**HTML NOTES**

**What is HTML?**

* HTML (Hyper Text Markup Language) is the standard language used to create web pages
* HTML was invented in late 1991. However, its first version was written by **Tim Berners-Lee** in the year 1993

**A Brief Overview of HTML Versions**

* HTML (HypeText Markup Language) has evolved over time, with each new version introducing new features and improvements. Here's a brief overview of the major versions:

**HTML 2.0 (1995)**

* **Basic Structure:** Established the fundamental structure of HTML documents, including elements like <html>, <head>, <body>, and <p>.
* **Limited Features:** Offered basic formatting options like bold, italic, and headings, but had limited capabilities for creating complex web pages.

**HTML 3.2 (1996)**

* **Enhanced Features:** Introduced features like tables, lists, and forms, making it possible to create more structured and interactive web content.
* **Widespread Adoption:** Became the standard for web development for a period.

**HTML 4.0 (1997)**

* **Major Revisions:** Introduced significant changes to HTML, including improved support for frames, scripting (JavaScript), and style sheets (CSS).
* **Two Versions:** Released in two versions: Strict, Transitional, and Frameset, offering different levels of compliance and features.

**HTML 5 (2014)**

* **Modernization:** A major overhaul of HTML, designed to be more flexible, semantic, and easier to use.
* **New Elements:** Introduced new elements like <header>, <footer>, <nav>, <article>, and <section> to improve the structure and meaning of web content.
* **Multimedia Support:** Enhanced support for multimedia elements like audio and video, making it easier to embed content directly into web pages.
* **Offline Applications:** Introduced features for creating offline web applications, allowing users to interact with content even without an internet connection.

**HTML 5.1 (2016)**

* **Minor Updates:** A minor revision of HTML 5, addressing some technical issues and adding a few new features.

**HTML 5.2 (2017)**

* **Further Refinements:** Another minor update to HTML 5, focusing on technical corrections and clarifications.

**HTML 5.3 (2022)**

* **Ongoing Development:** The latest version of HTML, which continues to evolve with new features and improvements.

**Note:** While HTML 5 is the most widely used version today, older versions are still supported by many browsers. However, it's generally recommended to use the latest version for optimal compatibility and access to the latest features.

**What is HTML extension?**

There are two types of extension.

i) .html , ii) .htm

**HTML Structure**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body> </body>

</html>

**This is a basic HTML5 structure.**

**<!DOCTYPE html>**: This declaration specifies the document type as HTML5, which is the latest version of HTML.

**<html lang="en">**: This element defines the root of an HTML document. The lang attribute specifies the language of the document, in this case, English.

**<head>**: This section contains meta-information about the HTML page, such as the character encoding, viewport settings, and title.

* **<meta charset="UTF-8">**: This meta tag specifies the character encoding used in the HTML document, which is UTF-8, a widely used character encoding that supports a broad range of languages and characters.
* **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: This meta tag is used to control the viewport, which is the area of the screen where the web page is displayed. The width=device-width part sets the viewport width to the width of the device, and initial-scale=1.0 sets the initial zoom level to 100%. This helps ensure that the web page is displayed correctly on different devices, such as smartphones and tablets.
* **<title>Document</title>**: This element sets the title of the web page, which is displayed in the browser's title bar or tab.

**<body>**: This section contains the visible content of the HTML page, including text, images, links, and other elements. In this example, the body is empty, so the page will be blank when viewed in a web browser.

* **UTF-8** (Unicode Transformation Format 8-bit) is a character encoding standard that's used to represent Unicode characters in electronic communication. It's the most common character encoding on the World Wide Web.

**What are tag, attribute, and element can you briefly explain?**

**Tags**

Tags are the building blocks of HTML documents. They are enclosed within angle brackets (< and >), and they define the structure and content of a web page. For example, <p> is a tag used to define a paragraph.

Ex:- <p> </p>, <div></div>, <h1></h1>

**Attributes**

Attributes provide additional information about an HTML element. They are placed inside the opening tag, separated by spaces, and consist of a name-value pair. For example, <img src="image.jpg"> uses the src attribute to specify the source of the image.

Ex:- <img src="">, <a herf="" target=""> </a>

Attribute:- src, herf, target

**Elements**

An element is a combination of a tag and its content. It consists of an opening tag, content, and a closing tag. For instance, <p>This is a paragraph.</p> is an element, where <p> is the opening tag, "This is a paragraph." is the content, and </p> is the closing tag.

Ex:- <p> Hello World! </p> , Hello world is element

**CSS NOTES**

**What is CSS?**

CSS stands for Cascading style sheets. It describes to the user how to display HTML elements on the screen in a proper format. CSS is the language that is used to style HTML documents. In simple words, cascading style sheets are a language used to simplify the process of making a webpage.

* 1994- Hakon Wium Lie proposed the idea of CSS. 1996- The first version of CSS was invented.

**CSS 1 (1996)**

* **Basic Styling:** Introduced fundamental styling properties like color, font, background, and text.
* **Limited Functionality:** Primarily focused on basic formatting and layout.

**CSS 2 (1998)**

* **Expanded Features:** Added support for more complex styling, including tables, lists, and positioning.
* **Improved Layout Control:** Introduced techniques like floats and absolute positioning for more flexible layout options.

**CSS 3 (2011, Ongoing)**

* **Modular Structure:** Divided into modules (e.g., Selectors, Box Model, Text Effects, Backgrounds) for easier adoption and maintenance.
* **New Features:** Introduced a wide range of new features, including:
  + **Flexible Box Model:** A powerful layout system for creating complex and responsive layouts.
  + **Grid Layout:** A grid-based layout system for arranging content in rows and columns.
  + **Animations:** Capabilities to create dynamic and interactive animations on web pages.
  + **Transitions:** Smoothly transition between different style properties.
  + **Transforms:** Apply various transformations to elements, such as rotation, scaling, and skewing.

**CSS Selectors: Targeting Elements**

**CSS Selectors** are used to specify which HTML elements should be styled. They act as a way to address and select specific elements within a web page.

Here are some common types of CSS selectors:

**The Universal Selectors:- \***

**1. Element Selectors**

* Target elements based on their tag name.
* Example: p (targets all paragraph elements)

**2. Class Selectors**

* Target elements with a specific class attribute.
* Example: .my-class (targets elements with the class "my-class")

**3. ID Selectors**

* Target elements with a unique ID attribute.
* Example: #my-id (targets the element with the ID "my-id")

**4. Attribute Selectors**

* Target elements based on their attributes.
* Examples:
  + [href] (targets elements with an href attribute)
  + [type="button"] (targets elements with a type attribute equal to "button")

**5. Pseudo-classes**

* Target elements based on their state or condition.
* Examples:
  + :hover (targets elements when the mouse hovers over them)
  + :active (targets elements when they are clicked or tapped)
  + :focus (targets elements when they have keyboard focus)

**6. Pseudo-elements**

* Target parts of an element that are not represented by a tag.
* Examples:
  + ::before (inserts content before the element's content)
  + ::after (inserts content after the element's content)

**7. Combinators**

* Combine selectors to target elements based on their relationship to other elements.
* Examples:
  + p.my-class (targets paragraph elements with the class "my-class")
  + div > p (targets paragraph elements that are direct children of a div element)
  + ul li (targets list items within an unordered list)

**Advantages of CSS**

**CSS saves time** - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.

**Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.

**Easy maintenance** - To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.

**Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cellphones or for printing.

**Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it’s a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

**CSS ─ INCLUSION**

There are four ways to associate styles with your HTML document. Most commonly used methods are inline CSS and External CSS.

**1. Embedded CSS - The <style> Element**

* You can put your CSS rules into an HTML document using the ... tags. Rules defined using this syntax will be applied to all the elements available in the document.

***Example:-***

<style type="text/css" media="all">

        h1{

            color: red;

            font-size: 2em;

            text-align: center;

        }

    </style>

**2. Inline CSS - The style Attribute**

* You can use style attribute of any HTML element to define style rules. These rules will be applied to that element only.

***Example:-***

<p style="background: green; font-size: 20px; color: white;">Hello World</p>

**3. External CSS - The <link> Element**

* The element can be used to include an external stylesheet file in your HTML document. An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using element.

***Example:-***

<head>

    <title>Document</title>

    <link rel="stylesheet" href="styles.css">

</head>

[**List the CSS Frameworks.**](https://www.geeksforgeeks.org/10-best-css-frameworks-for-frontend-developers-in-2020)

The best CSS frameworks are:

* Bootstrap
* UIKit
* Semantic UI
* Pure
* Tailwind CSS

**JAVASCRIPT – OVERVIEW**

**What is JavaScript?**

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

**Who is developed** **JavaScript?**

**1995:** Brendan Eich created JavaScript as a scripting language called LiveScript. It was designed to add interactivity to web pages and was initially implemented in Netscape Navigator 2.0.

**Client-Side JavaScript**

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

**What is DOM?**

The Document Object Model (DOM) is a programming interface that represents the structure of a web page as a tree of nodes and objects. It allows developers to manipulate the page's content, style, and structure using scripting languages like JavaScript.

**What are JavaScript Data Types?**

There are three major Data types in JavaScript.

* Primitive
  + [Numbers](https://www.geeksforgeeks.org/javascript-numbers/)
  + [Strings](https://www.geeksforgeeks.org/javascript-string/)
  + [Boolean](https://www.geeksforgeeks.org/javascript-boolean/)
  + [Symbol](https://www.geeksforgeeks.org/javascript-symbol-method/)
* Trivial
  + [Undefined](https://www.geeksforgeeks.org/undefined-in-javascript/)
  + [Null](https://www.geeksforgeeks.org/null-in-javascript/)
* Composite
  + [Objects](https://www.geeksforgeeks.org/objects-in-javascript/)
  + [Functions](https://www.geeksforgeeks.org/functions-in-javascript/)
  + [Arrays](https://www.geeksforgeeks.org/arrays-in-javascript/)

**What would be the result of 3+2+”7″?**

**What are other 5 JS frameworks other than React?**

There are several server-side JavaScript frameworks available to choose from, and **Node.js, Next.js, and Express.js**are the most popular choice among developers.

**Here are 5 of the most popular JavaScript libraries:**

* **jQuery:** A versatile library that simplifies DOM manipulation, event handling, and Ajax requests.
* **Angular:** A comprehensive framework for building web applications, offering a full-featured solution with built-in features like routing, dependency injection, and two-way data binding.
* **Vue.js:** A lightweight framework that focuses on the view layer of an application, providing a declarative syntax for rendering and updating the UI.
* **D3.js:** A powerful library for data visualization, allowing developers to create custom charts, graphs, and visualizations.

**What is React?**

React JS is a JavaScript library used to build user interfaces, especially for single-page applications (SPAs). It's designed to make it easier to create interactive web pages where the content can change dynamically without refreshing the whole page.

Here's a simple breakdown:

1. **Components**: In React, the web page is divided into small pieces called components. Each component can represent a part of the UI, like a button, form, or even a whole section. Components can be reused, making development faster and more efficient.
2. **JSX**: React uses a special syntax called JSX, which looks like HTML but allows you to write JavaScript inside it. This makes it easy to design the structure of your web page.
3. **State**: React components can hold and manage data using something called "state." When the state changes, React automatically updates the UI to reflect those changes.
4. **Props**: These are like function parameters for components. Props allow you to pass data from one component to another.