# HTML CSS & JAVASCRIPT

WEB ESSENTIALS: HTML, CSS, AND JAVASCRIPT FUNDAMENTALS

BLAIR TONY

# HTML

HTML also known as Hypertext Markup Language is the standard markup language used to create and design web pages. It provides the structure for web content by defining different elements such as headings, paragraphs, lists, links, images, and more. HTML elements are represented by tags enclosed in angle brackets (< >). Tags typically come in pairs, with an opening tag and a closing tag, and can contain attributes that provide additional information about the element. HTML serves as the backbone of the web, organizing and presenting information in a structured format that browsers can understand and render appropriately. Its simplicity and versatility make it an essential tool for building the foundation of virtually every website and web application on the internet.

# HTML element

The main parts of our element are as follows:

- **The opening tag:** This consists of the name of the element (in this case, p), wrapped in opening and closing **angle brackets**. This states where the element begins or starts to take effect — in this case where the paragraph begins.

- **The closing tag:** This is the same as the opening tag, except that it includes a *forward slash* before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.

- **The content:** This is the content of the element, which in this case, is just text.

- **The element:** The opening tag, the closing tag, and the content together comprise the element.

# Basic tags

- **<DOCTYPE html>:** Defines the document type and version of HTML being used.
- **<html>:** Represents the root element of an HTML document.
- **<head>:** Contains meta-information about the HTML document, such as title, links to external stylesheets or scripts, and meta tags
- **<title>:** Sets the title of the HTML document, which appears in the browser's title bar or tab.
- **<meta>:** Provides metadata about the HTML document, such as character encoding, viewport settings, and authorship information.
- **<body>:** Contains the content of the HTML document, including text, images, links, forms, and other elements displayed on the web page.
- **<h1>, <h2>, <h3>, <h4>, <h5>, <h6>:** Defines headings of various levels, from the most important (<h1>) to the least important (<h6>).
- **<p>:** Defines a paragraph of text.
- **<a>:** Creates hyperlinks to other web pages, files, or locations within the same page.
- **<img>:** Inserts images into the HTML document.
- **<ul>, <ol>, <li>:** Creates unordered lists (<ul>) and ordered lists (<ol>), with list items (<li>) nested within them.
- ** , :**  is a block-level element used to group and format content, while  is an inline element used for styling smaller portions of text.
- **<input>:** Defines an input control within a form, such as text fields, checkboxes, radio buttons, and buttons.
- **<form>:** Creates a form for collecting user input, which can include various input elements like text fields, buttons, checkboxes, and more.

# CSS

CSS, known as Cascading Style Sheets, serves as a potent language utilized to style and arrange HTML elements within a web page. It empowers web developers to manage the visual presentation of web content, encompassing elements like colors, fonts, spacing, positioning, and responsiveness.

## Key Concepts of CSS:

- **Selectors and Declarations:** CSS operates by selecting HTML elements and applying styling rules to them. Selectors target specific elements on the page, and declarations define how those elements should be styled. Declarations consist of property-value pairs, where the property specifies the aspect of the element to style (e.g., color, font-size) and the value defines the specific style to apply (e.g., red, 16px)

- **Box Model: The** box model is a fundamental concept in CSS that defines how elements are rendered on the web page. It consists of the content area, padding, border, and margin. By manipulating these components, developers can control the size, spacing, and layout of elements on the page.

- **Preprocessors and Postprocessors** CSS preprocessors like Sass and Less extend the capabilities of CSS by introducing features such as variables, mixins, and nested rules, which improve code organization and maintainability. Postprocessors like Autoprefixer automatically add vendor prefixes to CSS properties, ensuring compatibility with various web browsers.

- **Cascading and Specificity:** CSS follows a cascading order, meaning that multiple styles applied to the same element can cascade down from more general to more specific rules. Additionally, CSS employs specificity to determine which styles take precedence when conflicting rules are applied to the same element. Specificity is based on the combination of selectors used to target elements.

- **Responsive Design:** With the rise of mobile devices and varying screen sizes, responsive design has become essential in modern web development. CSS offers techniques such as media queries and flexible layout mechanisms (e.g., flexbox, CSS grid) to create websites that adapt and respond to different viewport sizes and orientations.

- **Animations and Transitions**: CSS provides mechanisms for creating animations and transitions, allowing developers to add dynamic visual effects to web pages without relying on JavaScript. Keyframe animations and transition properties enable smooth transitions, transformations, and animations of HTML elements.

## CSS Selectors

**Color and Text Properties:**

- **color**: Sets the color of text.
- **font-size**: Sets the size of the font.
- **font-family**: Sets the font family for text.
- **font-weight**: Sets the boldness of the font.
- **text-align**: Aligns text horizontally.
- **text-decoration**: Adds decorations such as underline or strikethrough to text.
- **line-height**: Sets the height of a line of text.

**Box Model Properties:**

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

**Positioning Properties:**

- **position**: Sets the positioning method of an element.
- **top**, **right**, **bottom**, **left**: Positions the element relative to its containing element.
- **float**: Specifies whether an element should float to the left or right side of its container.
- **clear**: Specifies which sides of an element where other floating elements are not allowed.

**Display Properties:**

- **display**: Specifies the display behaviour of an element.
- **visibility**: Specifies whether an element is visible or hidden.
- **overflow**: Specifies what to do with content that overflows the element's box.
- **Background and Border Properties:**
- **background-color**: Sets the background color of an element.
- **background-image**: Sets one or more background images for an element.
- **border-color**: Sets the color of the border.
- **border-width**: Sets the width of the border.
- **border-style**: Sets the style of the border.

# JAVASCRIPT

JavaScript is a versatile programming language primarily used for creating interactive and dynamic content on websites. It is an essential component of web development alongside HTML and CSS.

- **Cross-Browser Compatibility**: JavaScript enjoys support from all contemporary web browsers, including Chrome, Firefox, Safari, Edge, and various others. This widespread support establishes a uniform scripting environment across diverse browsers, guaranteeing consistent behavior for web applications, irrespective of the user's browser preference. Additionally, JavaScript's versatility extends beyond web development to encompass platforms such as React Native, Ionic, Electron for desktop applications, as well as game development using HTML5 Canvas, WebGL, and beyond.

- **Client-Side Scripting Language**: JavaScript functions as a client-side scripting language, executing within the user's web browser rather than on the web server. This characteristic enables JavaScript to dynamically modify the content of a web page in response to user actions and events.

- **Asynchronous Programming**: JavaScript supports asynchronous programming through mechanisms such as callbacks, promises, and async/await, allowing developers to execute non-blocking code and handle tasks like fetching data from servers, performing animations, or handling user input without blocking the main execution thread.

- **Dynamic Interactivity**: JavaScript enables developers to add interactive elements to web pages, such as form validation, interactive maps, animations, slideshow carousels, and more. With JavaScript, developers can respond to user input in real-time without requiring a page reload.

- **DOM Manipulation**: JavaScript interacts with the Document Object Model (DOM), which is a representation of the HTML elements of a web page. Developers can use JavaScript to access, modify, add, or remove HTML elements and their attributes dynamically, allowing for dynamic updates to the content and structure of a web page.

- **Event-Driven Programming**: JavaScript operates on an event-driven model, where actions or events (e.g., clicking a button, submitting a form, hovering over an element) trigger specific JavaScript functions or behaviors. Developers can define event handlers to respond to these events and execute custom actions accordingly.

- **Versatility and Extensibility**: JavaScript is a highly versatile language that can be used for a wide range of purposes beyond web development. It is commonly used in server-side development (Node.js), mobile app development (React
- **Libraries and Frameworks**: JavaScript has a vast ecosystem of libraries and frameworks that streamline web development and offer pre-built solutions for common tasks and challenges. Examples include jQuery, React.js, Angular, Vue.js, and Express.js, among others.