Date
Page

- Static & Dynamic Website

Static → In static website, web pages are written by the server which are pre-build source code files build using simple languages such as HTML, CSS or Java script there is no processing of content on the server in static website webpages are written by the server with no change therefore, static website are fast there is no interaction with database also less costly as no need to support server site processing with different languages.

Dynamic → In dynamic website webpage are written by the server which are process during run time they are not pre build webpages build during time to user language with the help of server site scripting language. Ex:- Php, ASP etc & many more supported by the server.

- Client server computing language - The clients request a resource & the server provide that resource. A server make serve multiple client at same time while a client is in contact with only one server. Both client & server usually communicate by a computer network but sometime he may decide in the same system.

Web client - It is an application that communicate with a web server using HTTP. The basic objective of the web server is to store, process & deliver web pages to the user.

Web Browser - A web browser is application software for accessing When a user request a web page from a particular website, the browser retrive its files from a web server & then display the page on the users screen Browser are used on range of devices including desktop, tablets & smartphones.

- Client side & server side scripting language - Web browser executes clients side scripting it is used when browser have all codes. Source code is used to transfer from web server to users computer over the internet & runs directly on the browsers. It is also used for validation & functioning for users event.

Web server are used to server side scripting they are basically used to create dynamic webpages it can also excess files system residing at the web server. A server side environment that runs on scripting language is web server

* HTML

Introduction to HTML - HTML documents structured tags HTML comments, text formatting, inserting special characters additional image & sounds, anchor tags, liststyle lists, tables, frames & floating frames, developing forms, Image maps

- HTML is the standard markup language for creating web page
- HTML describe structure of web page
- HTML consists of a series of element
- HTML elements tell the browser how to display the content
- Eg <!Doctype Html>
  <html>
  <Head>
  <title> page title </title>
  </Head>
  <Body>
  <h1> My first Heading </h1>
  <P> My first paragraph <P>
  </Body>
  </Html>

Html is not case sensitive

The style attribute - It is used to add styles to element such as color, font size & more

<P Style = "color: red"; > This is red Paragraph </P>

- Background color - The css bg color property defines the bg color of an HTML document
  <body style="background-color: powder blue;">
  <h1 style="background-color: powder blue;">This is heading </h1>
  <h1 style="color: blue">This is a heading </h1>

- Fonts - The css fonts family property defines the font to be used for an HTML element

  <P style="font-family: verdana;">This is a heading</P>
  <P style="font family courier">This is a para</P>

- Text size - The css font size property defines the tag size for an HTML element
  <h1 style="font size: 300%;">This is heading </h1>
  <P style="font size: 160%;">This is paragraph</P>

- Text alignment - The css text-align property defines the horizontal text alignment for an HTML element
  <h1 style="text-align: center;">center heading </h1>

- HTML colors are specified with predefined color name or with RGB, HEX, HSL, RGBA or HSLA
  In HTML a color can be
  Tomato, orange, Dodger Blue, Medium sea green, Gray, Sitate Blue, violet, light orange

- Border color
  <h1 style="border: 2px solid Tomato;">Hello world</h1>

- Color values - rgb(255,99,71)
  #ff6347
  hsl(9,100%,64%)
  rgba (255,99,71,05)
  hsla(9,100%,64%,0.5)

- hsl - hue, saturation & lightness
  ff6347 HSLA color values are an extension of HSL

- Saturation - is a % value 0 means shade of gray and 100% is full color
- lightness - is also a % value 0 is black 100% is wh

- HTML list - It allow web developers to group a set of related items in a list.
  Ex:=> unordered list
  - item
  - item

  Ordered list
  1. first item
  2. Second item

=> Unorder html list - An unordered list starts with the <ul> tag. each list item start with the <li> tag. The list item will be marked with a

bullets by default

Ex:- <ul>
<li> coffee</li>
<li> tea </li>
<li> Milk</li>
</ul>

Output
- Coffee
- Tea
- Milk

Ordered html list - An ordered list start with the <ol> tag. each list item start with the <li> tag. The list item will be marked with no's by default

Ex:- <ol>
<li> coffee </li>
<li>Tea </li>
<li>Milk </li>
</ol>

Output
1. Coffee
2. Tea
3. Milk

Html description list - Html also support html list. A description list is a list of terms, with a description of each term. The <dl> tag define the description list <dt> define the term (name) <dd> tag describe each term.

Ex:- <dl>
<dt> coffee </dt>
<dd> black hot drink</dd>
<dt> Milk </dt>
<dd> white cold drink </dd>
</dl>

Output
Coffee
- black hot drink
Milk
- white cold drink

Unordered list - The css list style property is used to define a style of the list item marker

| | value | description |
|---|---|---|
| 1. | disc | sets the list item marker to a bullets (default) |
| 2. | circle | sets the list item marker to a circle |
| 3 | square | Set the list item marker to a square |
| 4 | None | The list item will not marked |

<ul style = "list-style-type: disc;">
<li> coffee </li>
<li> tea </li>
<li> Milk </li>
<ul>

| Ordered list | Type | Description |
|---|---|---|
| 1 | Type="1" | The list item will be numbered with a numbers (default) |
| 2 | Type="A" | The list item will be ordered with upper case letter |
| 3 | Type="a" | The list item will be numbered with lower case |
| 4 | Type="I" | The list item will be numbered with upper case roman no. |
| 5 | Type="i" | The list item will be numbered with lower case roman no. |

Ex:- <ol Type="I">
<li> coffee </li>
<li> Tea </li>
<li> Milk </li>
</ol>

output:
I coffee
II Tea
III Milk

- HTML table - It allows web developers to arrange data into rows & columns.
- Table cells - Each table cell is defined by a <td> and a close tage </td>. Everything between <td> and </td> are the content of the table cell.
- Table rows - Each table row start with a <tr> tag & end with </tr> tag. tr stands for table row
- Table Header - Sometimes you want your cells to be table Header cells in those cases use the <TH> tag and end with a </th>. th stand for table header

* HTML Table tags

- <caption> - Define a table caption
- <col group> - It specify a group of one or more column in a table for formatting
- <col> It specify column property for each column with a <col group> element.
- <thread> - groups a header contain in a table
- <tbody> groups the body contained in a table.
- <tfoot> - groups the footer contain in a table

Eg. <table>
<tr>
<th> Person 1 </th>
<th> Person 2 </th>
<th> Person 3 </th>
</tr>
<td> Email </td>
<td> gmail </td>
<td> linux </td>
</tr>

```
<tr>
<td> 16 </td>
<td> 14 </td>
<td> 10 </td>
<tr>
</table>
```

Output

| Person1 | Person2 | Person3 |
|---|---|---|
| Email | gmail | xmxx |
| 10 | 14 | 10 |

```
<table style = "width 100%">
```

- HTML table borders - To add a border use the CSS border property on table, th and td elements

```
table, th, td {
border, 1 Px solid black;
}
```

- Table Padding & Spacing - Html tables can adjust the padding inside the cells and also the space b/w the cells by default padding is set to 0

Ex -
```
th, td {
Padding : 15 Px;
}
```

```
table {
border- spacing : 30 Px;
}
```

- Html Iframe - <frame> tag - The frame tag was used in html code to define one particular window (frame) within a <frame set> ||
Ex: use the <iframe> tag to embed another document within the current html document.

```
<iframe
src- "https://www.w3schools.com">
</iframe>
```

Html frames are used to divide your browser window into multiple section where each section can load a separate html document
A collection of frames in the browser window is know as frame set. The window is divided into frames in a similar way the tables are organised into rows & columns.

- Disadvantage of frame - Some small devices can't cope with frames often because there screen is not big enough to be divided up Sometimes your page will be displayed differently on different computers due to different screen resolution.

The frame set tag attributes

1. cols - Specifies how many columns are maintained in each set & the size of each column you can specify the width of each column in one of the 4 ways. abs
Absolute value in pixel
To create 3 vertical frames use cols="100,500,100"

2. Rows - The attribute works just like the cols attribute & take the same values, but it is used to specify the rows in the frameset.
Ex:- rows="10% %, 90%"

3. Border This attribute specify the width of the border of each frame if in pixel
Ex: border="5" A value of 0 means no border.

4. Frame border - This attribute specify the whether a three dimensional border should be displayed within frames. This attribute takes value either 1 or 0.
frameborder="0" specify no border.

5. frame spacing - This attribute specify the amount of space b/w frames in a frame set This can take any integer value
framespacing="10" means there should be 10 pixel spacing b/w each frame

frame tag attributes

1. Src - This attribute is used to give the file name that should be noted in the frame. Its value can be any URL

2. name - This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into this is specially important when you some want to create links in one frame that load page into an another frame, in which case the second frame is a name to identify itself as the target of the link.

3. Hargiral ~~width~~ height - This attribute allow you to specify the height of the space between the top & bottom of the frame border & its content. This value is given in the pixel

4. Hargiral width - This attribute allows you to specify the width of the space b/w left & right of the frame border & its content.

5. Scrolling - This attribute controls the appearance of the scroll bars. That appear on the frame This states value yes, no or auto

6. longdisc - This attribute allows you to provide link to another page containing a long description of the contents of the frame.
Ex: longdisc="framedescription.htm"

- The <label> element - The label tag defines a label for many forms elements. It is useful for screen reader users, because the screen reader will read out aloud the label when a user focused on the input element. The label element also help users who have difficulty clicking on very small regions (such as radio button, checkboxes) because when the user user clicks the text within the <label> element, it toggles the input, (this increase the hit area
Tip The for attribute of the <label> must be equal to the id attribute of related element to bind them together. A label can also be bound to an element by placing the element inside the <label> element.

- Radio Buttons
The <Input type="radio"> defines a radio button. let a user select one of a limited no. of choices

```
<form>
<input type="radio" id="html"
name="fav_language" value="HTML">
<label for="html"> HTML</label><br>
<input type="radio" id="css">
<label for="css"> CSS</label><br>
<input type="radio" id="javascript" name="fav
language"
value="javascript">
<label
for="javascript">Javascript</label>
</form>
```

- Checkboxes - Defines a checkbox. It let a user select zero or more options of a limited no. of choices

```
<input type="checkbox">
<form action="/action page.php">
<input type="chat box" id="vehicles"
name="vehicles" value="bike">
```

The action attribute specify where send form data

# CSS

* CSS is the language we use to style an html document. CSS describe how html document should be displayed.

```
<style>body {
    background-color: light blue;
}
h1 {
    color: white;
    text-align: center;
}
p {
    font family: verdana;
    font size: 20 px;
}
</style>
</head>
<body>
<h1> — </h1>
<p> — </p>
</body>
</html>
```

What is CSS?

CSS stands for Cascading style sheets tells how html elements are to be displayed on screen, paper or in other media

CSS save a lot of work it can control the layout of multiple webpage are all at once. External style sheet are stored in css file.

Ques. What is CSS?

CSS is used to define style for webpages, including the design, layout & variations in display for diff devices & screen sizes.

Syntax - A css rule consists of a seelctor & a declaration block

```
h1 {color: blue; font-size: 12 px;}
```

selector = h1; declaration = color: blue; font-size: 12 px; property = color, font-size; value = blue, 12 px

The selector points to the html elements you want to style. The declaration block contains one or more declaration separated by (;)

- Each declaration include css property name & a value, separated by (:)
- Multiple css declaration are separated with ; ← declaration block are surrounded by {}

Eg
```
p {
    color: red;
    text-align: center;
}
```

- CSS selector - They are use to find or select the html element you want to style we can divide CSS selectors into file categories.

1. Simple selectors - Selects elements based on name, id, class
2. Combinator selectors - Select elements based on a specific relationship b/w them.
3. pseudo class selectors - Select an elements based on certain state.
4. pseudo elements selectors - Select ~~&~~ style a part of an element.
5. Attribute selectors - Select elements based on a attribute & attribute value

The CSS element selector select html element based on element name

```
p {
    text-align: center;
    color: red;
}
```

- The CSS id selector - The id selector use the id attribute of an html element to select a specific element the id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a # hash character, followed by the id of the element

```
<style>
    text-align: center;
```

The CSS class selector - It selects HTML elements with a specific class attribute.
To select elements with a specific class, selectors characters followed by the class name.

<html>

You can also specify that only specific html element should be affected by a class.

Eg:- In this eg all <P> elements with class class="center" with be used & center aligned

<style>

Q How to add CSS

→ When a browser reads a stylesheet it will format the html document according to the information in the style sheet

★ Three ways to insert CSS

There are 3 ways of inserting a style sheet.

1. External CSS
2. Internal CSS
3. Inline CSS

1. External CSS- With an external style sheet you can change the look of an entire website by changing just one file. Each html page must include a reference to the external style sheet file inside the <link> element, inside head section.

Eg:- External styles are defined within a <link> element, inside the <head> section of an html page

```
<head>
<link rel="Style sheet"
href="mystyle.css">
</head>
<body>
<h1> This is a heading</h1>
<P> This is a paragraph</P>
</body>
```

Here is how they "my style.css" file looks:

"my style.css"

```
body {
    background-color: light blue;
}
h1 {
    color: navy;
    margin-left: 20 px;
}
```

An External style sheet can be written in any text editor & must be saved with .css extension. The external .css file should not contain any html tags.

2. Internal CSS- An internal style sheet may be used if one single html page has a unique style. The internal style is defined the <style> tag element, inside the head section. Eg:- Internal style a

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

3. Inline CSS- An inline style may be used to apply a unique style for a single element. To use inline style, at the style attribute to relevant element. The style attribute can contain many CSS property

```
<body>
<h1 style="color:blue; text-align:center;">This is heading</h1>
<P style="color:red;">This is a paragraph</p>
</body>
```

- Multiple style sheet - If some properties have been predefined for the same selector (element) in diff style sheet, the value from the last read style sheet will be used Eg- If the internal style is define after the link to the external style sheet, the <h1> elements will be arranged "orange"

```
<head>
<link rel="style sheet" type="text/CSS"
href="mystyle.css">
<style>
h1{
color: orange;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<P> The style of this document is a combination of an external style sheet & external style </P>
</body>
```

Creating styles

The CSS color property defines the text color to be used The css font family property defines the font to be used. The font size property defines the text size to be used. Eg:-

```
<head>
<style>
h1{
color: blue;
font-family: verdana;
font-size: 300%;
}
p{
color: red;
font-family: courier;
font-size: 100%
}
</style>
</head
<body>
<h1> This is a heading </h1>
<P> This is a paragraph <P>
</body>
```

- CSS borders- This property allow you to specify the style, width & color of an elements color borders. Dotted dash solid

Eg- <style>
P{
border: 2 Px solid Powder Blue;
}
</style>
</head>
<body>
<h1> This is a heading </h1>
<P> This is a paragraph </P>
<P> This is a paragraph </P>
</body>

- CSS Padding - The CSS padding property defines a padding (space) between the text & the border.

<style>
P{
border: 2 Px solid powder blue;
Padding: 30 Px;
}
</style>
</head>
<body>
<h1> This is a heading </h1>
<P> —— </P>
<P> —— </P>
<P> —— </P>
</body>
</html>

This is a para
This is a para
This is a para

- CSS margin - The CSS margin property defines a margin (space) outside the border.

<style>
P{
border: 2 Px solid Powder blue;
margin: 50 Px;
}
</style>
<body>
<h1> This is a heading </h1>
<P> This is a paragraph </P>
<P> —— </P>
<P> —— </P>
</body>
</html>

- CSS links can be s

- CSS Property - The following section contains a complete list of standard properties belonging to the latest CSS three specifications :-

| | PROPERTY | DESCRIPTION |
|---|---|---|
| (1) | align-content | Specify the alignment of flexible container items within the flex container |
| (2) | align-items | Specify the default alignment for items within the flex container |
| (3) | align-self | Specify the alignment for selected items within the flex container |
| (4) | animation | Specify the keyframe based animations. |

| | |
|---|---|
| ⑤ animation delay | specifies when the animation will start. |
| ⑥ animation direction | specifies whether the animations should play in reverse or alternate cycle or not. |
| ⑦ animation duration | specify the no. of seconds or milliseconds an animation should take to complete one cycle |
| ⑧ animation-fill-mode | specifies how a css animation should apply styles to its target before & after it is executed. |
| ⑨ animation-iteration-count | specify the no. of times an animation cycle should be played before stopping |
| ⑩ animation play state | specifies whether the animation is running or paused |
| ⑪ border-collapse | specify whether table cell borders are connected or separated |
| ⑫ border-radius | define the shape of the border corners of an element |
| ⑬ Clear | specify the placement of an element in relation to floating element. |
| ⑭ text justify | specify the justification method to use when the text align property is set to justify |

* CSS pseudo classes

Ques What are the pseudo classes

A pseudo classes is used to define a special state of an element. For Eg. It can be used to style an element when a user mouses over it style like

- Style visited & unvisited links - Style an element when it gets focused

- Syntax :- Selector: Pseudo-class {
  property: value;
  }

- Anchor pseudo classes

```
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}
/* visited link */
a:visited {
  color: green;
}
/* mouse over link */
a:hover {
  color: hotpink;
}
```

```
/* selected link */
a:active {
color: blue;
}
</style>
</head>
<body>
<h2> styling a link depending on state </h2>
</body>
```

- pseudo classes & HTML classes - pseudo classes can be combined with html classes when you hover the name in the example it will change color

```
<head>
<style>
a.highlight:hover {
color: #0000;
font size: 22 px;
}
</style>
</head>
<body>
<h2> Pseudo-classes and HTML classes </h2>
<P> when you hover over the first link it will change color & font size: </P>
<P> <a class="highligh"
href="CSS_Syntax.asp"> CSS syntax </a></P>
<P><a href="default.asp"> CSS Tutorial </a></P>
</body>
```

An example of using the hover class using the  element

```
<head>
<style>
div {
background-color: green;
color: white;
padding: 25 px;
text-align: center;
}
div:hover {
background-color: blue;
}
</style>
</head>
<body>
<P> Mouse over the div element below to change its background color: </p>
<div> Mouse over Me </div>
</body>
</html>
```

- All CSS pseudo classes

| Selector | Eg | Eg description |
|---|---|---|
| :active | a:active | selects the active link |
| :checked | input:checked | select every check |
| :disabled | input:disabled | Input element → Select every disabled <Input> element. |

| | | |
|---|---|---|
| :empty | p:empty | selects every <p> element that has no children |
| :enabled | input:enabled | Selects every <enabled> input element. |
| :first-child | p:first-child | selects every <p> element i.e, the first child of its parent |
| :hover | a:hover | selects link on mouse hover |
| :link | a:link | Select all unvisited link |
| :optional | input:optional | selects input element with no required attributes |

```
<head>
<style>
p:first-child {
color: blue;
}
</style>
</head>
<body>
<p>This is some text</p>
<p>this is some text</p>
<div>
<p>This is some text</p>
<p>This is some text</p>
</div>
</body>
```

UNIT-III

## Javascript

It is the world most popular programming language.
Javascript is the programming language of the web.
It is easy to learn.

Q Why study Javascript?

-> Javascript is one of the 3 language all web developers must learn.

1. HTML - To define a content of web pages.
2. Css - To specify the layout of webpages.
3. Javascript - To program the behaviour of web pages.

Javascript can change HTML content

-> One of many Javascript html method is getElementById()

```
<body>
<h2> ——— </h2>
<p id = 'demo'> Javascript can change html content </p>
<button type = 'button' onclick = 'document.getElementById('demo').
innerHTML = 'Hello Javascript!'> Click Me! </button>
</Body>
</html>
```

1. Javascript can display data in different ways writing into an html element using innerHtml
2. Writing into the html output using document.write()
3. Writing into an alert box using window.alert()

- To access an html dom element, Javascript can use the document.getElementById(id)

Id method - The id attribute defines the html element. The inner html property defines the html content

<Script> Tag
In HTML, Javascript code is inserted b/w <Script> tag & </Script>

```
<body>
<h2> ——— </h2>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML=
"My first Java Script";
</Script>
</body>
```

- Javascript values - The javascript defines 2 types of values
  1. Fixed value - they are called literals
  2. Variable values - they are called variables
- Javascript literals - The 2 most important syntax rules for fixed values are -
  Numbers are written with or without decimals
  Strings are text, written within double or single quotes

- Javascript variables - In a programming language, variables are used to store data values. Javascript use the keywords var, let and const to declare variables

```
let x;
x=6;
```

- Javascript operators - Javascript to compute values

```
<body>
<h2> Javascript operators </h2>
<p> ——— </p>
<p id="demo"></p>
<Script>
document.getElementById("demo").innerHTML
=(5+6)*10;
</Script>
</body>
```

- Javascript uses & assignment operator (=) to assign values to variables

```
<body>
<h2> ——— </h2>
<p> ——— </p>
<p id="demo"></p>
<Script>
let x, y;
x=5;
y=6;
document.getElementById("demo").innerHTML
```

=x+y;
</Script>
</Body>

→ Javascript Expressions - An Expression is a combination of values, variables & operator which computes to a value. The computation is an above abaluation

For Eg- The 5 * 10 abaluates to 50.

<body>
<h2> —— </h2>
 —— 

<script>
var x;
x=5;
document.getElementById("demo").innerHTML
=x*10;
</Script>
</Body>

→ Javascript Keywords - They are used to identify action to be performed. The let key tells the browser to create variables.

<body>
<h2> —— </h2>

<script>
let x,y;
x=5+6;
y=x*10;
document.getElementById("demo").innerHTML=y;
</script>
</Body>

The Var keyword also tells the browser to create variables. In these examples using var or let will produce the same result

- Javascript Comments Code after // or between /* and */ is treated as a comment. Comments are ignored & will not be executed

- Javascript identifiers/Names - Identifiers are Javascript names.
Identifiers are used to make variables & keywords & functions. The rules for legal names are the same in most programming language. A Javascript name must begin with a identifier
a) A letter (A-Z or a-z)
b) A dollar sign ($)
c) or an underscore (_)

- NOTE - Subsequent characters may be letters, digits, underscore or dollar sign. Numbers are not allowed as the first letter in name. All Javascript identifiers are case sensitive. The variables last Name and last name are two different variables

declare Javascript

1. Using var
2. Using let
3. Using const
4. Using nothing

Q. What are variables?

Variables are containers for storing data. (Storing data values)

Javascript Let - The let keyword was introduced in 2015 variables defines with that let can't be redeclared. Variables define with that must be declared before use. Var

- Can't be redeclared - Variables define with let can't be redeclared. You can't accidentaly redeclare a ~~var~~ variable.

```
Ex- let x= "John Doe"
    let x=0;
// syntax error: 'x' has already been declared
```

```
Ex -> User x = "John Doe";
      user x = 0;
```

classmate
Date
Page 1

- Block Scope - Before ES6(2016) Javascript add only global scope & function scope. ES6 introduces const these 2 keywords provide Block scope in Javascript Variable declared inside a {} can't be access from outside the block

```
{
  let x=2;
  {
    //x can not be used here
```

Variables declared with the var keyword can not have block scope.

Variables declared inside a {} block can be accessed from outside the block

```
{
  var x=2;
  {
    //x can be used here
```

```
<body>
<h2> —— </h2>
<p id="demo"></p>
<script>
var x=10;
{
  var x=2;
}
document.getElementById("demo").innerHTML=x;
</script>
</body>
```

```
<body>
<h2> ——— </h2>
<p id = "demo"></p>
<script>
let x=10;
{
let x=2;
}
document.getElementById("demo").innerHTML=x
</script>
</body>
```

o/p -> 10

```
var x = 2;     //allowed
let x = 3;     //Not allowed
{
let x = 2;     //allowed
let x=3;       //not allowed
}
{
let x=2;       //allowed
var x=3;       //not allowed
}
```

Date Page 3

⇒ Javascript Const - variables define with const can't be redeclared
assign a value they are declared must to

Correct -
Const PI = 3.14159265353;
Incorrect -
PI -3.14159263359;

Q When to use Javascript or const
Always declare variable with const when you know the value should not be changed use const

a) A new array
b) A new const
c) A new function
d) A new reject

- Const objects & array - The keyword const

It defines a constant reference to a value
Reassign a const value
Reassign

Change the val
Const: array - You change the elements of a cons
array.

```
<body>
<h2> —— </h2>
<p> —— </p>
<p id="demo"></p>
<script>
const cars=["Saab","Volvo","BMW"];
Cars[0]="Toyota";//change an element
cars.push("Audi");//Add an element
document.getElementById("demo").innerHTML=cars;
</script>
</body>
```

- Const objects - You can change the properties of a constant object

```
<body>
<h2> —— </h2>
<p> —— </p>
<p id="demo"></p>
<script>
const car={type:"fiat", model:"500",
color:"white"};// create an object
car.color="red";// change a property
car.owner="Johnson";// Add a property
document.getElementById("demo").innerHTML="car owner is"
+car.owner;
</script>
</body>
```

- Block scope - Declaring a variable with const is similar to let when it comes to block scope. The declaration in the block in this example, is not the same as it declared outside the block

```
<body>
<h2> —— </h2>
<p> —— </p>
<p id="demo"></p>
<script>
const x=10;
{
const x=2;
}
document.getElementById("demo").innerHTML
="x is"+x;
</script>
</body>
```

```
var x=2; // Allowed
var x=3; // Allowed
x=4; // Allowed
```

Ex- `var x=2;` //Allowed
`const x=2;` // Not allowed

```
{
let x=2; //Allowed
const x=2; //Not Allowed
}
{
const x=2; //Allowed
const x=2; //Not Allowed
}
```

```
const x=2;  //Allowed
x=2;        //Not Allowed
var x=2;    //Not Allowed
let x=2;    //Not Allowed
const x=2;  //Not Allowed
```

```
Ex-> const x=2; // Allowed
{
 const x=3; //Allowed
}
{
 const x=4; //Allowed
}
```

Javascript operators - The assignment operator assign a value to a variable.

Eg let x=10;

The additional operators

TYPES OF JAVASCRIPT OPERATORS - There are different types of Javascript operators.

rithmetic operators
signment operators

- Additional Operator - The additional operator (+) adds numbers.

```
let x=5;
let y=2;
let z=x+y;
```

- Types of Javascript Operators - There are different types of Javascript operator.
  - Arithmetic
  - Assignment
  - Comparison
  - Logical
  - Conditional
  - Type

- Arithmetic operators are used to perform arithmetic on numbers

| OPERATOR | DESCRIPTION |
|---|---|
| + | addition |
| - | Subtraction |
| * | Multiplication |
| ** | Exponentiation |
| / | Division |
| % | Modulus |
| ++ | increment |
| -- | decrement |

- Assignment operators - The assignment operators assign values to javascript variables the addition assignment operator adds a value to a variable.

```
<body>
<h1> —— </h1>
<h2> —— </h2>
<p id = "demo"></p>
<script>
var x=10;
x+=5;
document.getElementById("demo").innerHTML=x;
</script>
</body>
```

o/p ⇒ 15

| operator | Example | Same As |
|---|---|---|
| = | x=y | x=y |
| += | x+=y | x=x+y |
| -= | x-=y | x=x-y |
| *= | x*=y | x=x*y |
| /= | x/=y | x=x/y |
| %= | x%=y | x=x%y |
| **= | x**y | x=x**y |
| *= | x*=y | x=x*y |

- Javascript ~~strings~~ comparison operator -

| operator | Description |
|---|---|
| == | equal to |
| === | equal value & equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| ? | ternary operator |

Syntax - (condition) ? x:y

Example
(z<18) ? x:y

- The conditional (ternary) operator is javascript operator that take three operands; a condition followed by a question mark (?), that an expression to execute if the condition is true followed by a colon(:) and finally the expression to execute if the condition is false

- Javascript logical operators -

| operator | description |
|---|---|
| && | logical and |
| \|\| | logical or |
| ! | logical not |

→ Domain is the address, which allows a visitor to easily find your website online. It is URL people type in web browser's address bar to access your site. It offer a convenient way for people to access website

→ web hosting is an OR service that makes your website content accessible on the Internet. When you purchase a hosting plan, you are renting space on a physical server to store all the website's files & data

→ web publishing is a process of creating a website & placing it on web server, & published content may include text, images, videos & other media. Its main aim is to facilitate communication simply by adding content through style, emotion, & space

XML - extensible markup language
store & transport data, human & machine readable

XML
- structure
  - DTD - Document type definition
- content
  - XSD - XML schema definition

<? xml version='1.0' encoding='UTF-8'?>

- Building block of XML
1. Elements <> </>
2. Attributes additional info
3. Entities special chars
4. PC Data parsed text data
5. C Data chars not to parsed

| HTML | XML |
|---|---|
| displays data & describes the structure of a webpage | stores & transfers data |
| Appearance & presentation | focuses on exchange of information |

XML parsing is reading an XML document & providing an interface to the user application for accessing the document

- HTML - define content of web page
  CSS - layout of webpage
  Javascript - program behavior of webpage

Javascript
head tag, body tag, external file
<script type="text/javascript">
document.write("");
</script>

document.getElementBy...

</script>

</body>

else if statement

if (condition 1) {

// block of code to be executed if condition is true

}

else if (condition 2)

// block of code to be executed if the condition 1 is false & condition 2 is true } else {

// block of code to be executed if the condition 1 is false & condition 2 is false

}

Ex- • <body>

<h2> —— </h2>

 —— 

<script>

Const time = new Date().getHours();

let greeting;

if (time < 10) {

greeting = "Good morning";

} else if (time < 20) {

greeting = "Good Day";

} else {

greeting = "Good evening";

document.getElementById

</script>

Bootstrap have some new component, more buttons effects and responsiveness. Bootstrap supports some of stable release of all major browsers and platforms

- Responsive web design

An Internet domain is for accessing service on the internet

Internet domain are identified by unique IP addresses

The term domain is also used in other type of networks to describe other connected Physical network resources. It provides a structure to provide network resources to users under a single administrative

A network domain typically include servers, desktop, printers & other devices

- Virtual domain - It is be used of the domain name eg include. Google.com etc.

```
    text = 'on';
    break;
    default
    text = 'No value found';
    }
    document.getElementById.innerHTML = text;
    </Script>
    </Body>
```

O/P → No value

- if Statement

Syntax-

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

Use if to specify a block of code to be executed, if a specified condition is true

use else to specify a block of code to be executed, if the same condition is false

use else if to specify a new condition put else if the first condition is false

Eg- if Statement

```
<body>
<h2>
<p>
<p id = "demo"> Good evening ! </p>
<script>
if (new Date().getHours() < 18) {
document.getElementById("demo").innerHTML = "Good Day!";
}
</script>
</body>
```

O/P → Display "Good Days" if the hour is less than 18:00
Good day!

- else Statement

Syntax →

```
if (condition) {
  // block of code be executed if the condition is true } else {
    // block of code to be executed if the condition is
    }
  <p id = "demo"></p>
  <Script>
    Const hour = new Date(). getHours();
    let greeting;
    if ( hour < 18) {
    greeting = "Good Day";
    }
    else {
      greeting = "Good evening";
    }
```

four. If nothing matches, a default condition will be used.

2. If else — The if statement is the fundamental control system that allows javascript to make decision & execute statement conditionally.

3. While — The purpose of a while loop is to execute a statement or code block repeatedly as long as expression is true once expression becomes false the loop will be exit.

4. Do-while — Block of statements that are executed at least once and continues to be executed while condition is true.

5. For — Same as while, but initialization, condition and increment/decrement is done in the same line

6. for in — This loop is used to loop through an object properties.

7. Continue — ~~function~~ tells the interpreter to immediately start the next iteration of the loop & skip the remaining code block

Break-statement — It is used to exit a loop early breaking out of the enclosing curly braces

Function — It is a group of reusable code which can be called anywhere in your program the keyword function is used to declare a function

10. Return — It is used to return a value from a function.
11. Var — used to declare a variable.
12. Try — A block of statements on which error handling is implemented.
13. Catch — A block of statements that are executed when an error occur
14. Throw — used to throw an error

- Switchcase Syntax - Switch (expression) {
  case X:
  // code block
  break;
  Case Y:
  //
  code block
  break;
  default:
  // code block
  }

Ex=> <body>
<h2> —— </h2>

<script>
let x = '0';
switch (x) {
code case 0
text = 'off';

typeof(...)
</script>
</body>

O/p → String
String
number
number
number
number

→ <body>
<h2> —— </h2>
 —— 

<script>
let car;
document.getElementById("demo").innerHTML=
car + "<br>" + type of car;
</script>
</body>
O/p undefined
undefined

→ <body>
<h2> —— </h2>
 —— 

<script>
let car="";
document.getElementById("demo").innerHTML=
"The Value is:" +
car + "<br>" +
"the type is:" + type of car;
</script>
</body>

O/p – The value is: The type is: string

- Java Script Statements - They are the commands to tell the browser to what action to perform. Statement are separated by semi-colon (;)
  Eg of Java Script Statements
  document.getElementById("demo").innerHTML="welcome";

→ Following table shows the various Javascript statements.

| Sr. No. | Statement | Description |
|---|---|---|
| 1. | Switch Case | A block of statement in which execution of code depends upon different cases. The interpreter checks each case against the value of an expression until it matches |

→ Javascript control statements

```
<body>
<h2> ——— </h2>
<p> ——— </p>
<p id="demo"></p>
<script>
let x="volvo"+16+4;
document.getElementById("demo").innerHTML=x;
</script>
</body>
```

O/P → Volvo 164

→

```
<body>
<h2> ——— </h2>
<p> ——— </p>
<p id="demo"></p>
<script>

let x=5;
let y=5;
let z=6;
document.getElementById("demo").innerHTML=
(x==y)+"<br>"+(x==z);
</script>
</body>
```

O/P - Tcue
False

→

```
<body>
<h2> ——— </h2>
<p> ——— </p>
<p id="demo"></p>
<script>
const person={
first name: "John",
last name: "Doe",
age:50
eye color: "blue"
};
document.getElementById("demo").innerHTML=
Person.firstname + "is" + person.age + "years old";
</script>
</body>
```

O/P - John is 50 years old.

→

```
<body>
<h2> ——— </h2>
<p> ——— </p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML=
type of "," + "<br>" +
type of "John" + "<br>" +
type of 0 + "<br>" +
type of NaN + "<br>" +
type of (3) + "<br>"
```

- Javascript type operators

| operator | Description |
|---|---|
| type of | returns the type of a variable |
| instance of | returns true if an object is an instance of object type |

Javascript Bitwise operator - Bit operator work on 32 bit numbers any numeric operand in the operation each converted to 32 bit number the result is converted back to a Javascript number

| Same as | result | operator | Description | Eg |
|---|---|---|---|---|
| 0&0 3 | 1 | & | AND | 5&1 |
| 0\|0 a | 5 | \| | OR | 5\|1 |
| ~0\|0 1 | 10 | ~ | NOT | ~5 |
| 0^0 n | 4 | ^ | XOR | 5^1 |
| 0<<0 | 10 | << | left shift | 5<<1 |
| 0\|0<<1 | 12 | >> | right shift | 5>>1 |
| 0&0>>1 | 2 | >>> | unsigned | 5>>>1 |

- Javascript provides different datatypes to hold different types of values. There are two types of datatypes in javascript.

1. Primitive datatypes
2. Non-Primitive datatypes

Javascript is a dynamic type language means you don't need to

You need to use var here to specify the datatype It can hold any type of values numbers,

```
Var a = 40; // holding numbers
Var b = 'Rahul'; // holding string
```

- Javascript Primitive datatypes - There are 5 types of Primitive datatypes in Javascript

| Datatype | Description |
|---|---|
| String | Represent sequence of character Eg 'Hello' |
| Number | Eg - 100 |
| Boolean | Represents boolean value either false or true |
| undefined | Represent undefined value |
| NULL | Represent null that means no value at all |

- Javascript non-primitive datatypes - They are as follows-

| Datatype | Description |
|---|---|
| object | represents through which we can access no. |
| array | represents group of similar values |
| Reg Exp | represent regular expression |