HTML crash course

**0. what is HTML?**

**1. dev tools**

**2. cấu trúc file html**

file-html-structure.html

**3. comments trong html**

- mục đích

+) chú thích

+) vô hiệu hóa dòng code

**4. các thể html thông dụng**

|  |
| --- |
| - <h1> -> <h6>  - <img />  - <a />  - <ul>, <ol>, <li>  - <table>  - <button>  - <div>, <span> |

- An HTML is the building block of an HTML document

- format:

|  |  |  |
| --- | --- | --- |
| <tagname>Content </tagname>  vd: <div>, <span> | <tagname />  vd: <img />, <input /> | <tagname>  <tagname>Content<tagname>  </tagname> |

**5. attributes trong HTML**

1. **Syntax**: Attributes are specified within the opening tag of an element and consist of a name and a value, separated by an equals sign (**=**). For example: **<tagname attribute="value">**.
2. **Common Attributes**: HTML elements can have various attributes depending on their purpose. Some common attributes include **id**, **class**, **src**, **href**, **alt**, **title**, **style**, **target**, **width**, **height**, **type**, **name**, **value**, **placeholder**, **disabled**, **checked**, **required**, and many more.
3. **ID Attribute**: The **id** attribute provides a unique identifier for an element within the document. IDs must be unique within the document and are often used for JavaScript manipulation or CSS styling.
4. **Class Attribute**: The **class** attribute is used to define one or more classes for an element. Classes can be shared among multiple elements and are commonly used for styling purposes with CSS.
5. **Src Attribute**: The **src** attribute specifies the URL of the external resource, such as an image (**<img>**), script (**<script>**), or iframe (**<iframe>**).
6. **Href Attribute**: The **href** attribute is used to specify the URL of the linked resource, typically used with anchor (**<a>**) elements for hyperlinks.
7. **Alt Attribute**: The **alt** attribute provides alternative text for images (**<img>**) in case the image cannot be displayed. It is also used by screen readers for accessibility purposes.
8. **Title Attribute**: The **title** attribute provides additional information about the element, often displayed as a tooltip when the user hovers over the element.
9. **Style Attribute**: The **style** attribute allows inline CSS styles to be applied directly to an element.
10. **Data Attributes**: Custom attributes prefixed with **data-** can be used to store custom data for an element, which can be accessed by JavaScript.
11. **Boolean Attributes**: Some attributes are boolean, meaning they don't require a value to be specified. If present, the attribute is considered true. Examples include **disabled**, **checked**, and **readonly**.
12. **Form Attributes**: Attributes like **name**, **value**, **placeholder**, **disabled**, **checked**, **required**, etc., are commonly used within form elements (**<input>**, **<textarea>**, **<select>**, etc.) to define form behavior.

**6. semantic html**

Examples of semantic HTML elements include:

* **<header>**: Defines the header section of a document or a section.
* **<nav>**: Defines a set of navigation links.
* **<main>**: Specifies the main content of a document.
* **<article>**: Represents an independent piece of content within a document, such as a blog post or news article.
* **<section>**: Defines a section in a document, such as chapters, headers, footers, or any other thematic grouping of content.
* **<aside>**: Represents content that is tangentially related to the content around it, such as sidebars or pull quotes.

**7. <head />**

The **<head>** element in HTML is a container for metadata and other head elements, such as **<title>**, **<meta>**, **<link>**, **<style>**, **<script>**, and more. Here are some key points to know about the **<head>** element:

1. **Metadata**: The **<head>** element contains metadata about the HTML document, including information that isn't directly displayed on the page but is important for browsers and search engines.
2. **Title**: The **<title>** element inside the **<head>** defines the title of the HTML document, which appears in the browser's title bar or tab.
3. **Character Encoding**: The **<meta charset="UTF-8">** element specifies the character encoding for the document. UTF-8 is the most commonly used character encoding for web pages and supports a wide range of characters from various languages.
4. **Viewport Meta Tag**: The **<meta name="viewport" content="width=device-width, initial-scale=1.0">** element is used to control the viewport behavior on mobile devices. It ensures that the page is displayed properly across different screen sizes and devices by setting the width to the device's width and initial scale to 1.0.
5. **Other Meta Tags**: Other **<meta>** elements inside the **<head>** can include metadata such as description, keywords, authorship information, and more. These meta tags can be used for SEO (Search Engine Optimization) purposes.
6. **External Resources**: The **<link>** element inside the **<head>** is used to link external resources such as stylesheets, favicons, and web fonts.
7. **Internal Styles and Scripts**: The **<style>** and **<script>** elements inside the **<head>** can be used to embed CSS styles and JavaScript code directly into the HTML document.
8. **Search Engine Optimization (SEO)**: The content of the **<head>** element, including metadata and other tags, is crucial for SEO as search engines use this information to understand and rank the web page.
9. **block & inline element**
10. **Block-level Elements**:
    * Block-level elements typically start on a new line and take up the full width available to them, extending horizontally to the edges of their containing element (unless otherwise specified).
    * They create "blocks" of content on a web page.
    * Examples of block-level elements include **<div>**, **<p>**, **<h1>** to **<h6>**, **<ul>**, **<ol>**, **<li>**, **<header>**, **<footer>**, **<section>**, **<article>**, and **<nav>**.
11. **Inline Elements**:
    * Inline elements do not start on a new line and only take up as much width as necessary, allowing other elements to sit beside them horizontally.
    * They are typically used within block-level elements to style smaller parts of the content.
    * Examples of inline elements include **<span>**, **<a>**, **<strong>**, **<em>**, **<img>**, **<input>**, **<button>**, and **<br>**.

Here are some key differences between block-level and inline elements:

* **Layout**: Block-level elements create a block-level box, meaning they stack vertically on top of each other. Inline elements, on the other hand, flow within the text and do not disrupt the flow of content.
* **Width**: Block-level elements take up the full width available to them by default, while inline elements only occupy as much width as necessary for their content.
* **Line Breaks**: Block-level elements typically start on a new line, while inline elements do not force line breaks and can appear within a line of text.
* **Child Elements**: Block-level elements can contain both block-level and inline elements, while inline elements can only contain other inline elements or text.

It's important to note that the display behavior of elements can be altered using CSS. For example, you can change an inline element to display as a block-level element, or vice versa, using the **display** property in CSS (**display: block;** or **display: inline;**). This flexibility allows developers to create custom layouts and designs according to their specific needs.