DATE-04/07/25 DAY-01

AIM : BASICS OF HTML – EDITORS , ELEMENTS , ATTRIBUTES , HEADINGS , PARAGRAPH ,STYLES .

HTML EDITORS : HTML editors are tools used to write and edit HTML code for creating web pages. They come in two main types:

**Popular Editors:**

* VS Code (very popular, smart features)
* Sublime Text (lightweight, fast)
* Notepad++ (simple, user-friendly)
* Atom (customizable, open source)
* Brackets (designed for web development)

ELEMENTS : An **HTML element** is a building block of web pages. It defines the structure and content.

1. **Structural Elements**  
   Define layout/sections
   * <html>, <head>, <body>, <div>, <section>, <header>, <footer>
2. **Text Formatting Elements**  
   Style and format text
   * <h1> to <h6>, <p>, <strong>, <em>, <br>, <span>
3. **Form Elements**  
   Take user input
   * <form>, <input>, <textarea>, <button>, <label>, <select>
4. **List Elements**  
   Create lists
   * <ul>, <ol>, <li>, <dl>, <dt>, <dd>
5. **Link and Media Elements**  
   Add links and multimedia
   * <a>, <img>, <video>, <audio>, <iframe>
6. **Table Elements**  
   Create tables
   * <table>, <tr>, <td>, <th>, <thead>, <tbody>
7. **Empty (Void) Elements**  
   No closing tag
   * <br>, <img>, <hr>, <meta>, <input>, <link>

ATTRIBUTES : **Attributes** provide **extra information** about HTML elements. They are always written in the **opening tag**.

| **Attribute** | **Used In** | **Purpose** |
| --- | --- | --- |
| href | <a> | Specifies link URL |
| src | <img>, <script> | Image or file source |
| alt | <img> | Alternative text if image fails |
| title | Most elements | Tooltip text on hover |
| id | All elements | Unique identifier |
| class | All elements | Assigns a class for CSS/JS |
| style | All elements | Inline CSS styling |
| type | <input>, <button> | Specifies input type |
| value | <input>, <button> | Sets default value |
| name | <form> elements | Identifies input fields |
| target | <a> | Where to open the link (\_blank, etc.) |

HEADINGS : Headings (<h1> to <h6>) Used to give heading to the content .

<h1>This is Heading 1</h1>

<h2>This is Heading 2</h2>

<h3>This is Heading 3</h3>

<h4>This is Heading 4</h4>

<h5>This is Heading 5</h5>

<h6>This is Heading 6</h6>

PARAGRAPHS IN HTML : <p> content </p>used to create a paragraph block .

STYLE :Style is used to add CSS (Cascading Style Sheets) rules to your elements to change their appearance (like color, size, alignment, etc.).

Inline style :-

<p style="color:red; font-size:20px;">This is a red paragraph.</p>

Internal style:-

<head>

<style>

p {

color: blue;

font-size: 18px;

}

</style>

</head>

DATE-05/07/25 DAY-02

AIM: HTML FORMATTING , HTML QUATATIONS , COMMENTS , HTML CSS , HTML LINKS.

HTML FORMATTING: Contains several elements for defining text with a special meaning.

|  |  |
| --- | --- |
| [<b>](https://www.w3schools.com/tags/tag_b.asp) | Defines bold text |
| [<em>](https://www.w3schools.com/tags/tag_em.asp) | Defines emphasized text |
| [<i>](https://www.w3schools.com/tags/tag_i.asp) | Defines a part of text in an alternate voice or mood |
| [<small>](https://www.w3schools.com/tags/tag_small.asp) | Defines smaller text |
| [<strong>](https://www.w3schools.com/tags/tag_strong.asp) | Defines important text |
| [<sub>](https://www.w3schools.com/tags/tag_sub.asp) | Defines subscripted text |
| [<sup>](https://www.w3schools.com/tags/tag_sup.asp) | Defines superscripted text |
| [<ins>](https://www.w3schools.com/tags/tag_ins.asp) | Defines inserted text |
| [<del>](https://www.w3schools.com/tags/tag_del.asp) | Defines deleted text |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |

Input : <p><b>This text is bold.</b></p>

Output : **This text is bold**

Input: <p><strong>This text is important!</strong></p>

Output : **This text is important!**

Input : <p><i>This text is italic.</i></p>

Output: *This text is italic.*

Input : <p><em>This text is emphasized.</em></p>

Output : *This text is emphasized.*

Input : <p><small>This is some smaller text.</small></p>

Output : This is some smaller text.

Input: <p>Do not forget to buy <mark>milk</mark> today.</p>

Output : Do not forget to buy milk today.

Input: <p>My favorite color is <del>blue</del> red.</p>

Output : My favorite color is  red.

Input : <p>My favorite color is <del>blue</del> <ins>red</ins>.</p>

Output: My favorite color is  red.

Input : <p>This is <sub>subscripted</sub> text.</p>

Output : This is subscripted text.

Input : <p>This is <sup>superscripted</sup> text.</p>

Output: This is superscripted text.

Quotations in HTML : The HTML <q> tag defines a short quotation. Browsers normally insert quotation marks around the quotation.

Input : <p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>

Output : WWF's goal is to: “Build a future where people live in harmony with nature.”

<abbr>

Input : <p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>

Output : WHO was founded in 1948.

There are two more quotations they are basically <address> to add any residential address it uses a italic format . and second one is <bdo> (bi-directional override) it reverse the order of sentence suppose the word is “Rishi” then it will override as “ihsiR”

COMMENTS IN HTML: TML comments are not displayed in the browser, but they can help document your HTML source code.

<!-- Write your comments here -->

COLORS IN HTML : HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values.

Font color:

INPUT : <h1 style="color:Tomato;">Hello World</h1>

OUTPUT: Tomato Hello World

Background Color:

INPUT : <h1 style="Background-color:Tomato;">Hello World</h1>

OUTPUT: Tomato Hello World

HTML CSS :

* **Inline** - by using the style attribute inside HTML elements
* **Internal** - by using a <style> element in the <head> section
* **External** - by using a <link> element to link to an external CSS file
* Use the HTML style attribute for inline styling
* Use the HTML <style> element to define internal CSS
* Use the HTML <link> element to refer to an external CSS file
* Use the HTML <head> element to store <style> and <link> elements
* Use the CSS color property for text colors
* Use the CSS font-family property for text fonts
* Use the CSS font-size property for text sizes
* Use the CSS border property for borders
* Use the CSS padding property for space inside the border
* Use the CSS margin property for space outside the border

HTML LINKS: The HTML <a> tag defines a hyperlink.

Syntax : <a href="*url*">*link text*</a>

DATE: 06/07/2025 DAY-03

AIM: INSERT AN IMAGES AND TABLES IN HTML.

Images can be add to website using :

<img src="pic\_trulli.jpg" alt="Italian Trulli">

Now you can add alt if in case your images cannot be

Display then alt will handle in such case adding height

And width in some case make your website more reada-b

ble.

<img src="programming.gif" alt="Computer Man" style="width:48px;height:48px;">

IMAGE MAP : The HTML <map> tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more <area> tags.

Css for background images :

<style>  
body {  
  background-image: url('img\_girl.jpg');  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: 100% 100%;  
}  
</style>

HTML TABLES: HTML tables allow web developers to arrange data into rows and columns.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<table>](https://www.w3schools.com/tags/tag_table.asp) | Defines a table |
| [<th>](https://www.w3schools.com/tags/tag_th.asp) | Defines a header cell in a table |
| [<tr>](https://www.w3schools.com/tags/tag_tr.asp) | Defines a row in a table |
| [<td>](https://www.w3schools.com/tags/tag_td.asp) | Defines a cell in a table |

<table>

<tr>

<th>Company</th>

<th>Contact</th>

<th>Country</th>

</tr>

<tr>

<td>Alfreds Futterkiste</td>

<td>Maria Anders</td>

<td>Germany</td>

</tr>

<tr>

<td>Centro comercial Moctezuma</td>

<td>Francisco Chang</td>

<td>Mexico</td>

</tr>

<tr>

<td>Ernst Handel</td>

<td>Roland Mendel</td>

<td>Austria</td>

</tr>

<tr>

<td>Island Trading</td>

<td>Helen Bennett</td>

<td>UK</td>

</tr>

<tr>

<td>Laughing Bacchus Winecellars</td>

<td>Yoshi Tannamuri</td>

<td>Canada</td>

</tr>

<tr>

<td>Magazzini Alimentari Riuniti</td>

<td>Giovanni Rovelli</td>

<td>Italy</td>

</tr>

</table>

DATE -08/07/25 DAY-04

HTML LISTS : HTML lists allow web developers to group a set of related items in lists.

INPUT :

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

OUTPUT :

* Coffee
* Tea
* Milk

INPUT

<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>

OUTPUT

Cofee

Tea

Milk

INLINE : An inline element does not start on a new line An inline element only takes up as much width as necessary.

e.g. : <span> inline content <span>

BLOCK: The <div> element is often used as a container for other HTML elements.

<div>block-level element</div>

Class (class)

* Used to style multiple elements.
* Reusable.

<p class="red">Hello</p>

ID (id)

* Used to style or target one unique element.
* Not reusable.

<h1 id="main">Welcome</h1>

frame (<iframe>)

* Used to embed another webpage inside your page

. <iframe src="https://example.com" width="400" height="300"></iframe>

DATE -09/07-25 DAY -05

AIM: CSS LAYOUTS , RESPONSIVE ,MEDIA QUERIES

CSS LAYOUTS

HTML CODE:

CSS :

MEDIA QUERIES : For better display of website in different screens

<html>

<head>

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

<style>

\* {

  box-sizing: border-box;

}

.left {

  background-color: #2196F3;

  padding: 20px;

  float: left;

  width: 20%; /\* The width is 20%, by default \*/

}

.main {

  background-color: #f1f1f1;

  padding: 20px;

  float: left;

  width: 60%; /\* The width is 60%, by default \*/

}

.right {

  background-color: #04AA6D;

  padding: 20px;

  float: left;

  width: 20%; /\* The width is 20%, by default \*/

}

/\* Use a media query to add a break point at 800px: \*/

**@media screen and (max-width: 800px) {**

**.left, .main, .right {**

**width: 100%;** /\* The width is 100%, when the viewport is 800px or smaller \*/

  }

}

</style>

</head>

<body>

<h2>Media Queries</h2>

<p>Resize the browser window.</p>

<p>Make sure you reach the breakpoint at 800px when resizing this frame.</p>

<div class="left">

  <p>Left Menu</p>

</div>

<div class="main">

  <p>Main Content</p>

</div>

<div class="right">

  <p>Right Content</p>

</div>

</body>

</html>

Output :

WEB LAYOUTS :

SEMANTIC TAGS :

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

aside {

width: 30%;

padding-left: 15px;

margin-left: 15px;

float: right;

font-style: italic;

background-color: lightgray;

}

</style>

</head>

<body>

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>

<aside>

<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>

</aside>

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>

</body>

</html>

OUTPUT :

DATE-10/07/25 DAY 6

AIM: HTML FORMS – INPUT ATTRIBUTES , HTML CANVAS ,

Checkboxes

Code:

<!DOCTYPE html>

<html>

<body>

<h2>Checkboxes</h2>

<p>The <strong>input type="checkbox"</strong> defines a checkbox:</p>

<form action="/action\_page.php">

<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">

<label for="vehicle1"> I have a bike</label><br>

<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">

<label for="vehicle2"> I have a car</label><br>

<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">

<label for="vehicle3"> I have a boat</label><br><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

Form target attributes :

<!DOCTYPE html>

<html>

<body>

<h2>The form target attribute</h2>

<p>When submitting this form, the result will be opened in a new browser tab:</p>

<form action="/action\_page.php" target="\_blank">

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

<input type="submit" value="Submit">

</form>

</body>

</html>

Select tag :

Text Area :

Grouping Data Elements :

DATE-12-07-25 DAY-07

AIM:

INPUT SIZE ATTRIBUTE :

The input min and max attributes :

The input multiple attributes :

The input placeholder attribute :

The input required attribute :

The input height and width attributes :

HTML CANVAS :

<!DOCTYPE html>

<html>

<body>

<p>Image to use:</p>

<img id="scream" src="img\_the\_scream.jpg" alt="The Scream" width="220" height="277">

<p>Canvas to fill:</p>

<canvas id="myCanvas" width="250" height="300"

style="border:1px solid #d3d3d3;">

Your browser does not support the HTML canvas tag.</canvas>

<p><button onclick="myCanvas()">Try it</button></p>

<script>

function myCanvas() {

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

var img = document.getElementById("scream");

ctx.drawImage(img,10,10);

}

</script>

</body>

</html>

HTML SVG:

DATE-14/07/25 DAY -08

AIM: HTML VIDEOS ,HTML AUDIOS AND IFRAME

HTML VIDEOS:

<!DOCTYPE html>

<html>

<body>

<div style="text-align:center">

<button onclick="playPause()">Play/Pause</button>

<button onclick="makeBig()">Big</button>

<button onclick="makeSmall()">Small</button>

<button onclick="makeNormal()">Normal</button>

<br><br>

<video id="video1" width="420">

<source src="mov\_bbb.mp4" type="video/mp4">

<source src="mov\_bbb.ogg" type="video/ogg">

Your browser does not support HTML video.

</video>

</div>

<script>

var myVideo = document.getElementById("video1");

function playPause() {

if (myVideo.paused)

myVideo.play();

else

myVideo.pause();

}

function makeBig() {

myVideo.width = 560;

}

function makeSmall() {

myVideo.width = 320;

}

function makeNormal() {

myVideo.width = 420;

}

</script>

<p>Video courtesy of <a href="https://www.bigbuckbunny.org/" target="\_blank">Big Buck Bunny</a>.</p>

</body>

</html>

OUTPUT:

<!DOCTYPE html>

<html>

<body>

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

</body>

</html>

IFRAME : TO EMBED ANY WEBSITE INTO YOUR CURRENT WEBSITE .

DATE-15/07/25 DAY -09

AIM: INTRODUCTION TO CSS, Background colors and Colors .

Background color to different blocks

Transparent Boxes:

DATE-16/07/25 DAY -10

AIM: BORDER STYLE,BOX MODEL PROPERTY .

Border Style:

Border Radius:

BOX MODEL PROPERTY:

DATE-17/07/25 DAY -11

AIM: TEXT FORMATTING

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

div {

  border: 1px solid gray;

  padding: 8px;

}

h1 {

  text-align: center;

  text-transform: uppercase;

  color: #4CAF50;

}

p {

  text-indent: 50px;

  text-align: justify;

  letter-spacing: 3px;

}

a {

  text-decoration: none;

  color: #008CBA;

}

</style>

</head>

<body>

<div>

  <h1>text formatting</h1>

  <p>This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties.

  The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from this colored

  <a target="\_blank" href="tryit.asp?filename=trycss\_text">"Try it Yourself"</a> link.</p>

</div>

</body>

</html>

OUTPUT:

CODE :

<!DOCTYPE html>

<html>

<head>

<style>

h1 {

  text-decoration: overline;

}

h2 {

  text-decoration: line-through;

}

h3 {

  text-decoration: underline;

}

p.ex {

  text-decoration: overline underline;

}

</style>

</head>

<body>

<h1>Overline text decoration</h1>

<h2>Line-through text decoration</h2>

<h3>Underline text decoration</h3>

<p class="ex">Overline and underline text decoration.</p>

<p><strong>Note:</strong> It is not recommended to underline text that is not a link, as this often confuses

the reader.</p>

</body>

</html>

TEXT SHADOW :

DATE-18/07/25 DAY -12

AIM: FONT FORMATTING , LINK FORMATTING ,CURSOR PROPERTY ,LIST STYLING

The font Property :

LINK FORMATTING:

CURSOR PROPERTY :

LIST STYLING :

<!DOCTYPE html>

<html>

<head>

<style>

ul.a {

  list-style-type: circle;

}

ul.b {

  list-style-type: square;

}

ol.c {

  list-style-type: upper-roman;

}

ol.d {

  list-style-type: lower-alpha;

}

</style>

</head>

<body>

<h2>The list-style-type Property</h2>

<p>Example of unordered lists:</p>

<ul class="a">

  <li>Coffee</li>

  <li>Tea</li>

  <li>Coca Cola</li>

</ul>

<ul class="b">

  <li>Coffee</li>

  <li>Tea</li>

  <li>Coca Cola</li>

</ul>

<p>Example of ordered lists:</p>

<ol class="c">

  <li>Coffee</li>

  <li>Tea</li>

  <li>Coca Cola</li>

</ol>

<ol class="d">

  <li>Coffee</li>

  <li>Tea</li>

  <li>Coca Cola</li>

</ol>

</body>

</html>

DATE-19/07/25 DAY -13

AIM: TABLE STYLING ,DISPLAY, POSITION , FLOAT , OVERFLOW , FLOAT , ALIGN , COMBINATORS , OPACITY

TABLE

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

table, td, th {

border: 1px solid black;

}

table {

border-collapse: collapse;

width: 100%;

}

td {

height: 50px;

vertical-align: bottom;

}

</style>

</head>

<body>

<h2>The vertical-align Property</h2>

<p>This property sets the vertical alignment (like top, bottom, or middle) of the content in th or td.</p>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

OUTPUT:

BORDER COLLAPSED TABLES

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

table {

  border-collapse: collapse;

  width: 100%;

}

th, td {

  padding: 8px;

  text-align: left;

  border-bottom: 1px solid #ddd;

}

</style>

</head>

<body>

<h2>Bordered Table Dividers</h2>

<p>Add the border-bottom property to th and td for horizontal dividers:</p>

<table>

  <tr>

    <th>Firstname</th>

    <th>Lastname</th>

  <th>Savings</th>

  </tr>

  <tr>

    <td>Peter</td>

    <td>Griffin</td>

    <td>$100</td>

  </tr>

  <tr>

    <td>Lois</td>

    <td>Griffin</td>

    <td>$150</td>

  </tr>

  <tr>

    <td>Joe</td>

    <td>Swanson</td>

    <td>$300</td>

  </tr>

  <tr>

    <td>Cleveland</td>

    <td>Brown</td>

    <td>$250</td>

  </tr>

</table>

</body>

</html>

DATE-21/07/25 DAY -14

AIM: NAVIGATION BAR CSS & DROPDOWN CSS

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

ul {

list-style-type: none;

margin: 0;

padding: 0;

width: 200px;

background-color: #f1f1f1;

}

li a {

display: block;

color: #000;

padding: 8px 16px;

text-decoration: none;

}

/\* Change the link color on hover \*/

li a:hover {

background-color: #555;

color: white;

}

</style>

</head>

<body>

<h2>Vertical Navigation Bar</h2>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#news">News</a></li>

<li><a href="#contact">Contact</a></li>

<li><a href="#about">About</a></li>

</ul>

</body>

</html>

WEBLAYOUT :

<!DOCTYPE html>

<html>

<head>

<style>

body {

  margin: 0;

}

ul {

  list-style-type: none;

  margin: 0;

  padding: 0;

  width: 25%;

  background-color: #f1f1f1;

  position: fixed;

  height: 100%;

  overflow: auto;

}

li a {

  display: block;

  color: #000;

  padding: 8px 16px;

  text-decoration: none;

}

li a.active {

  background-color: #04AA6D;

  color: white;

}

li a:hover:not(.active) {

  background-color: #555;

  color: white;

}

</style>

</head>

<body>

<ul>

  <li><a class="active" href="#home">Home</a></li>

  <li><a href="#news">News</a></li>

  <li><a href="#contact">Contact</a></li>

  <li><a href="#about">About</a></li>

</ul>

<div style="margin-left:25%;padding:1px 16px;height:1000px;">

  <h2>Fixed Full-height Side Nav</h2>

  <h3>Try to scroll this area, and see how the sidenav sticks to the page</h3>

  <p>Notice that this div element has a left margin of 25%. This is because the side navigation is set to 25% width. If you remove the margin, the sidenav will overlay/sit on top of this div.</p>

  <p>Also notice that we have set overflow:auto to sidenav. This will add a scrollbar when the sidenav is too long (for example if it has over 50 links inside of it).</p>

  <p>Some text..</p>

  <p>Some text..</p>

  <p>Some text..</p>

  <p>Some text..</p>

  <p>Some text..</p>

  <p>Some text..</p>

  <p>Some text..</p>

</div>

</body>

</html>

HORIZONTAL NAV BAR :

CODE :

<!DOCTYPE html>

<html>

<head>

<style>

ul {

  list-style-type: none;

  margin: 0;

  padding: 0;

  overflow: hidden;

}

li {

  float: left;

}

li a {

  display: block;

  padding: 8px;

  background-color: #dddddd;

}

</style>

</head>

<body>

<ul>

  <li><a href="#home">Home</a></li>

  <li><a href="#news">News</a></li>

  <li><a href="#contact">Contact</a></li>

  <li><a href="#about">About</a></li>

</ul>

<p><b>Note:</b> If a !DOCTYPE is not specified, floating items can produce unexpected results.</p>

<p>A background color is added to the links to show the link area. The whole link area is clickable, not just the text.</p>

<p><b>Note:</b> overflow:hidden is added to the ul element to prevent li elements from going outside of the list.</p>

</body>

</html>

Fixed Top Navigation Bar:

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

body {margin:0;}

ul {

  list-style-type: none;

  margin: 0;

  padding: 0;

  overflow: hidden;

  background-color: #333;

  position: fixed;

  top: 0;

  width: 100%;

}

li {

  float: left;

}

li a {

  display: block;

  color: white;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

}

li a:hover:not(.active) {

  background-color: #111;

}

.active {

  background-color: #04AA6D;

}

</style>

</head>

<body>

<ul>

  <li><a class="active" href="#home">Home</a></li>

  <li><a href="#news">News</a></li>

  <li><a href="#contact">Contact</a></li>

  <li><a href="#about">About</a></li>

</ul>

<div style="padding:20px;margin-top:30px;background-color:#1abc9c;height:1500px;">

  <h1>Fixed Top Navigation Bar</h1>

  <h2>Scroll this page to see the effect</h2>

  <h2>The navigation bar will stay at the top of the page while scrolling</h2>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

  <p>Some text some text some text some text..</p>

</div>

</body>

</html>

DROP DOWN :

<!DOCTYPE html>

<html>

<head>

<style>

.dropbtn {

  background-color: #4CAF50;

  color: white;

  padding: 16px;

  font-size: 16px;

  border: none;

  cursor: pointer;

}

.dropdown {

  position: relative;

  display: inline-block;

}

.dropdown-content {

  display: none;

  position: absolute;

  background-color: #f9f9f9;

  min-width: 160px;

  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

  z-index: 1;

}

.dropdown-content a {

  color: black;

  padding: 12px 16px;

  text-decoration: none;

  display: block;

}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {

  display: block;

}

.dropdown:hover .dropbtn {

  background-color: #3e8e41;

}

</style>

</head>

<body>

<h2>Dropdown Menu</h2>

<p>Move the mouse over the button to open the dropdown menu.</p>

<div class="dropdown">

  <button class="dropbtn">Dropdown</button>

  <div class="dropdown-content">

  <a href="#">Link 1</a>

  <a href="#">Link 2</a>

  <a href="#">Link 3</a>

  </div>

</div>

<p><strong>Note:</strong> We use href="#" for test links. In a real web site this would be URLs.</p>

</body>

</html>

Dropdown Menu inside a Navigation Bar:

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

ul {

  list-style-type: none;

  margin: 0;

  padding: 0;

  overflow: hidden;

  background-color: #333;

}

li {

  float: left;

}

li a, .dropbtn {

  display: inline-block;

  color: white;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

}

li a:hover, .dropdown:hover .dropbtn {

  background-color: red;

}

li.dropdown {

  display: inline-block;

}

.dropdown-content {

  display: none;

  position: absolute;

  background-color: #f9f9f9;

  min-width: 160px;

  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

  z-index: 1;

}

.dropdown-content a {

  color: black;

  padding: 12px 16px;

  text-decoration: none;

  display: block;

  text-align: left;

}

.dropdown-content a:hover {background-color: #f1f1f1;}

.dropdown:hover .dropdown-content {

  display: block;

}

</style>

</head>

<body>

<ul>

  <li><a href="#home">Home</a></li>

  <li><a href="#news">News</a></li>

  <li class="dropdown">

    <a href="javascript:void(0)" class="dropbtn">Dropdown</a>

    <div class="dropdown-content">

      <a href="#">Link 1</a>

      <a href="#">Link 2</a>

      <a href="#">Link 3</a>

    </div>

  </li>

</ul>

<h3>Dropdown Menu inside a Navigation Bar</h3>

<p>Hover over the "Dropdown" link to see the dropdown menu.</p>

</body>

</html>

DATE-22/07/25 DAY -15

AIM: IMAGE GALLERY IN HTML

Image Gallery code :

<!DOCTYPE html>

<html>

<head>

<style>

div.gallery {

margin: 5px;

border: 1px solid #ccc;

float: left;

width: 180px;

}

div.gallery:hover {

border: 1px solid #777;

}

div.gallery img {

width: 100%;

height: auto;

}

div.desc {

padding: 15px;

text-align: center;

}

</style>

</head>

<body>

<div class="gallery">

<a target="\_blank" href="img\_5terre.jpg">

<img src="img\_5terre.jpg" alt="Cinque Terre" width="600" height="400">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="gallery">

<a target="\_blank" href="img\_forest.jpg">

<img src="img\_forest.jpg" alt="Forest" width="600" height="400">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="gallery">

<a target="\_blank" href="img\_lights.jpg">

<img src="img\_lights.jpg" alt="Northern Lights" width="600" height="400">

</a>

<div class="desc">Add a description of the image here</div>

</div>

<div class="gallery">

<a target="\_blank" href="img\_mountains.jpg">

<img src="img\_mountains.jpg" alt="Mountains" width="600" height="400">

</a>

<div class="desc">Add a description of the image here</div>

</div>

</body>

</html>

Output :

DATE-23/07/25 DAY -16

AIM: ATTRIBUTE SELECTOR , RESPONSIVE FORM

CODE:

<!DOCTYPE html>

<html>

<style>

input[type=text], select {

  width: 100%;

  padding: 12px 20px;

  margin: 8px 0;

  display: inline-block;

  border: 1px solid #ccc;

  border-radius: 4px;

  box-sizing: border-box;

}

input[type=submit] {

  width: 100%;

  background-color: #4CAF50;

  color: white;

  padding: 14px 20px;

  margin: 8px 0;

  border: none;

  border-radius: 4px;

  cursor: pointer;

}

input[type=submit]:hover {

  background-color: #45a049;

}

div {

  border-radius: 5px;

  background-color: #f2f2f2;

  padding: 20px;

}

</style>

<body>

<h3>Using CSS to style an HTML Form</h3>

<div>

  <form action="/action\_page.php">

    <label for="fname">First Name</label>

    <input type="text" id="fname" name="firstname" placeholder="Your name..">

    <label for="lname">Last Name</label>

    <input type="text" id="lname" name="lastname" placeholder="Your last name..">

    <label for="country">Country</label>

    <select id="country" name="country">

    <option value="india">India</option>

      <option value="australia">Australia</option>

      <option value="canada">Canada</option>

      <option value="usa">USA</option>

    </select>

    <input type="submit" value="Submit">

  </form>

</div>

</body>

</html>

Animate width of search input:

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

input[type=text] {

  width: 130px;

  box-sizing: border-box;

  border: 2px solid #ccc;

  border-radius: 4px;

  font-size: 16px;

  background-color: white;

  background-image: url('searchicon.png');

  background-position: 10px 10px;

  background-repeat: no-repeat;

  padding: 12px 20px 12px 40px;

  transition: width 0.4s ease-in-out;

}

input[type=text]:focus {

  width: 100%;

}

</style>

</head>

<body>

<h2>Animate width of search input</h2>

<form>

  <input type="text" name="search" placeholder="Search..">

</form>

</body>

</html>

RESPONSIVE FORM

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

\* {

  box-sizing: border-box;

}

input[type=text], select, textarea {

  width: 100%;

  padding: 12px;

  border: 1px solid #ccc;

  border-radius: 4px;

  resize: vertical;

}

label {

  padding: 12px 12px 12px 0;

  display: inline-block;

}

input[type=submit] {

  background-color: #04AA6D;

  color: white;

  padding: 12px 20px;

  border: none;

  border-radius: 4px;

  cursor: pointer;

  float: right;

}

input[type=submit]:hover {

  background-color: #45a049;

}

.container {

  border-radius: 5px;

  background-color: #f2f2f2;

  padding: 20px;

}

.col-25 {

  float: left;

  width: 25%;

  margin-top: 6px;

}

.col-75 {

  float: left;

  width: 75%;

  margin-top: 6px;

}

/\* Clear floats after the columns \*/

.row::after {

  content: "";

  display: table;

  clear: both;

}

/\* Responsive layout - when the screen is less than 600px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 600px) {

  .col-25, .col-75, input[type=submit] {

    width: 100%;

    margin-top: 0;

  }

}

</style>

</head>

<body>

<h2>Responsive Form</h2>

<p>Resize the browser window to see the effect. When the screen is less than 600px wide, make the two columns stack on top of each other instead of next to each other.</p>

<div class="container">

  <form action="/action\_page.php">

  <div class="row">

    <div class="col-25">

      <label for="fname">First Name</label>

    </div>

    <div class="col-75">

      <input type="text" id="fname" name="firstname" placeholder="Your name..">

    </div>

  </div>

  <div class="row">

    <div class="col-25">

      <label for="lname">Last Name</label>

    </div>

    <div class="col-75">

      <input type="text" id="lname" name="lastname" placeholder="Your last name..">

    </div>

  </div>

  <div class="row">

    <div class="col-25">

      <label for="country">Country</label>

    </div>

    <div class="col-75">

      <select id="country" name="country">

        <option value="australia">Australia</option>

        <option value="canada">Canada</option>

        <option value="usa">USA</option>

      </select>

    </div>

  </div>

  <div class="row">

    <div class="col-25">

      <label for="subject">Subject</label>

    </div>

    <div class="col-75">

      <textarea id="subject" name="subject" placeholder="Write something.." style="height:200px"></textarea>

    </div>

  </div>

  <br>

  <div class="row">

    <input type="submit" value="Submit">

  </div>

  </form>

</div>

</body>

</html>

DATE-24/07/25 DAY -17

AIM: CSS WEBLAYOUT ,UNITS , SPECIFICITY , ROUND CORNERS

CODE:

<!DOCTYPE html>

<html>

<head>

<style>

\* {

  box-sizing: border-box;

}

body {

  font-family: Arial;

  padding: 10px;

  background: #f1f1f1;

}

/\* Header/Blog Title \*/

.header {

  padding: 30px;

  text-align: center;

  background: white;

}

.header h1 {

  font-size: 50px;

}

/\* Style the top navigation bar \*/

.topnav {

  overflow: hidden;

  background-color: #333;

}

/\* Style the topnav links \*/

.topnav a {

  float: left;

  display: block;

  color: #f2f2f2;

  text-align: center;

  padding: 14px 16px;

  text-decoration: none;

}

/\* Change color on hover \*/

.topnav a:hover {

  background-color: #ddd;

  color: black;

}

/\* Create two unequal columns that floats next to each other \*/

/\* Left column \*/

.leftcolumn {

  float: left;

  width: 75%;

}

/\* Right column \*/

.rightcolumn {

  float: left;

  width: 25%;

  background-color: #f1f1f1;

  padding-left: 20px;

}

/\* Fake image \*/

.fakeimg {

  background-color: #aaa;

  width: 100%;

  padding: 20px;

}

/\* Add a card effect for articles \*/

.card {

  background-color: white;

  padding: 20px;

  margin-top: 20px;

}

/\* Clear floats after the columns \*/

.row::after {

  content: "";

  display: table;

  clear: both;

}

/\* Footer \*/

.footer {

  padding: 20px;

  text-align: center;

  background: #ddd;

  margin-top: 20px;

}

/\* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 800px) {

  .leftcolumn, .rightcolumn {

    width: 100%;

    padding: 0;

  }

}

/\* Responsive layout - when the screen is less than 400px wide, make the navigation links stack on top of each other instead of next to each other \*/

@media screen and (max-width: 400px) {

  .topnav a {

    float: none;

    width: 100%;

  }

}

</style>

</head>

<body>

<div class="header">

  <h1>My Website</h1>

  <p>Resize the browser window to see the effect.</p>

</div>

<div class="topnav">

  <a href="#">Link</a>

  <a href="#">Link</a>

  <a href="#">Link</a>

  <a href="#" style="float:right">Link</a>

</div>

<div class="row">

  <div class="leftcolumn">

    <div class="card">

      <h2>TITLE HEADING</h2>

      <h5>Title description, Dec 7, 2017</h5>

      <div class="fakeimg" style="height:200px;">Image</div>

      <p>Some text..</p>

      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>

    </div>

    <div class="card">

      <h2>TITLE HEADING</h2>

      <h5>Title description, Sep 2, 2017</h5>

      <div class="fakeimg" style="height:200px;">Image</div>

      <p>Some text..</p>

      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>

    </div>

  </div>

  <div class="rightcolumn">

    <div class="card">

      <h2>About Me</h2>

      <div class="fakeimg" style="height:100px;">Image</div>

      <p>Some text about me in culpa qui officia deserunt mollit anim..</p>

    </div>

    <div class="card">

      <h3>Popular Post</h3>

      <div class="fakeimg"><p>Image</p></div>

      <div class="fakeimg"><p>Image</p></div>

      <div class="fakeimg"><p>Image</p></div>

    </div>

    <div class="card">

      <h3>Follow Me</h3>

      <p>Some text..</p>

    </div>

  </div>

</div>

<div class="footer">

  <h2>Footer</h2>

</div>

</body>

</html>

The border-radius Property:

<!DOCTYPE html>

<html>

<head>

<style>

#rcorners1 {

  border-radius: 25px;

  background: #73AD21;

  padding: 20px;

  width: 200px;

  height: 150px;

}

#rcorners2 {

  border-radius: 25px;

  border: 2px solid #73AD21;

  padding: 20px;

  width: 200px;

  height: 150px;

}

#rcorners3 {

  border-radius: 25px;

  background: url(paper.gif);

  background-position: left top;

  background-repeat: repeat;

  padding: 20px;

  width: 200px;

  height: 150px;

}

</style>

</head>

<body>

<h1>The border-radius Property</h1>

<p>Rounded corners for an element with a specified background color:</p>

<p id="rcorners1">Rounded corners!</p>

<p>Rounded corners for an element with a border:</p>

<p id="rcorners2">Rounded corners!</p>

<p>Rounded corners for an element with a background image:</p>

<p id="rcorners3">Rounded corners!</p>

</body>

</html>

DATE-25/07/25 DAY -18

AIM: MASKING ,

MASKING CODE:

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<h1>The mask-image Property</h1>

<h3>An SVG mask layer (formed as circles):</h3>

<svg width="600" height="400">

<mask id="svgmask3">

<circle fill="#ffffff" cx="75" cy="75" r="75"></circle>

<circle fill="#ffffff" cx="80" cy="260" r="75"></circle>

<circle fill="#ffffff" cx="270" cy="160" r="75"></circle>

</mask>

<image xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="img\_5terre.jpg" mask="url(#svgmask3)"></image>

</svg>

<h3>Original image:</h3>

<img src="img\_5terre.jpg" alt="Cinque Terre" width="600" height="400">

</body>

</html>

OUTPUT:

DATE-28/07/25 DAY -19

AIM: INTRODUCTION TO JAVASCRIPT,WHERE TO , OUTPUTS , STATEMENTS ,SYNTAX , COMMENTS.

JavaScript in Head:

<!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>

INPUT :

<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph.</p>

<p>Never call document.write after the document has finished loading.

It will overwrite the whole document.</p>

<script>

document.write(5 + 6);

</script>

</body>

</html>

OUTPUT:

**My First Web Page**

My first paragraph.

Never call document.write after the document has finished loading. It will overwrite the whole document.

11

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Statements</h2>

<p>A <b>JavaScript program</b> is a list of <b>statements</b> to be executed by a computer.</p>

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

<script>

let x, y, z; // Statement 1

x = 5; // Statement 2

y = 6; // Statement 3

z = x + y; // Statement 4

document.getElementById("demo").innerHTML =

"The value of z is " + z + ".";

</script>

</body>

</html>

OUTPUT:

**JavaScript Statements**

A **JavaScript program** is a list of **statements** to be executed by a computer.

The value of z is 11.

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Statements</h2>

<p>JavaScript code blocks are written between { and }</p>

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

<p id="demo1"></p>

<p id="demo2"></p>

<script>

function myFunction() {

document.getElementById("demo1").innerHTML = "Hello Dolly!";

document.getElementById("demo2").innerHTML = "How are you?";

}

</script>

</body>

</html>

OUTPUT:

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Comments</h2>

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

<script>

let x = 5; // Declare x, give it the value of 5

let y = x + 2; // Declare y, give it the value of x + 2

// Write y to demo:

document.getElementById("demo").innerHTML = y;

</script>

</body>

</html>

OUTPUT:

**JavaScript Comments**

7

DATE-29/07/25 DAY -20

AIM: JS OPERATORS, JS STRING OPERATORS , JS ARITHMETIC , AND JS ASSIGNMENT.

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Operators</h1>

<h2>The Assignment (=) Operator</h2>

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

<script>

// Assign the value 5 to x

let x = 5;

// Assign the value 2 to y

let y = 2;

// Assign the value x + y to z

let z = x + y;

// Display z

document.getElementById("demo").innerHTML = "The sum of x + y is: " + z;

</script>

</body>

</html>

OUTPUTS:

**JavaScript Operators**

**The Assignment (=) Operator**

The sum of x + y is: 7

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript String Operators</h1>

<p>All conditional operators can be used on both numbers and strings.</p>

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

<script>

let text1 = "A";

let text2 = "B";

let result = text1 < text2;

document.getElementById("demo").innerHTML = "Is A less than B? " + result;

</script>

</body>

</html>

OUTPUTS:

**JavaScript String Operators**

All conditional operators can be used on both numbers and strings.

Is A less than B? true

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arithmetic</h1>

<h2>The ++ Operator</h2>

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

<script>

let x = 5;

--x;

let z = x;

document.getElementById("demo").innerHTML = z;

</script>

</body>

</html>

OUTPUTS:

**JavaScript Arithmetic**

**The ++ Operator**

4

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arithmetic</h1>

<h2>Math.pow()</h2>

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

<script>

let x = 5;

document.getElementById("demo").innerHTML = Math.pow(x,7);

</script>

</body>

</html>

OUTPUT:

**JavaScript Arithmetic**

**Math.pow()**

78125

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arithmetic</h1>

<h2>Operator Precedence</h2>

<p>Multiplication has precedence over addition.</p>

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

<script>

document.getElementById("demo").innerHTML = 100 + 50 \* 3;

</script>

</body>

</html>

OUTPUT:

**JavaScript Arithmetic**

**Operator Precedence**

Multiplication has precedence over addition.

250

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Assignments</h1>

<h2>Simple Assignment</h2>

<h3>The = Operator</h3>

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

<script>

let y = 50

let x = 10 + y;

document.getElementById("demo").innerHTML = "Value of x is: " + x;

</script>

</body>

</html>

OUTPUT:

**JavaScript Assignments**

**Simple Assignment**

**The = Operator**

Value of x is: 60

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Assignments</h1>

<h2>Addition Assignment</h2>

<h3>The += Operator</h3>

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

<script>

let text = "Hello";

text += " World";

document.getElementById("demo").innerHTML = text;

</script>

</body>

</html>

OUTPUT:

**JavaScript Assignments**

**Addition Assignment**

**The += Operator**

Hello World

DATE-30/07/25 DAY -21

AIM: DATA TYPES ,FUNCTIONS AND JS OBJECTS

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Data Types</h2>

<p>JavaScript has dynamic types. This means that the same variable can be used to hold different data types:</p>

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

<script>

let x; // Now x is undefined

x = 5; // Now x is a Number

x = "John"; // Now x is a String

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

OUTPUT:

**JavaScript Data Types**

JavaScript has dynamic types. This means that the same variable can be used to hold different data types:

John

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Strings</h2>

<p>Strings are written with quotes. You can use single or double quotes:</p>

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

<script>

let carName1 = "Volvo XC60";

let carName2 = 'Volvo XC60';

document.getElementById("demo").innerHTML =

carName1 + "<br>" +

carName2;

</script>

</body>

</html>

OUTPUTS:

**JavaScript Strings**

Strings are written with quotes. You can use single or double quotes:

Volvo XC60  
Volvo XC60

INPUTS:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Numbers</h2>

<p>Numbers can be written with, or without decimals:</p>

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

<script>

let x1 = 34.00;

let x2 = 34;

let x3 = 3.14;

document.getElementById("demo").innerHTML =

x1 + "<br>" + x2 + "<br>" + x3;

</script>

</body>

</html>

OUTPUT:

**JavaScript Numbers**

Numbers can be written with, or without decimals:

34  
34  
3.14

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavScript Bigint</h1>

<p>A BigInt can not have decimals.</p>

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

<p>You cannot perform math between a BigInt type and a Number type.</p>

<script>

let x = BigInt("123456789012345678901234567890");

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

OUTPUT:

**JavScript Bigint**

A BigInt can not have decimals.

123456789012345678901234567890

You cannot perform math between a BigInt type and a Number type.

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Booleans</h2>

<p>Booleans can have two values: true or false:</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>

</html>

OUTPUT:

**JavaScript Booleans**

Booleans can have two values: true or false:

true  
false

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Arrays</h2>

<p>Array indexes are zero-based, which means the first item is [0].</p>

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

<script>

const cars = ["Saab","Volvo","BMW"];

document.getElementById("demo").innerHTML = cars[1];

</script>

</body>

</html>

OUTPUT:

**JavaScript Arrays**

Array indexes are zero-based, which means the first item is [0].

Volvo

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Functions</h1>

<p>Invoke (call) a function to convert from Fahrenheit to Celsius:</p>

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

<script>

function toCelsius(f) {

  return (5/9) \* (f-32);

}

let value = toCelsius();

document.getElementById("demo").innerHTML = value;

</script>

</body>

</html>

OUTPUT:

**JavaScript Functions**

Invoke (call) a function to convert from Fahrenheit to Celsius:

NaN

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Functions</h1>

<p>Outside myFunction() carName is undefined.</p>

<p id="demo1"></p>

<p id="demo2"></p>

<script>

let text = "Outside: " + typeof carName;

document.getElementById("demo1").innerHTML = text;

function myFunction() {

  let carName = "Volvo";

  let text = "Inside: " + typeof carName + " " + carName;

  document.getElementById("demo2").innerHTML = text;

}

myFunction();

</script>

</body>

</html>

OUTPUT:

<http://127.0.0.1:3000/OUTCOME.HTML>

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>Creating JavaScript Objects</h1>

<h2>Using an Object Literal</h2>

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

<script>

// Create an Object:

const person = {

  firstName: "John",

  lastName: "Doe",

  age: 50,

  eyeColor: "blue"

};

// Display Data from the Object:

document.getElementById("demo").innerHTML =

person.firstName + " is " + person.age + " years old.";

</script>

</body>

</html>

OUTPUT:

**Creating JavaScript Objects**

**Using an Object Literal**

John is 50 years old.

DATE-31/07/25 DAY-22

AIM : OBJECT PROPERTIES , METHODS , DISPLAY , EVENTS , STRINGS .

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Objects</h1>

<h2>Adding a Method</h2>

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

<script>

// Create an Object

const person = {

firstName: "John",

lastName: "Doe",

id: 5566,

};

// Add a Method

person.name = function() {

return this.firstName + " " + this.lastName;

};

// Display Object Data

document.getElementById("demo").innerHTML =

"My father is " + person.name();

</script>

</body>

</html>

OUTPUT:

**JavaScript Objects**

**Adding a Method**

My father is John Doe

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript HTML Events</h1>

<h2>The onclick Attribute</h2>

<button onclick="document.getElementById('demo').innerHTML=Date()">The time is?</button>

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

</body>

</html>

OUTPUT:

**JavaScript HTML Events**

**The onclick Attribute**

The time is?

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Strings</h1>

<p>The escape sequence \" inserts a double quote in a string.</p>

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

<script>

let text = "We are the so-called \"Vikings\" from the north.";

document.getElementById("demo").innerHTML = text;

</script>

</body>

</html>

OUTPUT:

**JavaScript Strings**

The escape sequence \" inserts a double quote in a string.

We are the so-called "Vikings" from the north.

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript String</h1>

<h2>The charCodeAt() Method</h2>

<p>The charCodeAt() method returns the unicode of the character at a given position in a string:</p>

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

<script>

let text = "HELLO WORLD";

document.getElementById("demo").innerHTML = text.charCodeAt(1);

</script>

</body>

</html>

OUTPUT:

**JavaScript String**

**The charCodeAt() Method**

The charCodeAt() method returns the unicode of the character at a given position in a string:

69

DATE-01/08/25 DAY-23

AIM :JAVASCRIPT ARRAY .

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arrays</h1>

<h2>Bracket Indexing</h2>

<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>

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

<script>

const cars = ["Saab", "Volvo", "BMW"];

document.getElementById("demo").innerHTML = cars[1];

</script>

</body>

</html>

**JavaScript Arrays**

**Bracket Indexing**

JavaScript array elements are accessed using numeric indexes (starting from 0).

Volvo

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arrays</h1>

<h2>Bracket Indexing</h2>

<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>

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

<script>

const cars = ["Saab", "Volvo", "BMW"];

cars[0] = "Opel";

document.getElementById("demo").innerHTML = cars;

</script>

</body>

</html>

OUTPUT:

**JavaScript Arrays**

**Bracket Indexing**

JavaScript array elements are accessed using numeric indexes (starting from 0).

Opel,Volvo,BMW

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Objects</h1>

<p>JavaScript uses names to access object properties.</p>

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

<script>

const person = {firstName:"John", lastName:"Doe", age:46};

document.getElementById("demo").innerHTML = person.age;

</script>

</body>

</html>

OUTPUT:

**JavaScript Objects**

JavaScript uses names to access object properties.

46

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arrays</h1>

<h2>Bracket Indexing</h2>

<p>JavaScript array elements are accesses using numeric indexes (starting from 0).</p>

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

<script>

const fruits = ["Banana", "Orange", "Apple", "Mango"];

document.getElementById("demo").innerHTML = fruits[fruits.length-1];

</script>

</body>

</html>

OUTPUT:

**JavaScript Arrays**

**Bracket Indexing**

JavaScript array elements are accesses using numeric indexes (starting from 0).

Mango

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arrays</h1>

<h2>Creating an Array</h2>

<p>Avoid using new Array(). Use [] instead.</p>

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

<script>

//const points = new Array(40, 100, 1, 5, 25, 10);

const points = [40, 100, 1, 5, 25, 10];

document.getElementById("demo").innerHTML = points[0];

</script>

</body>

</html>

OUTPPUT:

**JavaScript Arrays**

**Creating an Array**

Avoid using new Array(). Use [] instead.

40

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arrays</h1>

<h2>Nested JavaScript Objects and Arrays.</h2>

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

<script>

let x = "";

const myObj = {

  name: "John",

  age: 30,

  cars: [

    {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},

    {name:"BMW", models:["320", "X3", "X5"]},

    {name:"Fiat", models:["500", "Panda"]}

  ]

}

for (let i in myObj.cars) {

  x += "<h2>" + myObj.cars[i].name + "</h2>";

  for (let j in myObj.cars[i].models) {

    x += myObj.cars[i].models[j] + "<br>";

  }

}

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

OUTPUT:

**JavaScript Arrays**

**Nested JavaScript Objects and Arrays.**

**Ford**

Fiesta  
Focus  
Mustang

**BMW**

320  
X3  
X5

**Fiat**

500  
Panda

DATE-04/07/25 DAY-24

AIM : JS SWITCH , JS OBJECT , JS INHERITANCE

SWITCH :

CODE:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript switch</h2>

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

<script>

let day;

switch (new Date().getDay()) {

case 0:

day = "Sunday";

break;

case 1:

day = "Monday";

break;

case 2:

day = "Tuesday";

break;

case 3:

day = "Wednesday";

break;

case 4:

day = "Thursday";

break;

case 5:

day = "Friday";

break;

case 6:

day = "Saturday";

}

document.getElementById("demo").innerHTML = "Today is " + day;

</script>

</body>

</html>

OUTPUT:

**JavaScript switch**

Today is Thursday

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Classes</h1>

<p>Creating two car objects from a car class:</p>

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

<script>

class Car {

constructor(name, year) {

this.name = name;

this.year = year;

}

}

const myCar1 = new Car("Ford", 2014);

const myCar2 = new Car("Audi", 2019);

document.getElementById("demo").innerHTML =

myCar1.name + " " + myCar2.name;

</script>

</body>

</html>

OUTPUT:

**JavaScript Classes**

Creating two car objects from a car class:

Ford Audi

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Class Inheritance</h1>

<p>Use the "extends" keyword to inherit all methods from another class.</p>

<p>Use the "super" method to call the parent's constructor function.</p>

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

<script>

class Car {

constructor(brand) {

this.carname = brand;

}

present() {

return 'I have a ' + this.carname;

}

}

class Model extends Car {

constructor(brand, mod) {

super(brand);

this.model = mod;

}

show() {

return this.present() + ', it is a ' + this.model;

}

}

const myCar = new Model("Ford", "Mustang");

document.getElementById("demo").innerHTML = myCar.show();

</script>

</body>

</html>

OUTPUT:

**JavaScript Class Inheritance**

Use the "extends" keyword to inherit all methods from another class.

Use the "super" method to call the parent's constructor function.

I have a Ford, it is a Mustang

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Class Static Methods</h1>

<p>A static method is created with the "static" keyword, and you can only call the method on the class itself.</p>

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

<script>

class Car {

constructor(name) {

this.name = name;

}

static hello() {

return "Hello!!";

}

}

const myCar = new Car("Ford");

//You can call 'hello()' on the Car Class:

document.getElementById("demo").innerHTML = Car.hello();

// But NOT on a Car Object:

// document.getElementById("demo").innerHTML = myCar.hello();

// this will raise an error.

</script>

</body>

</html>

OUTPUT:

**JavaScript Class Inheritance**

Use the "extends" keyword to inherit all methods from another class.

Use the "super" method to call the parent's constructor function.

I have a Ford, it is a Mustang

INPUT:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Maps</h1>

<h2>The new Map Method()</h2>

<p>Creating a map from an array:</p>

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

<script>

// Create a Map

const fruits = new Map([

["apples", 500],

["bananas", 300],

["oranges", 200]

]);

let numb = fruits.get("bananas");

document.getElementById("demo").innerHTML = "There are " + numb + " bananas.";

</script>

</body>

</html>

OUTPUT:

**JavaScript Maps**

**The new Map Method()**

Creating a map from an array:

There are 300 bananas.

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript ISO Dates</h2>

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

<script>

const d = new Date("2015-03-25");

document.getElementById("demo").innerHTML = d;

</script>

</body>

</html>

OUTPUT:

**JavaScript ISO Dates**

Wed Mar 25 2015 05:30:00 GMT+0530 (India Standard Time)

INPUT:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Validation</h2>

<p>Please input a number between 1 and 10:</p>

<input id="numb">

<button type="button" onclick="myFunction()">Submit</button>

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

<script>

function myFunction() {

// Get the value of the input field with id="numb"

let x = document.getElementById("numb").value;

// If x is Not a Number or less than one or greater than 10

let text;

if (isNaN(x) || x < 1 || x > 10) {

text = "Input not valid";

} else {

text = "Input OK";

}

document.getElementById("demo").innerHTML = text;

}

</script>

</body>

</html>

OUTPUT: