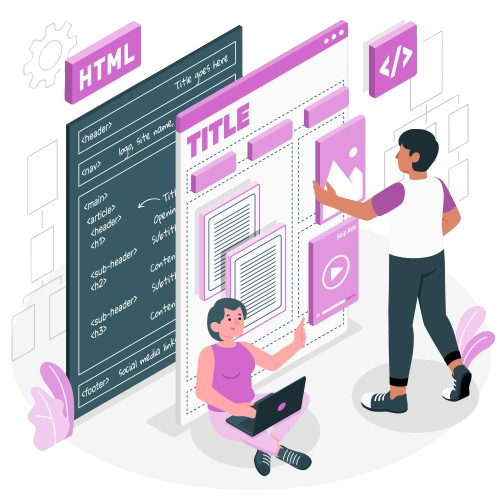
**HTML BRIEF NOTES**

**<!--HYPER TEXT MARKUP LANGUAGE-->**



***BY – SAURABH KUMAR KANNOJIYA***

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **SR.NO.** | **TITLE** | **PAGE NO.** |
| **1.** |  |  |
| **2.** |  |  |
| **3.** |  |  |
| **4.** |  |  |
| **5.** |  |  |
| **6.** |  |  |
| **7.** |  |  |
| **8.** |  |  |
| **9.** |  |  |
| **10.** |  |  |

**WHAT IS HTML?**

**HTML** stands for Hypertext Markup Language. It's the standard markup language used to create and design web pages. **HTML** provides a set of elements or tags that define the structure and content of a web page. These elements are enclosed in angular brackets ‘**< >’** and are usually written in pairs, with an opening tag and a closing tag. The content to be affected by the tag is placed between the opening and closing tags.

**EXPLAIN WHY WE USE HTML WITH EXAMPLE.**

**HTML** is also known as syntactical language or simple and declarative language. In this all **HTML** tags are playing some important role and the core part of **HTML**. Here's an explanation of why we use **HTML**, along with some examples:

1. **STRUCTURE:** **HTML** defines the basic structure of a web page, including headings, paragraphs, lists, and other elements.

For example:

<!DOCTYPE **HTML**>

<**HTML**>

<head>

<title>My First Web Page</title>

</head>

<body>

<h1>Welcome to My Website</h1>

<p>This is a paragraph of text.</p>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

</body>

</**HTML**>

In this example, `<**HTML**>` defines the root element of the document, `<head>` contains metadata like the page title, and `<body>` contains the visible content. Within the `<body>`, we have a heading (`<h1>`), a paragraph (`<p>`), and an unordered list (`<ul>`) with list items (`<li>`).

1. **ELEMENTS:** **HTML** documents consist of a variety of elements, each serving a different purpose. Elements are typically written as opening tags, content, and closing tags. For example, `<p>` is an opening tag for a paragraph, `</p>` is a closing tag, and the text between them is the content of the paragraph.
2. **SEMANTIC MEANING:** **HTML** provides semantic elements that convey the meaning and purpose of content, which is important for accessibility and search engine optimization (SEO).

For example:

<article>

<h2>Article Title</h2>

<p>This is the content of the article.</p>

</article>

<aside>

<h3>Related Links</h3>

<ul>

<li><a href="#">Link 1</a></li>

<li><a href="#">Link 2</a></li>

</ul>

</aside>

Here, `<article>` and `<aside>` elements provide semantic meaning to the content, indicating that one section is the main article and the other is related supplementary content.

1. **HYPERLINKS:** **HTML** allows us to create hyperlinks to navigate between different pages or sections within a page.

For example:

**HTML**

<a href="https://www.example.com">Visit Example Website</a>

This creates a link labeled "Visit Example Website" that, when clicked, takes the user to the URL specified in the `href`` attribute.

Overall, **HTML** is fundamental to web development because it provides the structure, semantics, and interactivity necessary for creating and displaying content on the web. It forms the backbone of every web page and is essential for building a cohesive and user-friendly web experience.

**“HTML WITH VSCODE”**

**How to install VSCODE?**

To install Visual Studio Code (VSCode), you can follow these general steps:

1. **Download VSCode:** 
   1. Go to the official Visual Studio Code website at https://code.visualstudio.com/.
   2. Click on the "Download" button for your operating system (Windows, macOS, or Linux).
2. **Install VSCode:**

**For Windows:**

* 1. Once the download is complete, double-click the downloaded file (usually named something like `VSCodeSetup.exe`).
  2. Follow the instructions in the installation wizard.

**For macOS:**

* 1. Open the downloaded `.dmg` file.
  2. Drag and drop the Visual Studio Code icon into the Applications folder.

**For Linux:**

Installation methods can vary depending on the distribution. You might use package managers like `apt` for Ubuntu/Debian, `yum` for CentOS/Fedora, or `pacman` for Arch Linux. Refer to the official documentation or guides specific to your Linux distribution for detailed instructions.

1. **Open VSCode:**
2. Once the installation is complete, you can typically find Visual Studio Code in your applications menu (Windows/Linux) or launch it from the Applications folder (macOS).
3. Alternatively, you can open VSCode by searching for it in your operating system's search bar.
4. **Optional: Install Extensions:**
5. VSCode has a rich ecosystem of extensions that can enhance its functionality for various programming languages, frameworks, and tools.
6. You can explore and install extensions by clicking on the Extensions view icon in the Activity Bar on the side of the VSCode window, or by pressing `Ctrl+Shift+X` (Cmd+Shift+X on macOS).
7. **Configure Settings:** Customize VSCode to your preferences by configuring settings. You can access settings by clicking on the gear icon in the lower-left corner of the window or by pressing `Ctrl+,` (Cmd+, on macOS).
8. (!) **Emmet Abbreviation -** Emmet is a short hand syntax for writing **HTML** and CSS code quickly and efficiently. It allows you to write abbreviated code snippets that expand into full **HTML** or CSS markup. These snippets are called abbreviations.

For example, in Emmet, the abbreviation **(ul > li \* 3)** would expand to:

**<ul>**

**<li></li>**

**<li></li>**

**<li></li>**

**</ul>**

Here, **ul** represents an unordered list, > indicates a child element, **li** is a list item, and **\*3** specifies that the **li** element should be repeated three times.

Emmet supports a wide range of abbreviations and shortcuts for generating **HTML** structure, as well as **CSS** properties and selectors. It's a powerful tool for improving the speed and efficiency of web development workflows.

1. **Boiler Plate Code –**

<!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>

<!-- This is Boiler Plate Code -->

<h1>Hello World</h1>

</body>

</**HTML**>

**<!DOCTYPE HTML>** This is the document type declaration, which specifies the version of **HTML** being used. In this case, it's **HTML**5.

**<HTML lang="en">** This tag marks the beginning of the **HTML** document and specifies the language of the document (English in this case).

**<head>** This is the opening tag of the head section of the **HTML** document. The head section typically contains meta-information about the document, such as its title, character encoding, stylesheets, scripts, and more.

**<meta charset="UTF-8">** This meta tag specifies the character encoding of the document. `UTF-8` is a widely used character encoding that supports a vast range of characters from various languages and symbols. UTF-16 (16-bit Unicode Transformation Format): It's a character encoding capable of encoding all Unicode characters. UTF-16 uses 16 bits (2 bytes) to represent most characters but can use 4 bytes for characters that are not within the "Basic Multilingual Plane" (BMP), such as emoji or certain characters in less commonly used languages.

**<meta name="viewport" content="width=device-width, initial-scale=1.0">** This meta tag is commonly used in web development for responsive design. It sets the viewport width to the width of the device and sets the initial zoom level to 1.0, ensuring that the web page is displayed properly on different devices with varying screen sizes.

**<title>Document</title>** This tag defines the title of the **HTML** document, which appears in the browser's title bar or tab. In this case, it's set to "Document", but you would typically replace it with a meaningful title that describes the content of your webpage.

**</head>** This is the closing tag for the head section of the **HTML** document.

**<body>** This is the opening tag of the body section of the **HTML** document. The body section contains the actual content of the webpage that will be displayed to users, such as text, images, links, and other elements.

**<!-- This is Boiler Plate Code -->** This is an **HTML** comment, which is used to add notes or annotations to the code that are not displayed in the browser. Comments are ignored by the browser and are useful for documenting your code or temporarily disabling certain parts of it.

**<h1>Hello World</h1>** This is a heading element (`<h1>`) that displays the text "Hello World". Headings are used to define the structure and hierarchy of the content on a webpage, with `<h1>` being the highest level and `<h6>` being the lowest level.

**</body>** This is the closing tag for the body section of the **HTML** document.

**</HTML>** This is the closing tag for the **HTML** document. It marks the end of the **HTML** code. All the **HTML** content of the webpage should be placed between the opening `<**HTML**>` tag and the closing `</**HTML**>` tag.

1. **APACHE 2.0 was the 1st Software License** - The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. This project is part of the Apache Software Foundation. In addition, hundreds of users have contributed ideas, code, and documentation to the project. This file is intended to briefly describe the history of the Apache HTTP Server and recognize the many contributors.
2. **MDN Reference** – MDN stands for **Mozilla Developer Network**. It's a comprehensive resource for web developers, maintained by Mozilla, the organization behind the Firefox web browser. The MDN web documentation includes information on **HTML**, CSS, JavaScript, and various web APIs. In the context of **HTML**, the MDN reference provides detailed documentation and guides on **HTML** elements, attributes, events, and related topics. This documentation covers everything from basic **HTML** syntax to more advanced features like multimedia embedding, forms, and accessibility. The MDN reference is highly regarded in the web development community for its accuracy, depth, and usability. It's often used by developers as a primary source of information when learning about **HTML** or when looking up specific details or best practices while coding.
3. **HTML** Tags: Tags are the building blocks of **HTML** markup. They define different parts of the content and structure of a web page. Tags are enclosed in angle brackets **`< >`**. There are two main types of tags:
   1. Opening Tags: They denote the beginning of an element and are written with the element's name inside angle brackets, e.g., **`<tagname>`**.
   2. Closing Tags: They denote the end of an element and are similar to opening tags but include a forward slash before the element's name, e.g., **`</tagname>`**. Not all elements have closing tags; some are self-closing.
4. **HTML** Elements: Elements are made up of tags, content, and attributes. They define the structure and content of the web page. An element consists of an opening tag, content, and a closing tag (in the case of elements that contain content).

For example:

<p>This is a paragraph element</p>

1. **HTML** Attributes: Attributes provide additional information about **HTML** elements. They are added to the opening tag of an element and are written as name-value pairs, separated by an equal sign **(`=`)** and enclosed in double or single quotes.

For example:

<img src="image.jpg" alt="Image Description">

In this example, **`src`** and `alt` are attributes of the **`<img>`** element.

**HTML ATTRIBUTES**

**HTML** attributes provide additional information about **HTML** elements and control their behavior and appearance. Here's a list of some common **HTML** attributes along with their descriptions:

* + 1. **class:** Assigns one or more class names to an element. Classes can be used by CSS and JavaScript to select and style specific elements.

**HTML**

<div class="container main-content"></div>

* + 1. **id:** Assigns a unique identifier to an element. This ID can be used to uniquely identify the element with CSS or JavaScript.

**HTML**

<div id="header"></div>

* + 1. **style:** Defines inline CSS styles for an element.

**HTML**

<p style="color: red;">This is a red paragraph.</p>

* + 1. **title:** Provides additional information about an element, often displayed as a tooltip when the mouse hovers over the element.

**HTML**

<button title="Submit Form">Submit</button>

* + 1. **href:** Specifies the URL of a link.

**HTML**

<a href="https://www.example.com">Visit Example</a>

* + 1. **src:** Specifies the URL of an embedded image, video, or other resource.

**HTML**

<img src="image.jpg" alt="Description of image">

* + 1. **alt:** Provides alternative text for an image if it cannot be displayed.

**HTML**

<img src="image.jpg" alt="Description of image">

8. **value:** Specifies the initial value of an input element.

**HTML**

<input type="text" value="Enter your name">

9. **name:** Specifies the name of an input element, which is submitted with the form data.

**HTML**

<input type="text" name="username">

10. **placeholder:** Provides a hint to the user about what to enter in the input field.

**HTML**

<input type="text" placeholder="Enter your email">

11. **readonly:** Makes an input field read-only, meaning the user cannot modify its value.

**HTML**

<input type="text" value="Read-only value" readonly>

12. **disabled:** Disables an input field, making it uneditable and preventing its value from being submitted.

**HTML**

<input type="text" value="Disabled value" disabled>

13. **checked:** Indicates whether a checkbox or radio button is selected by default.

**HTML**

<input type="checkbox" checked>

14. **maxlength:** Specifies the maximum number of characters allowed in an input field.

**HTML**

<input type="text" maxlength="10">

15. **min and max:** Define the minimum and maximum values for input elements of types number, range, date, etc.

**HTML**

<input type="number" min="1" max="10">

16. **step:** Specifies the legal number intervals for an input field.

**HTML**

<input type="number" step="2">

17. **pattern:** Specifies a regular expression that the input's value must match for validation.

**HTML**

<input type="text" pattern="[A-Za-z]{3}">

18. **autofocus:** Automatically focuses an element when the page loads.

**HTML**

<input type="text" autofocus>

19. **required:** Specifies that an input field must be filled out before submitting the form.

**HTML**

<input type="text" required>

20. **multiple:** Allows the user to select more than one value from a file input or a select element.

**HTML**

<input type="file" multiple>

21. **form:** Associates the input element with a form element by its ID, which is useful when an input is outside the form element.

**HTML**

<input type="text" form="form1">

22. **formaction:** Overrides the form's action attribute for a specific input element (typically used with submit buttons).

**HTML**

<button type="submit" formaction="submit.php">Submit</button>

23. **formenctype:** Overrides the form's enctype attribute for a specific input element.

**HTML**

<button type="submit" formenctype="multipart/form-data">Submit</button>

24. **formmethod:** Overrides the form's method attribute for a specific input element.

**HTML**

<button type="submit" formmethod="post">Submit</button>

25. **formtarget:** Overrides the form's target attribute for a specific input element.

**HTML**

<button type="submit" formtarget="\_blank">Submit</button>

These attributes help customize the behavior and appearance of **HTML** elements to create more interactive and user-friendly web pages.

|  |  |  |
| --- | --- | --- |
| **\_self** | **\_top** | **\_parent** |
| **Description:** Opens the linked document in the same frame or window as the one that contains the link. | **Description:** Opens the linked document in the full body of the window, removing all frames. | **Description:** Opens the linked document in the parent frame of the frame that contains the link. |
| **Usage:** This is the default behavior if the target attribute is not specified. | **Usage:** Useful when you want to break out of all frames and display the linked document in the entire window. | **Usage:** Used when dealing with nested frames. If the current frame is nested inside another frame, \_parent will open the document in the immediate parent frame. |
| **Example:** <a href="page.html" target="\_self">Link</a> | **Example:** <a href="page.html" target="\_top">Link</a> | **Example:** <a href="page.html" target="\_parent">Link</a> |

**<!DOCTYPE html>**

<!-- The <html> HTML element represents the root (top-level element) of an HTML document, so it is also referred to as the root element. All other elements must be descendants of this element. -->

**<html lang="en">**

<!-- The <head> HTML element contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets. -->

**<head>**

<!-- The <meta> HTML element represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> or <title>. This attribute defines the name of a piece of document-level metadata. It should not be set if one of the attributes itemprop, http-equiv or charset is also set. -->

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width,**

**initial-scale=1.0">**

<!-- The <link> HTML element specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things. -->

**<link rel="stylesheet" href="style.css">**

<!-- The <title> HTML element defines the document's title that is shown in a browser's title bar or a page's tab. It only contains text; tags within the element are ignored. -->

**<title>Colors page</title>**

**</head>**

<!-- The <body> HTML element represents the content of an HTML document. There can be only one <body> element in a document. -->

**<body>**

<!-- The <h1> to <h6> HTML elements represent six levels of section headings. <h1> is the highest section level and <h6> is the lowest. By default, all heading elements create a block-level box in the layout, starting on a new line and taking up the full width available in their containing block. -->

**<h1>formatting</h1>**

<!-- The <p> HTML element represents a paragraph. Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or first-line indentation, but HTML paragraphs can be any structural grouping of related content, such as images or form fields. -->

**<p>**

**This is my**

<!-- The <b> HTML element is used to draw the reader's attention to the element's contents, which are not otherwise granted special importance. This was formerly known as the Boldface element, and most browsers still draw the text in boldface. However, you should not use <b> for styling text or granting importance. If you wish to create boldface text, you should use the CSS font-weight property. If you wish to indicate an element is of special importance, you should use the <strong> element. -->

**<b>paragraph</b>**

**</p>**

**<p>**

**This is my**

<!-- The <i> HTML element represents a range of text that is set off from the normal text for some reason, such as idiomatic text, technical terms, taxonomical designations, among others. Historically, these have been presented using italicized type, which is the original source of the <i> naming of this element. -->

**<i>paragraph</i>**

**</p>**

**<p>**

**This is my**

<!-- The <strong> HTML element indicates that its contents have strong importance, seriousness, or urgency. Browsers typically render the contents in bold type. -->

**<strong>paragraph</strong>**

**</p>**

**<p>**

**This is my**

<!-- The <em> HTML element marks text that has stress emphasis. The <em> element can be nested, with each level of nesting indicating a greater degree of emphasis. -->

**<em>paragraph</em>**

**</p>**

<!-- The <ul> HTML element represents an unordered list of items, typically rendered as a bulleted list. -->

**<ul>**

<!-- The <li> HTML element is used to represent an item in a list. It must be contained in a parent element: an ordered list (<ol>), an unordered list (<ul>), or a menu (<menu>). In menus and unordered lists, list items are usually displayed using bullet points. In ordered lists, they are usually displayed with an ascending counter on the left, such as a number or letter. -->

**<li>b; strong</li>**

**<li>i; em</li>**

**<li>del; markquee</li>**

**<li>sub; sup</li>**

**<li>small</li>**

**</ul>**

<!-- The <marquee> HTML element is used to insert a scrolling area of text. You can control what happens when the text reaches the edges of its content area using its attributes. -->

**<marquee behavior="scroll" direction="right">Hello</marquee>**

<!-- The <sub> HTML element specifies inline text which should be displayed as subscript for solely typographical reasons. Subscripts are typically rendered with a lowered baseline using smaller text. -->

**<p>H<sub>2</sub>O</p>**

<!-- The <sup> HTML element specifies inline text which is to be displayed as superscript for solely typographical reasons. Superscripts are usually rendered with a raised baseline using smaller text. -->

**<p>12<sup>th</sup> standard</p>**

<!-- The <small> HTML element represents side-comments and small print, like copyright and legal text, independent of its styled presentation. By default, it renders text within it one font-size smaller, such as from small to x-small. -->

**<small>This is small content</small>**

<!-- The <del> HTML element represents a range of text that has been deleted from a document. This can be used when rendering "track changes" or source code diff information, for example. The <ins> element can be used for the opposite purpose: to indicate text that has been added to the document. -->

**<del>This is the content</del>**

**<h1 style="text-align: center; background-color: rgb(120, 184, 136); color: #000;">**

<!-- The rgb() functional notation expresses a color in the sRGB color space according to its red, green, and blue components. An optional alpha component represents the color's transparency.

R-RED, G-GREEN, B-BLUE, and A-ABSOLUTE value (rgb'a')

-->

**This tag is needs some styling**

**</h1>**

**<p>**

<!-- The <blockquote> HTML element indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation (see Notes for how to change it). A URL for the source of the quotation may be given using the cite attribute, while a text representation of the source can be given using the <cite> element. -->

**<blockquote>**

**Hello my name is saurabh**

**</blockquote>**

**</p>**

**<p>**

<!-- The <q> HTML element indicates that the enclosed text is a short inline quotation. Most modern browsers implement this by surrounding the text in quotation marks. This element is intended for short quotations that don't require paragraph breaks; for long quotations use the <blockquote> element. -->

**<q>**

**Hello my name is saurabh**

**</q>**

**</p>**

<!-- The <abbr> HTML element represents an abbreviation or acronym. When including an abbreviation or acronym, provide a full expansion of the term in plain text on first use, along with the <abbr> to mark up the abbreviation. This informs the user what the abbreviation or acronym means.

The optional title attribute can provide an expansion for the abbreviation or acronym when a full expansion is not present. This provides a hint to user agents on how to announce/display the content while informing all users what the abbreviation means. If present, title must contain this full description and nothing else.-->

**<abbr title="Hyper Text Markup Language">HTML</abbr>**

<!-- The <address> HTML element indicates that the enclosed HTML provides contact information for a person or people, or for an organization. -->

**<address>what to do</address>**

<!-- The <cite> HTML element is used to mark up the title of a cited creative work. The reference may be in an abbreviated form according to context-appropriate conventions related to citation metadata. -->

**<cite>hello saurabh</cite>**

**<br>**

<!-- The <bdo> HTML element overrides the current directionality of text, so that the text within is rendered in a different direction. -->

**<bdo dir="ltr">SAURABH KUMAR KANNOJIYA</bdo>**

**</body>**

**</html>**