheit) {
return (5/9) * (fahrenheit - 32);
}
</ script >
</ body >
</ html >
```

→ function used as variable names

→ Local variables

- It can only be accessed from within the function.
- Example -

```
<!DOCTYPE html>
<html>
<body>
<p> Outside myfunction() fruitname is defined </p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
myfunction();

function myfunction() {
let fruitname = "Pineapple";
document.getElement By Id("demo1").innerHTML =
typeof fruitname + " " + fruitname;
}

document.getElement By Id("demo2").innerHTML =
typeof fruitname;
</script>
</body>
</html>
```

* DOM Manipulation in JS.

```
<!DOCTYPE html>
<html>
<body>
<h3> Javascript HTML Events </h3>
<h4 onclick="this.innerHTML='Hello User...'">
Click on this text! </h4>
<hr>

<h3> Javascript HTML Events </h3>
<p> Click the button to display date and time. </p>
<button onclick="displayDate()"> The Cuurent
time is ? </button>
```

DATE
PAGE

```
<script>
function displayDate() {
document.getElementById("demo").innerHTML = Date();
}
</script>

<p id="demo"></p>
<p> Click "Try it" to execute the displayDate()
function. </p>

<button id="myBtn"> Try it </button>

<script>
document.getElementById("demo").onclick =
displayDate;

function displayDate() {
document.getElementById("demo").innerHTML = Date();
}
</script>
<hr>

<h3> Javascript HTML Events </h3>
Enter your name: <input type="text" id="fname"
onchange="upperCase()">
<p> When you leave the input field, a function
is triggered which transforms the input text to
upper case. </p>

<script>
function upperCase() {
const x = document.getElementById("fname");
x.value = x.value.toUpperCase();
}
</script>
```

DATE
PAGE

```
<hr>

<div onmouseover = "mOver(this)" onmouseout = "mOut
                                              (this)"
style = " background-color: #75cf9a; width: 120px;
height: 15px; padding: 40px;" >
Mouse Over Me </div>

<script>
function mOver (obj) {
obj. style . backgroundColor = " gray" ;
obj. innerHTML = " Thank You" ;
}

function mOut (obj) {
obj. innerHTML = " Mouse Over Me"
}
</script>
<hr>

<div onmousedown = " mDown (this)" onmouseup = " mUp
(this)"
 style = background-color: #e967 9d; width : 90px;
height : 20px; padding: 40px;" >
Click Me </div>

 <script>
function mDown (obj) {
obj. style . background Color = " #1ec5e5" ;
obj. innerHTML = "Release Me" ;
}
function mUp (obj) {
obj. style . background Color = " # powderblue" ;
obj. innerHTML = " Thank You" ;
}
</script>
</body>
</html>
```

DATE
PAGE

JQUERY

- Jquery is a Javascript library.
- Jquery greatly simplifies javascript programming.
- Jquery is easy to learn.

* Jquery CDN

- CDN (Content Delivery Network / Content Distribution Network)
- It works together to provide fast delivery of Internet content.

```
<! DOCTYPE html>
<html>
<body> <head>
<script src = " https://ajax.googleapis.com/ajax/libs/jquery/
 3.6.3 /jquery.min.js "> </script>
<script>
$ document. ready (function () {
  $("button"). click (function () {
   $("p"). hide ();
  });
});
</script>
</head>
<body>
<h3> This is a heading. </h3>
<p> This is a paragraph. </p>
<p> This is another paragraph. </p>
<button> Click Me </button>
</body>
</html>
```

2204030100654

* Jquery Selectors

- Jquery selectors are one of the most important part of the jquery library.
- It allow us to select and manipulate HTML element.

(i) Element Selector

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>
<h3>This is a heading.</h3>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click Me to hide paragraphs.</button>
</body>
</html>
```

(ii) Id Selector

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

```
<head>
<script src = "https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"> </script>
<script>
$ document.ready (function () {
$("button").click (function () {
$("#test").hide ();
});
});
</script>
</head>
<body>
<p> This is a heading. </p>
<p id = "test"> This is a paragraph. </p>
<button> Click Me </button>
</body>
<html>
```

(iii) Class Selector

```
<!DOCTYPE html>
<html>
<head>
<script src = "https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"> </script>
<script>
$ document.ready (function () {
$("button").click (function () {
$(".test").hide ();
});
});
</script>
</head>
<body>
```

```
<h3 class="test"> This is a heading. </h3>
<p class="test"> This is a paragraph. </p>
<p> This is another paragraph. </p>

<button> Click Me </button>
</body>
</html>
```

* More Examples of Jquery Selectors

1. $("*") - selects all elements
2. $(this) - selects the current HTML element
3. $("p.intro") - selects all <p> elements with class = "intro".
4. $("p:first") - selects the first <p> element.
5. $("ul li:first") - selects the first <li> element of the first
6. $("ul li:first-child") - selects the first <li> element of every
7. $("[href]") - Selects all elements with an href attribute
8. $("a[target!='_blank']") - selects all <a> elements with a target attribute value NOT equal to "_blank"
9. $("a[target='_blank']") - selects all <a> elements with a target attribute value equal to "_blank"
10. $(":button") - selects all <button> elements and <input> elements of type = "button".

11. $("tr:even") - selects all even <tr> elements.

12. $("tr:odd") - selects all odd <tr> elements.

* Jquery Event Methods

→ Jquery is tailor-made to respond to events in an HTML page.

Events - All the different visitors actions that a web page can respond to are called events.

Mouse Events	Keyboard Events	Form Events	Document / Window events
click	Keypress	submit	load
dblclick	Keydown	change	resize
mouseenter	Keyup	focus	scroll
mouseleave		blur	unload

→ Commonly Used jQuery Event Methods.

1. click()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
        3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("p").click(function(){
  $(this).hide();
});
```

```
});
</script>
</head>
<body>
<p> If you click on me, I will disappear. </p>
<p> Click me too! </p>
<div> If you click on me. I will not disappear. </div>
</body>
</html>
```

2. dblclick ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>
<p> If you click on me, I will disappear. </p>
<p> Click me too! </p>
</body>
</html>
```

3. mouseenter ()

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

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#p1").mouseenter(function(){
alert("You entered p1!");
});
});
</script>
</head>
<body>
<p id="p1">Enter this paragraph.</p>
</body>
</html>
```

4. mouseleave()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#p1").mouseleave(function(){
alert("Bye! You now leave p1!");
});
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

5. mousedown()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#p1").mousedown(function(){
alert("Mouse down over p1!");
});
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

6. mouseup()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#p1").mouseup(function(){
alert("Mouse up over p1!");
});
});
</script>
</head>
```

```
<body>
<p id="p1"> This is a paragraph. </p>
</body>
</html>
```

7. hover()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#p1").hover(function(){
alert("You entered p1!");
});, function(){
alert("Bye! You now leave p1!");
});
});
</script>
</head>
<body>
<p id="p1"> This is a paragraph. </p>
</body>
</html>
```

8. focus() and blur()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"></script>
```

```
<script>
$(document).ready(function(){
 $("input").focus(function(){
  $(this).css("background-color","yellow");
 });
 $("input").blur(function(){
  $(this).css("background-color","green");
 });
});
</script>
</head>
<body>
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">
</body>
</html>
```

9. The on() Method

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
        3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("p").on("click",function(){
  $(this).hide();
 });
});
</script>
</head>
<body>
<p>If you click on me, I will disappear.</p>
<p>Click me too!</p>
```

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

→ Adding multiple events :-

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("p").on({
mouseenter: function(){
$(this).css("background-color", "lightgray");
},
mouseleave: function(){
$(this).css("background-color", "lightblue");
},
click: function(){
$(this).css("background-color", "yellow");
}
});
});
</script>
</head>
<body>
<p> Click or move the mouse pointer over this
paragraph. </p>
</body>
</html>
```

DATE
PAGE

* Jquery Effects :-

(i) jQuery Hide / show

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
        3.6.3/jquery.min.js"></script>
<script>
 $(document).ready(function(){
  $("#hide").click(function(){
   $("p").hide();
  });
  $("#show").click(function(){
   $("p").show();
  });
 });
</script>
</head>
<body>
<p> If you click on the "Hide" button, I will disappear.
</p>
<button id="hide"> Hide </button>
<button id="show"> Show </button>
</body>
</html>
```

→ This example demonstrates the speed parameter ~~hide~~ with hide():

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

```
<script src = " https: // ajax. googleapis . com / ajax / libs /
 jquery / 3.6.3 / jquery. min. js " > </script>
<script >
$ (document) . ready ( function ( ) {
 $ ("button") . click (function ( ) {
  $ ("p") . hide (2000) ;
 }) ;
}) ;
</ script >
</ head >
< body >
< button > Hide </ button >
<p> This is a paragraph with little content. </p>
</ body >
</ html >
```

→ jQuery toggle ( )

```
<! DOCTYPE html >
< html >
< head >
<script src = " https: // ajax. googleapis. com / ajax / libs / jquery /
  3.6.3 / jquery. min. js " > </script>
<script >
$ (document) . ready ( function ( ) {
 $ ("button") . click ( function ( ) {
  $ ("p") . toggle ( ) ;
 }) ;
}) ;
</ script >
</ head >
< body >
<button > Toggle between hiding and showing the paragraph
 </ button >
```

 This is a paragraph with some little content.
</body>
</html>

(ii) jQuery fade ()

→ jQuery has following fade methods :
- fadeIn ()
- fadeOut ()
- fadeToggle ()
- fadeTo ()

★ jQuery fadeIn()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("button").click(function(){
$("#div1").fadeIn();
$("#div2").fadeIn("slow");
$("#div3").fadeIn(3000);
});
});
</script>
</head>
<body>
<p> Demonstrate fadeIn() with different parameters. </p>
<button> Click to fade in boxes. </button>

<div id="div1" style="width:80px; height:80px; display:none; background-color:red;"></div>
```

```
<div id="div2" style="width:80px; height:80px; display:
none; background-color: green;"></div>
<div id="div3" style="width:80px; height:80px;
display: none; background-color: blue;"></div>
</body>
</html>
```

* jQuery fadeOut()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(".div1").fadeOut();
    $(".div2").fadeOut("slow");
    $(".div3").fadeOut(3000);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeOut() with different parameters.
</p>
<button>Click to fade out boxes.</button><br><br>
<div class="div1" style="width:80px; height:50px;
display background-color: blue;"></div>
<div class="div2" style="width:80px; height:80px;
background-color: green;"></div>
<div class="div3" style="width:80px; height:80px;
background-color: red;"></div>
```

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

* jQuery fadeToggle()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeToggle() with different parameters.</p>
<button>Click to fade in/out boxes.</button>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

DATE
PAGE

* jQuery fadeTo()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("button").click(function(){
$("#div1").fadeTo("slow", 0.15);
$("#div2").fadeTo("slow", 0.4);
$("#div3").fadeTo("slow", 0.7);
});
});
</script>
</head>
<body>
<p> Demonstrate fadeTo() with different parameters. </p>
<button> Click to fade boxes. </button> <br> <br>
<div id="div1" style="width: 80px; height: 80px;
background-color: red;"> </div> <br> <br>
<div id="div2" style="width: 80px; height: 80px;
background-color: green;"> </div> <br> <br>
<div id="div3" style="width: 80px; height: 80px;
background-color: blue;"> </div>
</body>
</html>
```

(iii) * jQuery slide()

→ jQuery has following slide effects -

- slideDown()
- slideUp()
- slideToggle()

220403010065Y

* jQuery slideDown()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#flip").click(function(){
$("#panel").slideDown("slow");
});
});
</script>
<style>
#panel, #flip {
padding: 5px;
text-align: center;
background-color: #e5eecc;
border: solid 1px #c3c3c3;
}
#panel {
padding: 50px;
display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide down panel.</div>
<div id="panel">Hello World!</div>
</body>
</html>
```

DATE
PAGE

* jQuery slideUp()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
  3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#flip").click(function(){
  $("#panel").slideUp("slow");
 });
});
</script>
<style>
#panel, #flip {
padding: 5px;
text-align: center;
background-color: #e5eecc;
border: solid 1px #c3c3c3;
}
#panel {
 padding: 50px;
}
</style>
</head>
<body>
<div id="flip">Click to slide up panel</div>
<div id="panel">Hello World!</div>
</body>
</html>
```

DATE
PAGE

* jQuery slideToggle()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideToggle("slow");
  });
});
</script>
<style>
#panel, #flip {
padding: 5px;
text-align: center;
background-color: #e5eecc;
border: solid 1px #c3c3c3;
}
#panel {
padding: 50px;
display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide the panel down or up.</div>
<div id="panel">Hello World!</div>
</body>
</html>
```

(iv) * jQuery Animate

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#btn").click(function(){
$("div").animate({left:'250px'});
$("div").animate({top:'250px'});
$("div").animate({left:'10px'});
$("div").animate({top:'130px'});
$("div").css("background-color","yellow");
});
});
</script>
<script>
$(document).ready(function(){
$("#btn1").click(function(){
$("p1").animate({
left:'250px', opacity:'0.5', height:'150px',
width:'150px'});
$("p1").css("background-color","powderblue");
});
});
</script>
<script>
$(document).ready(function(){
$("#btn2").click(function(){
var div1 = $("div1");
div1.animate({height:'300px'},"slow");
div1.animate({width:'300px', opacity:'0.4'},"slow");
div1.animate({height:'100px', opacity:'0.7'},"slow");
```

```
    div1.animate({width:'100px'}, "slow");
    $("div1").css("background-color", "lightpink");
  });
});
</script>
<script>
  $(document).ready(function(){
    $("#btn3").click(function(){
      var div2 = $("div2");
      div2.animate({left:'100px'}, "slow");
      div2.animate({height:'10px'}, "slow");
      div2.animate({fontSize:'3em'}, "slow");
    });
  });
</script>
<script>
  $(document).ready(function(){
    $("#btn4").click(function(){
      $("div3").animate({height:'toggle'});
      $("div3").css("background-color", "gray");
    });
  });
</script>
<script>
  $(document).ready(function(){
    $("#btn5").click(function(){
      $("div4").animate({
    left:'250px', height '+=150px', width:'+=150px'
    });
  });
});
</script>
</head>
<body>
```

DATE
PAGE

```
<h3> Jquery animate() method </h3>
<button id = "btn" > Start Animation </button>
<div style =" background : #98bf21 ; height : 100px ;
width : 100px ; position : absolute ;" > </div>
<br> <br> <br>
<hr>

<h3> Jquery animate() using multiple properties. </h3>
<button id = "btn1" > Start Animation </button>
<p1 style =" background : rgb (78, 167, 105) ; height : 100px ;
width : 100px ; position : absolute ;" > </div>
<br> <br> <br>
<hr>

<h3> Jquery animate() with Queue functionality </h3>
<button id = "btn2" > Start Animation </button>
<div
<p1 style = " background : #bf219a ; height : 100px ;
width : 100px ; position : absolute ;" > </div1>
<br> <br> <br>
<hr>

<button id = "btn3" > Start Animation </button>
<div2 style =" background : # c059cf ; height : 100px ;
width : 100px ; position : absolute ;" > </div2>
                                        HELLO
<br> <br> <br>
<hr>

<h3> Jquery animate() using predefined values </h3>
<button id = "btn4" > Start Animation <button>
<div3 style =" background : # bf9d21 ; height : 100px ;
width : 100px ; position : absolute ;" > </div3>
<br> <br> <br>
<hr>
```

DATE
PAGE

```
<h3> Jquery animate() using relative values </h3>
<button id = "btn5"> start Animation </button>
<div4 style = "background: #987eec; width: 100px;
 height: 100px; position: absolute;"> </div4>
</body>
</html>
```

(v) jquery stop()

```
<!DOCTYPE html>
<html>
<head>
<script src = "https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"> </script>
<script>
 $(document).ready(function(){
  $("#start").click(function(){
   $("#panel").slideDown(2000);
 });
  $("#stop").click(function(){
   $("#panel").stop();
 });
});
</script>
<style>
#panel, #flip {
padding: 5px; font-size: 18px; text-align: center;
background-color: lemonchiffon; color: crimson;
border-radius: 3px; border: solid 1px lemonchiffon;
}
</style>
</head>
<body>
```

```
<h3> Jquery stop() method </h3>
<button id = "start"> start sliding </button>
<button id = "stop"> Stop sliding </button>
<br> <br>
<div id = "flip"> Click to slide down panel </div>
<div id = "panel"> Hello World! </div>
</body>
</html>
```

(vi) jquery Callback

→ A callback function is executed after the current effect is finished.

⇒ The example below has a callback parameter that a function that will be executed after the hide effect is completed.

```
<!DOCTYPE html>
<html>
<head>
<script src = " https://ajax.googleapis.com/ajax/libs/jquery/
3.6.3/jquery.min.js"> </script>
<script>
 $(document).ready(function(){
   $("button").click(function(){
    $("p").hide("slow", function(){
  alert("The paragraph is now hidden");
 });
 });
 });
</script>
</head>
<body>
```

```
<button> Hide </button>
<p> This is a paragraph with some little content.
</p>
</body>
</html>
```

⇒ The example below has no callback parameter, and the alert box will be displayed before the hide effect is completed :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("button").click(function(){
$("p").hide(3000);
alert("The paragraph is now hidden");
});
});
</script>
</head>
<body>
<button> Hide </button>
<p> This is a paragraph with some little content.</p>
</body>
</html>
```

(vii) jQuery chaining()

→ Chaining allows us to run multiple jQuery methods within a single statement.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/
3.6.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("button").click(function(){
$("#p1").css("color","red").slideDown(2000).
  slideUp(2000);
});
});
</script>
</head>
<body>
<p id="p1">jQuery is fun!</p>
<button>Click me</button>
</body>
</html>
```