**Giới thiệu HTML**

* HTML, viết tắt của Hyper Text Markup Language, là ngôn ngữ đánh dấu siêu văn bản.
* HTML cho phép chúng ta xây dựng cấu trúc các phần tử của trang web.
* HTML KHÔNG PHẢI là ngôn ngữ lập trình mà là ngôn ngữ đánh dấu, mục đích là tạo nên cấu trúc cho nội dung của trang web.
* Code HTML chỉ đơn giản là các thẻ lồng nhau chứa tất cả thông tin của trang web (văn bản, hình ảnh, video…).

Ví dụ: HTML định nghĩa cấu trúc. Một trang web có thể có nhiều cấu trúc HTML khác nhau tùy theo mục tiêu của từng trang.

**HTML: Nguyên tắc cơ bản**

**Một vài nguyên tắc về HTML:**

● Sử dụng cú pháp XML (các thẻ có thuộc tính, có thể chứa các thẻ khác bên trong).

<tag\_name> Nội dung </tag\_name>

● Lưu trữ tất cả thông tin sẽ hiển thị ra cho người dùng.

● Mỗi thẻ HTML có công dụng riêng và có các thuộc tính để tùy chỉnh riêng.

● Thông tin được lưu trữ theo cấu trúc dạng cây (các nút chứa các nút bên trong) được

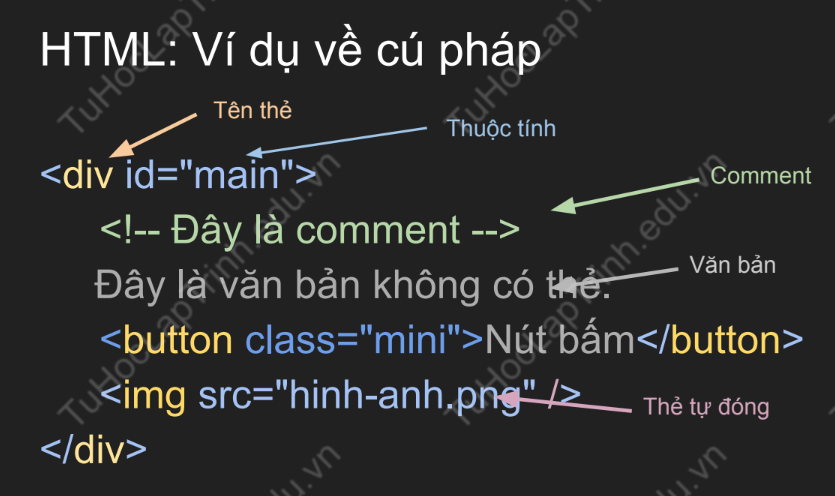
gọi là DOM (Document Object Model).

● HTML giúp phân loại dữ liệu theo từng cấu trúc riêng (title, section, form...) giúp máy

tính hiểu được nội dung trang web.

● HTML có cung cấp một số thẻ và thuộc tính để cài đặt hiển thị như màu sắc, font chữ,

đậm nhạt... nhưng đó không phải là nhiệm vụ chính. Việc cài đặt hiển thị là trách nhiệm của CSS.



**HTML: Những thẻ chính**

Mặc dù HTML có rất nhiều thẻ, nhưng 99% các trang web chỉ sử dụng các thẻ thông dụng.

Dưới đây là một số thẻ quan trọng thường dùng:

● **<div>:** Một khung chứa với các dữ liệu khác bên trong.

● **<img/>:** Một tấm ảnh.

● **<a>:** Một liên kết có thể nhấp để đi đến một URL khác.

● **<p>:** Một đoạn văn bản.

● **<h1>:** Một tiêu đề (h2, h3, h4 là các tiêu đề ít quan trọng hơn).

● **<input>:** Một ô nhập cho phép người dùng gõ vào thông tin.

● **<style>:** Để chèn các quy tắc CSS.

● **<script>:** Để thực thi Javascript.

**● <span>:** Một thẻ null không làm bất cứ điều gì.

Các thẻ sau đây có thể sẽ rất hữu ích trong một số trường hợp nhất định:

● **<button>:** Để tạo một nút bấm.

● **<audio>:** Để phát âm thanh.

● **<video>:** Để phát video.

● **<canvas>:** Để vẽ đồ họa từ Javascript.

● **<iframe>:** Để nhúng một trang web khác vào bên trong.

**HTML: Bao bọc thông tin**

* Chúng ta sử dụng các thẻ HTML để bao bọc các thông tin khác nhau tạo nên cấu trúc của trang web.
* Cấu trúc càng phong phú bạn càng dễ dàng tiếp cận và trình bày nó.
* Chúng ta có thể thay đổi các thông tin được hiển thị trên màn hình tùy thuộc vào các thẻ chứa nó, vì vậy đừng nên lo lắng về việc sử dụng quá nhiều thẻ.

**HTML: Định danh & Nhận dạng chung**

Thật tốt khi tất cả các thông tin được bao bọc đúng cách trong các thẻ để cung cấp cho nó một số ngữ nghĩa.

Chúng ta cũng có thể mở rộng ngữ nghĩa bằng cách thêm các thuộc tính bổ sung vào các thẻ:

● id: Định danh mã duy nhất cho thẻ này

● class: Cài đặt nhận dạng chung cho thẻ này

<div id="profile-picture" class="mini-image">...</div>

**CSS**

Cho phép chỉ định cách trình bày thông tin được lưu trữ trong HTML.

Cho phép kiểm soát tất cả các khía cạnh của hình ảnh và một số tính năng khác:

● Colors: Nội dung, nền, viền.

● Margins: Lề trong, lề ngoài.

● Position: Vị trí để đặt nó.

● Sizes: Chiều rộng, chiều cao.

● Behaviour: Thay đổi khi rê chuột vào.

CSS: Ví dụ

\*{

  color: blue; /\* Một comment \*/

  margin: 10px;

  font: 14px Tahoma;

}

Đoạn code sẽ thay đổi tất cả các thẻ trong web (‘\*’ có nghĩa là tất cả) thành màu blue

với phông chữ Tahoma có kích thước 14px, chừa lề xung quanh 10px.

**CSS: Cài đặt**

Đây là danh sách các cài đặt CSS phổ biến nhất:

● color: #FF0000; red; rgba(255,00,100,1.0); // Cài đặt màu theo nhiều cách

● background-color: red;

● background-image: url('file.png');

● font: 18px 'Tahoma';

● border: 2px solid black;

● border-top: 2px solid red;

● border-radius: 2px; // Bo tròn các góc

● margin: 10px; // Khoảng cách từ viền đến các yếu tố bên ngoài

● padding: 2px; // Khoảng cách từ viền đến các yếu tố bên trong

● width: 100%; 300px; 1.3em; // Cài đặt chiều rộng theo nhiều đơn vị

● height: 200px;

● text-align: center;

● box-shadow: 3px 3px 5px black;

● cursor: pointer;

● display: inline-block;

● overflow: hidden;

**CSS: Làm sao để thêm vào**

Có 3 cách để thêm CSS vào trang web:

● Chèn mã bên trong thẻ style

<style>

p { color: blue }

</style>

● Khai báo sử dụng một tệp CSS bên ngoài

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

● Sử dụng kiểu thuộc tính trên thẻ

  <p style="color: blue; margin: 10px"></p>

**CSS. Bộ chọn (Selectors)**

Điều gì sẽ xảy ra nếu chúng ta muốn thay đổi một thẻ cụ thể (chứ không phải tất cả các thẻ cùng loại)?

Selector chỉ định chính xác thành phần nào trong HTML để nhắm mục tiêu và để áp dụng các style (như màu sắc, kích thước, vị trí...).

Các selector thường nhắm đến một attribute như một id hoặc class hay nhắm đến các elements như <h1> hoặc <p>. Trong ví dụ:

* Selector dạng: .class như .intro sẽ select tất cả các elements có class là class="intro"
* Selector dạng: #id như #firstname sẽ select element có id là id="firstname"
* Selector dạng: element như p sẽ select tất cả các element <p>

Chúng ta có thể chỉ định các bộ chọn chính xác hơn bên cạnh tên của thẻ. Ví dụ theo class hoặc id:

p.intro {

  color: red;

  }

Cài đặt văn bản màu đỏ sẽ sẽ chỉ ảnh hưởng đến thẻ

  <p class="intro"></p>

**Các bộ chọn chính là:**

● Tên thẻ: Chỉ đơn giản sử dụng tên thẻ

○ p { ... } // Ảnh hưởng đến tất cả các thẻ <p>

● Chấm (.): Ảnh hưởng đến các thẻ có class đó

○ p.highlight { ... } // Ảnh hưởng đến tất cả các thẻ <p> có class="highlight"

● Dấu thăng (#): Chỉ ảnh hưởng đến duy nhất id đó

○ p#intro { ... } // Ảnh hưởng đến thẻ <p> có id="intro"

● Dấu hai chấm (:): Trạng thái hành vi (như rê chuột lên chẳng hạn)

○ p:hover { ... } // Ảnh hưởng đến thẻ <p> khi rê chu ột lên

● Cặp ngoặc vuông ([attr='value']): Các thẻ có thuộc tính attr với giá trị ‘value’

○ input[type="text"] {...} // Ảnh hưởng đến các ô nhập liệu

Bạn cũng có thể chỉ định các thẻ theo ngữ cảnh của nó, ví dụ các thẻ nằm bên trong các thẻ khớp với một bộ chọn. Chỉ cần tách các bộ chọn bằng một khoảng trắng:

div#main p.intro { ... }

Điều này sẽ ảnh hưởng đến các thẻ p của class intro nằm bên trong thẻ div của id main

<div id="main">

    <p class="intro">....</p> ← Ảnh hưởng đến cái này

    </div>

    <p class="intro">....</p> ← Nhưng không phải cái này

< /div>

Và bạn có thể kết hợp các bộ chọn để thu hẹp phạm vi nhiều hơn:

div#main.intro:hover { ... }

Sẽ áp dụng CSS cho thẻ div đồng thời có id main và class intro khi rê chuột lên.

Nếu bạn muốn chỉ định các phần tử là con trực tiếp của một phần tử khác, hãy sử dụng dấu >:

ul.menu > li { ... }

Cuối cùng, nếu bạn muốn cài đặt chung giá trị CSS cho nhiều phần tử, bạn có thể sử dụng dấu ,:

div, p { ... } ← Áp dụng cho tất cả thẻ div và thẻ p

**CSS: Hiển thị**

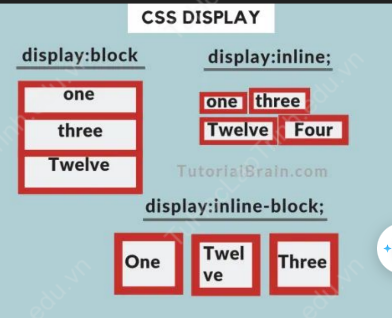
Bạn cần hiểu cách trình duyệt sắp xếp các phần tử trên màn hình.

Xem hướng dẫn này để nắm được các cách khác nhau mà một phần tử có thể được sắp xếp trên màn hình.

Bạn có thể thay đổi cách sắp xếp các phần tử bằng cách sử dụng display.

div { display: inline-block; }

Đừng quên tìm hiểu thêm về float.



**CSS: Cách tính kích thước**

Mặc định, bất kỳ chiều rộng và chiều cao nào được chỉ định cho một phần tử sẽ không tính đến lề của nó, vì vậy một div có chiều rộng 100px và lề 10px sẽ chiếm 120px trên màn hình, không phải 100px.

Đây có thể sẽ là một trong những nguyên nhân làm vỡ layout của bạn.

Bạn có thể thay đổi box model của phần tử để chiều rộng sử dụng đường viền ngoài cùng bằng box-sizing:

div { box-sizing: border; }

A box sizing chart with text

Description automatically generated

**Layout**

Một trong những phần khó nhất của CSS là hiểu bố cục trang web của bạn (cấu trúc bên

trong cửa sổ).

Theo mặc định, HTML có xu hướng đặt mọi thứ trong một cột, điều này không thật sự lý tưởng.

Đã có nhiều cách để giải quyết vấn đề này (table, fixed div, flex, grid, ...).

A screen shot of a computer

Description automatically generated

**Grid system**

Grid system là cách bố trí layout tốt nhất ở thời điểm hiện tại.

A screenshot of a computer screen

Description automatically generatedHãy xem bài này để có thể tạo cấu trúc trang web một các dễ dàng. A colorful rectangular shapes on a purple background

Description automatically generated

**Fullscreen**

Đôi khi chúng ta muốn có một div bao phủ toàn bộ màn hình (để tạo một ứng dụng web)

thay vì một trang web cuộn (giống các tài liệu thông thường hơn).

Trong trường hợp đó, hãy nhớ sử dụng tỷ lệ phần trăm để xác định kích thước của các

phần tử, nhưng hãy nhớ rằng tỷ lệ phần trăm các phần tử con sẽ được tính theo kích

thước của cha nó, do vậy bạn phải đặt kích thước phần tử <body> là 100%.



**Mẹo canh giữa**

Canh giữa các div đôi khi có thể rất khó, hãy sử dụng thủ thuật sau:

.horizontal-and-vertical-centering {

  display: flex;

  justify-content: center;

  align-items: center;

  }

[Giới thiệu HTML, CSS và Javascript - Chiến Binh Full Stack (tuhoclaptrinh.edu.vn)](https://fullstack.tuhoclaptrinh.edu.vn/gioi-thieu-html-css-va-javascript-102.html)

[Giới thiệu về HTML | DEVSNE](https://devsne.vn/learn/html-html-gioi-thieu-html)