**HTML**

Introduction to HTML

* HTML describes the content of the page
* HTML stands for **Hyper Text Markup Language**
* HTML is not a programming language, it is a **markup language.**
* HTML elements are represented by tags.
* HTML tags are of two types:
* **Container Tag:** (or Pair Tag) The first tag in a pair is the start tag, the second tag is the end tag. The end tag is written like the start tag, with a forward slash before the tag name. Start and end tags are also called opening tags and closing tags like <b> and </b>.
* **Empty Tag:** Only opening tag is used. For example, <br> or <hr>.

HTML Doc Structure

<HTML> Identifies the doc as an HTML doc.

<HEAD> Includes title, scripts used, style definitions and document descriptions.

<TITLE> Contains document title that appears at the title bar.

</HEAD>

<BODY> All the tags, attributes and information in the webpage body goes here.

</BODY>

</HTML>

HTML Elements and Attributes

1. **Text Elements**

* **<h1> to <h6>:** Headings (Level 1 to Level 6)
* **<p>:** Paragraph
* **<span>:** Inline container for styling or scripting

1. **Links and Anchors**

* **<a>:** Anchor/link
* **href:** URL of the linked resource
* **target:** Specifies where to open the linked document
* **rel:** Relationship between the current document and the linked document
* **title:** Tooltip text
* **<link>:** External resource link
* **rel:** Relationship of the linked document to the current document
* **href :** URL of the linked resource
* **type:** MIME type of the linked resource

1. **Images**

* **<img>:** Image
* **src:** URL of the image
* **alt:** Alternative text for accessibility and SEO
* **width, height**: Dimensions of the image
* **title:** Tooltip text
* **loading:** Specifies the loading behavior
* **decoding:** Specifies how the image should be decoded

1. **Lists**

* **<ul>:** Unordered list
* **<ol>**: ordered list
* **<li>:** List item

1. **Tables**

* **<table>:** Table
* **border:** Border width
* **width:** Width of the table
* **cellspacing:** Spacing between cells
* **cellpadding**: Padding within cells
* **<tr>:** Table row
* **<th>:** Table header cell
* **<td>:** Table data cell

1. **Forms**

* **<form>:** Form
* **Action:** URL where form data is submitted
* **Method:** HTTP method for form submission (e.g., **GET**, **POST**)
* **Target:** Specifies where to display the response after form submission
* **<input>:** Input field
* **type:** Type of input field
* **name:** Name of the input field
* **value:** Default value of the input field
* **placeholder:** Placeholder text
* **required:** Indicates whether the input is required
* **<textarea>:** Text area
* **rows:** Number of visible rows
* **cols:** Number of visible columns
* **maxlength:** Maximum number of characters allowed
* **<select>:** Dropdown/select menu
* **<button>:** Button
* **Type**: Type of button (e.g., submit, reset)

1. **Sematic elements**

* <**header>, <footer>, <nav>, <section>, <aside>, <main>**

1. **Global Attributes**

* **id:** Unique identifier
* **class**: Class name(s) for styling and scripting
* **style:** Inline CSS styles

Text Formatting Tags

* HTML provides several tags for formatting text to convey emphasis, importance, or other visual distinctions.
* Some common text formatting tags include:
* **<strong>:** Represents strong importance or emphasis.
* **<em>:** Represents emphasized text, typically displayed in italics.
* **<mark>:** Highlights or marks text for reference or emphasis.
* **<sub>:** Renders text as subscript, typically used for footnotes or mathematical expressions.
* **<sup>:** Renders text as superscript, commonly used for exponents or references.

Inline and Block Level Elements

HTML elements can be classified as either inline or block-level elements, which affect how they are displayed within the document's layout.

* **Inline Elements**

Inline elements do not start on a new line and only take up as much width as necessary. Examples include <span>, <a>, <strong>,<em>,<img>,<input>,etc.

* **Block-level Elements**

Block-level elements start on a new line and take up the full width available.

Examples include <div>, <p>, <h1> to<h6>, <ul>, <ol>, <li>, <table>,etc.

HTML5 Features

* Semantic Elements
* Audio and Video Support
* Canvas Drawing API
* Offline Web Applications
* Form Enhancements and Input Types

**CSS**

* CSS stands for **Cascading Style Sheets** (CSS)
* Describes the appearance, layout, and presentation of information on a web page
* Describes how information is to be displayed, not what is being displayed
* Can be embedded in HTML document or placed into separate .css file

There are three ways of inserting a style sheet:

1. **Inline Styles**

* An inline style may be used to apply a unique style for a single element.
* To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

1. **Internal Style Sheet**

* An internal style sheet may be used if one single page has a unique style.
* Internal styles are defined within the <style> element, inside the <head> section of an HTML page.

1. **External Style Sheet**

* With an external style sheet, we can change the look of an entire website by changing just one file.
* Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section.

CSS Syntax

* A CSS rule has two main parts: a **Selector** and one or more **Declarations**:

h1{

color:blue;

font-size:12px;

}

* The selector is the HTML element you want to style.
* Each declaration consists of a property and a value.
* The property is the style for the attribute you want to change.
* Each property has a value.

CSS Selectors

1. **Element Selector:** Selects HTML elements based on their tag name.

p{

color: blue;

}

1. **Class Selector:** Selects elements based on their class attribute.

.hi{

background-color: yellow;

}

1. **Id Selector:** Selects a single element based on its unique ID attribute.

#m{

font-size: 24px;

}

1. **Universal selector:** selects all elements in an HTML document.

\*{

padding: 0;

margin: 0;

}

CSS Properties

1. **Text Styling**

* **color:** Sets the color of text.
* **font-family:** Defines the font for text.
* **font-size:** Sets the size of the font
* **font-weight:** Specifies the weight (boldness) of the font.
* **text-align:** Aligns the text horizontally within its container.

1. **Backgrounds**

* **background-color:** Sets the background color of an element.
* **background-image:** Sets an image as the background.
* **background-size:** Sets the size of the background image.
* **background-position:** Sets the position of the background image.
* **background-repeat:** Defines how the background image repeats.

1. **Borders**

* **border:** Sets the width, style, and color of borders.
* **border-radius:** Sets the radius of rounded corners.
* **border-color:** Sets the color of the border.
* **border-width:** Sets the width of the border.
* **border-style:** Sets the style of the border (solid, dashed, dotted, etc.).

1. **Box Model**

* **width, height:** Sets the width and height of an element.
* **margin:** Sets the margin space around an element.
* **padding:** Sets the padding space inside an element.
* **box-sizing:** Defines how the width and height are calculated.

**JAVASCRIPT**

* It is designed to add interactivity to HTML pages.
* It is a scripting language (lightweight programming language)
* It is an interpreted language (it executes without preliminary compilation)
* Usually embedded directly into HTML pages.

Javascript can be inserted to HTML file in two ways:

**Internal JS:** We can add JavaScript directly to our HTML page by writing code inside the <script> tag.

**External JS:** We can write JavaScript code in another files having an extension .js and link this file inside <head> tag of HTML page in which we want to add this code.

JavaScript Variables

* Variables are used to store data values.
* They are declared using the var, let, or const keywords.
* **var:** Traditional way of declaring variables. It has function scope.
* **let:** Allows you to declare variables with block scope.
* **const:** Declares a constant value that cannot be reassigned.

Example :

var x = 10;

let y = “hello”;

const PI = 3.14;

JavaScript Datatypes

JavaScript has the following **primitive data types**:

* **Numbers:** Represents numeric values.
* **Strings:** Represents sequences of characters.
* **Boolean:** Represents true or false values.
* **Undefined:** Represents a variable that has been declared but not assigned a value.
* **Null:** Represents an intentional absence of any object value.

Conditional Statements

* **if statement:** The if statement executes a block of code if a specified condition is true.

if (condition){

//code

}

* **if…else statement:** The if…else statement executes one block of code if the condition is true and another block if the condition is false.

if (condition){

//code to be executed if the condition is true

}

else{

//code to be executed if the condition is true

}

* **else if statement:** The else if statement allows you to specify multiple conditions to execute different blocks of code.

if (condition 1){

//code to be executed if the condition1 is true

}

else if (condition 2){

//code to be executed if the condition2 is true

}

else{

//code to be executed if all conditions are false

}

LOOPS

* **for loop:** The for loop repeats a block of code a specified number of times.

for (initialization; condition; increment/decrement){

code to be executed in each iteration

}

* **while loop:** The while loop repeats a block of code while a specified condition is true.

while(condition){

code to be executed as long as the condition is true

}

* **do…while loop:** The do…while loop is similar to the while loop, but it always executes the block of code at least once before checking the condition.

do{

code to be executed atleast once

}while(condition);

JavaScript Outputs

* **innerHTML():** It is used to get or set the HTML content of an element. It allows you to manipulate the HTML structure of an element dynamically.
* **console.log():** It is used to print messages to the console, which is commonly used for debugging purposes.
* **window.alert():** Displays a dialog box with a specified message and an OK button.
* **document.write():** writes HTML expressions or JavaScript code to a document.