

# Họ và tên Trần Huy Hải Anh

## Lớp 19-06

### HỌC PHẦN 1: NỀN TẢNG CĂN BẢN (Bài 1 - 2)

#### Bài 1: Giới thiệu về HTML & Các khái niệm cốt lõi

-**Html** ngôn ngữ đánh dấu tiêu chuẩn được sử dụng để **tạo cấu trúc** và **trình bày nội dung** trên **World Wide Web** (Mạng lưới Toàn cầu).

-Vai trò của HTML trong bộ ba **HTML-CSS-JavaScript** có thể được hình dung như sau:

- **HTML** (HyperText Markup Language) là **Cấu trúc (Structure)** hay **Bộ xương (Skeleton)**.
- **CSS** (Cascading Style Sheets) là **Hình thức/Phong cách (Presentation)** hay **Da/Quần áo (Skin/Clothes)**.
- **JavaScript** là **Hành vi/Tương tác (Behavior)** hay **Cơ bắp/Hệ thần kinh (Muscles/Nervous System)**.

-Trong HTML, một **Phần tử (Element)** thường bao gồm ba thành phần chính:

1. **Thẻ Mở (Opening Tag)**
2. **Nội dung (Content)**
3. **Thẻ Đóng (Closing Tag)**

#### 1. Thẻ Mở (Opening Tag)

- **Mục đích:** Bắt đầu định nghĩa một phần tử và xác định **loại** nội dung sắp tới.
- **Cấu trúc:** Được đặt trong dấu ngoặc nhọn (< và >) và chứa **tên thẻ**.
- *Ví dụ:* <p>, <h1>, <a>.
- **Đặc điểm bổ sung:** Thẻ mở có thể chứa **Thuộc tính (Attributes)** để cung cấp thêm thông tin cho trình duyệt về cách thẻ đó nên hoạt động hoặc hiển thị.
  - *Ví dụ:* <a href="https://google.com">.

#### 2. Nội dung (Content)

- **Mục đích:** Là thông tin thực tế mà người dùng sẽ thấy trên trang web.
- **Cấu trúc:** Là bất kỳ thứ gì nằm giữa thẻ mở và thẻ đóng. Nội dung có thể là văn bản, hoặc thậm chí là các phần tử HTML khác (lồng ghép).
- *Ví dụ:* Trong <h1>Xin chào</h1>, "Xin chào" là nội dung.

#### 3. Thẻ Đóng (Closing Tag)

- **Mục đích:** Kết thúc định nghĩa của phần tử.

- **Cấu trúc:** Giống như thẻ mở, nhưng có thêm dấu **gạch chéo (/)** ngay trước tên thẻ.
- **Ví dụ:** `</p>`, `</h1>`, `</a>`.

-Thuộc tính bao gồm ba phần:

1. **Tên Thuộc tính (Attribute Name):** Tên định danh thuộc tính mà bạn muốn thiết lập.
2. **Dấu bằng (=):** Dùng để gán giá trị.
3. **Giá trị Thuộc tính (Attribute Value):** Thông tin hoặc cài đặt cụ thể được đặt trong **dấu ngoặc kép** (" " hoặc ' ').

`<tên_thẻ tên_thuộc_tính="giá_trị_thuộc_tính">Nội dung</tên_thẻ>`

## Bài 2: Cấu trúc một file HTML chuẩn

### -<!DOCTYPE html>

Dòng này không phải là thẻ HTML mà là một khai báo (declaration) giúp trình duyệt biết đây là tài liệu HTML chuẩn phiên bản HTML5. Nó giúp trình duyệt hiển thị trang web đúng cách và theo chuẩn hiện đại nhất.

-Ba cặp thẻ `<html>`, `<head>`, và `<body>` là những phần tử cấu trúc cơ bản trong tài liệu HTML.

- `<html>` là thẻ gốc bao phủ toàn bộ nội dung trang web, là phần tử cha chứa các phần tử `<head>` và `<body>`. Thẻ này định nghĩa một tài liệu HTML và có thể có thuộc tính như ngôn ngữ (ví dụ: `lang="vi"`).
- `<head>` nằm trong thẻ `<html>`, chứa các siêu dữ liệu (metadata) như tiêu đề trang (`<title>`), thông tin charset, liên kết tới file CSS, các đoạn mã JavaScript, và các dữ liệu phụ trợ khác. Nội dung trong thẻ này không hiển thị trực tiếp trên trình duyệt nhưng rất quan trọng cho việc hiển thị và xử lý trang.
- `<body>` cũng nằm trong `<html>`, chứa tất cả nội dung hiển thị lên trình duyệt như văn bản, hình ảnh, video, liên kết, biểu mẫu, và các thành phần tương tác khác của trang web.

-Các thẻ quan trọng thường xuất hiện trong thẻ `<body>` của tài liệu HTML bao gồm:

- Các thẻ tiêu đề: `<h1>` đến `<h6>`, dùng để phân cấp tiêu đề với kích thước giảm dần, trong đó `<h1>` là tiêu đề quan trọng nhất, thường dùng cho tiêu đề chính của trang.
- `<p>`: thẻ đoạn văn dùng để chứa các đoạn văn bản.
- `<div>`: thẻ phân vùng, dùng để nhóm các phần tử lại với nhau để dễ dàng định dạng hoặc xử lý.

- "`<a>`": thẻ liên kết, giúp tạo liên kết đến trang khác hoặc phần tử khác.
- "`<img>`": thẻ hình ảnh để nhúng ảnh vào nội dung trang.
- Các thẻ danh sách: "`<ul>`", "`<ol>`", và "`<li>`" dùng cho danh sách không thứ tự và có thứ tự.
- "`<form>`": thẻ biểu mẫu, dùng để thu thập dữ liệu người dùng.
- Các thẻ tạo bảng: "`<table>`", "`<tr>`", "`<td>`", "`<th>`" để trình bày dữ liệu bảng.
- "`<header>`", "`<nav>`", "`<main>`", "`<footer>`": các thẻ HTML5 dùng để đánh dấu các phần quan trọng trong bố cục trang như phần đầu, menu điều hướng, nội dung chính và chân trang.

## **-cách tạo và chạy 1 fiel index.html**

### **B1**Tạo file index.html

- Mở một trình soạn thảo văn bản đơn giản như Notepad (Windows), TextEdit (Mac, chuyển về chế độ plain text) hoặc các trình soạn thảo mã nguồn như Visual Studio Code, Sublime Text.
- Tạo một file mới và lưu với tên là "index.html". Tên file bắt buộc phải có phần mở rộng ".html".

### **B2**Viết nội dung HTML cơ bản vào file index.html

### B3 Lưu file lại

### B4 Mở file index.html trong trình duyệt:

- Tìm đến file vừa tạo trên máy tính.
- Nhấp đúp chuột vào file, trình duyệt web mặc định sẽ mở và hiển thị nội dung trang.
- Hoặc trong trình duyệt, chọn "File" > "Open File" (hoặc tổ hợp phím Ctrl+O), rồi chọn file index.html.

**B5** Sửa đổi file sẽ được cập nhật khi làm mới trang trong trình duyệt (nhấn F5 hoặc nút tải lại). Đây là cách tạo và chạy một trang web cơ bản bằng file index.html mà không cần máy chủ hay phần mềm phức tạp nào.

## **HỌC PHẦN 2: XÂY DỰNG NỘI DUNG (Bài 3 - 7)**

### **Bài 3: Định dạng Văn bản**

Các thẻ tiêu đề (Headings) trong HTML nằm trong khoảng từ "<h1>" đến "<h6>", với ý nghĩa và cách sử dụng như sau:

- "<h1>" là thẻ tiêu đề quan trọng nhất, thường dùng làm tiêu đề chính của trang hoặc nội dung lớn nhất. Một trang chỉ nên có một thẻ "<h1>" để thể hiện chủ đề chính.
- "<h2>" dùng cho tiêu đề cấp hai, chia nhỏ nội dung chính thành các phần chính.
- "<h3>" là tiêu đề cấp ba, thường là tiêu đề phụ bên trong phần do "<h2>" quản lý.
- Các thẻ "<h4>", "<h5>", và "<h6>" là các tiêu đề cấp thấp hơn, dùng để chia nhỏ chi tiết hơn nữa các phần nội dung con.

Các thẻ tiêu đề này giúp phân cấp rõ ràng cấu trúc nội dung trong trang, không chỉ để trình bày mà còn hỗ trợ công cụ tìm kiếm (SEO) hiểu được cấu trúc và tầm quan trọng của các phần nội dung. Việc sử dụng hợp lý các thẻ tiêu đề tăng trải nghiệm người dùng và giúp tổ chức nội dung một cách logic.

─Thẻ đoạn văn trong HTML là thẻ "<p>". Thẻ này được dùng để định nghĩa một đoạn văn bản trong trang web.

Cách sử dụng thẻ đoạn văn:

- Thẻ mở là "<p>", thẻ đóng là "</p>".
- Nội dung đoạn văn nằm giữa hai thẻ này.
- Mỗi thẻ "<p>" tương ứng với một đoạn văn độc lập, giúp chia tách các phần nội dung khác nhau.
- Trình duyệt tự động tạo khoảng cách dòng (dòng trống) giữa các đoạn văn, giúp văn bản dễ đọc hơn.
- Các đoạn văn có thể chứa văn bản, liên kết, hình ảnh, hoặc các thẻ inline khác như "<strong>", "<em>", "<a>"...

─Thẻ Xuống Dòng: <br> (Break)

Thẻ <br> được sử dụng để tạo một **ngắt dòng (line break)**, giống như khi bạn nhấn phím **Enter** trong một văn bản.

## Cách hoạt động

- Nó chèn một dòng mới tại vị trí của thẻ.
- Thẻ `<br>` thường được dùng trong nội dung của một đoạn văn bản (`<p>`) khi bạn cần ép văn bản xuống dòng mà **không bắt đầu một đoạn văn mới** (vì bắt đầu đoạn văn mới sẽ tạo ra khoảng cách lớn hơn giữa các dòng).

### ─Thẻ Kẻ Ngang: `<hr>` (Horizontal Rule)

Thẻ `<hr>` tạo một **đường kẻ ngang (horizontal rule)** để chia nội dung và chỉ ra sự **thay đổi chủ đề** giữa các phần của trang web.

- Thẻ `<hr>` hiển thị dưới dạng một đường kẻ mỏng trên màn hình.
- Nó là một **phần tử cấp khối (block-level element)**, nghĩa là nó chiếm toàn bộ chiều rộng có sẵn và ngắt dòng trước và sau nó.
- Về mặt ngữ nghĩa, nó đại diện cho một **đường phân cách theo chủ đề (thematic break)**.

Thẻ	Tầm quan trọng (Ngữ nghĩa)	Hình thức Mặc định	Khi nào nên dùng?
<code>&lt;strong&gt;</code>	<b>Rất Quan trọng</b>	<b>In Đậm</b>	Đánh dấu thông tin cấp bách, nghiêm trọng.
<code>&lt;em&gt;</code>	<b>Nhấn Mạnh giọng điệu</b>	<i>In Nghiêng</i>	Thay đổi trọng âm hoặc ngữ điệu đọc.
<code>&lt;b&gt;</code>	<b>Không quan trọng</b>	<b>In Đậm</b>	Làm nổi bật mà không thêm ngữ nghĩa (ví dụ: từ khóa).
<code>&lt;i&gt;</code>	<b>Không quan trọng</b>	<i>In Nghiêng</i>	Thuật ngữ nước ngoài, tên sách, v.v. (theo quy ước).

## Bài 4: Chèn Liên kết (Links)

Thẻ `<a>` (viết tắt của **Anchor**) được sử dụng để tạo ra **siêu liên kết (hyperlink)**. Siêu liên kết cho phép người dùng nhấp vào và di chuyển đến một tài liệu, trang web, hoặc một vị trí khác trên cùng một trang.

---

**Thuộc Tính Bắt Buộc:** `href`

Thẻ `<a>` trở nên hữu ích nhờ thuộc tính **href** (Hypertext Reference).

- **Mục đích:** Chỉ định **đích đến (URL)** của liên kết

HTML

```
<a href="URL_đến_trang_đích">Văn bản liên kết</a>
```

### Cách hoạt động của `target="_blank"`

Khi bạn thêm thuộc tính `target="_blank"` vào một thẻ `<a>`, trình duyệt sẽ thực hiện như sau:

1. **Mở trong Tab/Cửa sổ Mới:** Thay vì điều hướng người dùng rời khỏi trang web hiện tại của bạn, liên kết sẽ được mở trong một **tab trình duyệt mới** hoặc một cửa sổ mới.
2. **Giữ người dùng lại:** Điều này giúp **giữ người dùng ở lại** trang web của bạn trong khi họ xem nội dung của liên kết đó, cải thiện trải nghiệm người dùng và thời gian họ ở lại trang (time on site).

HTML

```
<a href="URL_của_trang_đích" target="_blank">Văn bản liên kết</a>
```

## Liên kết đến một trang khác

- Dùng thẻ `<a>` với thuộc tính `href` chứa URL hoặc đường dẫn đến trang cần liên kết hoặc liên kết đến trang trong cùng thư mục

## Liên kết nội bộ trong cùng một trang (dùng ID)

- Để tạo liên kết nội bộ, ta cần thiết lập một phần tử có thuộc tính `id`
- Sau đó liên kết đến ID đó bằng thẻ `<a>` với `href` có dạng `#id`

## Bài 5: Chèn Hình ảnh (Images)

-Thẻ `<img>` trong HTML được dùng để chèn hình ảnh vào trang web.

Cú pháp cơ bản của thẻ `<img>`

```

```

## Các thuộc tính quan trọng

- `src`: Đường dẫn đến tệp hình ảnh (bắt buộc). Có thể là đường dẫn tuyệt đối hoặc tương đối.
- `alt`: Văn bản mô tả hình ảnh, hiển thị khi hình ảnh không tải được hoặc cho người dùng sử dụng trình đọc màn hình (bắt buộc).
- `width` và `height`: Kích thước hình ảnh (tùy chọn).
- `title`: Văn bản hiển thị khi người dùng di chuột lên hình ảnh (tùy chọn).

Thuộc tính `src` trong thẻ `<img>` dùng để chỉ định đường dẫn (URL) tới hình ảnh mà bạn muốn chèn vào trang web. Có thể sử dụng hai dạng đường dẫn phổ biến:

### 1. Ảnh online (đường dẫn tuyệt đối)

- Sử dụng một URL đầy đủ dẫn đến hình ảnh trên internet
- Trình duyệt sẽ tải hình ảnh từ địa chỉ URL này.

### 2. Ảnh trong cùng thư mục hoặc thư mục con (đường dẫn tương đối)

- Nếu ảnh nằm trong cùng thư mục với file HTML, chỉ cần khai báo tên ảnh:
- Nếu ảnh nằm trong thư mục con, khai báo đường dẫn tương đối đến thư mục đó, ví dụ ảnh trong thư mục `images`:
- Đường dẫn tương đối dựa trên vị trí file HTML hiện tại.

Thuộc tính `alt` (Alternative Text) trong thẻ `<img>` có vai trò rất quan trọng trong HTML và SEO.

### Tầm quan trọng của thuộc tính alt

- **Mô tả nội dung hình ảnh khi ảnh không hiển thị:** Khi trình duyệt gặp lỗi tải ảnh hoặc mạng yếu, văn bản trong `alt` sẽ được hiển thị thay thế để người dùng hiểu nội dung ảnh.

- **Hỗ trợ người dùng khuyết tật:** Các trình đọc màn hình dùng `alt` để đọc mô tả ảnh cho người mù hoặc giảm thị lực, giúp trang web tiếp cận đa dạng người dùng.
- **Tác động tích cực đến SEO:** Công cụ tìm kiếm như Google không thể "đọc" ảnh trực tiếp, nên `alt` giúp công cụ này hiểu ngữ cảnh, nội dung của hình ảnh, cải thiện thứ hạng tìm kiếm hình ảnh và trang web.
- **Tránh lỗi SEO và tăng trải nghiệm người dùng:** Việc thêm `alt` chuẩn giúp trang web chuyên nghiệp, dễ truy cập hơn và được đánh giá cao bởi các công cụ tìm kiếm.

Thuộc tính `width` và `height` trong thẻ `<img>` dùng để xác định chiều rộng và chiều cao hiển thị của hình ảnh trên trang web.

## Chi tiết thuộc tính width và height

- Các thuộc tính này thường được đặt bằng giá trị số nguyên, mặc định đơn vị là pixel (px).
- Nếu chỉ đặt một trong hai thuộc tính, hình ảnh sẽ được tự động điều chỉnh tỉ lệ sao cho không bị méo mó dựa trên tỉ lệ gốc của ảnh.
- Việc khai báo trước chiều rộng và chiều cao giúp trình duyệt dự trù không gian hiển thị ngay từ đầu, tránh hiện tượng nhảy layout khi ảnh tải xong, cải thiện trải nghiệm người dùng và hiệu suất trang.
- Ngoài ra, có thể dùng CSS (`style="width:300px; height:200px;"`) để kiểm soát kích thước linh hoạt hơn, ví dụ dùng phần trăm (%), nhưng thuộc tính trong thẻ `<img>` vẫn rất hữu ích để đặt kích thước mặc định tĩnh.
- Nếu không đặt kích thước, trang có thể tải ảnh đúng kích thước gốc, nhưng có thể làm thay đổi bố cục khi tải hình ảnh.

## Bài 6: Tạo Danh sách (Lists)

Trong HTML, để tạo danh sách không có thứ tự (unordered list) và các phần tử trong danh sách (list item), ta sử dụng hai thẻ sau:

### Thẻ `<ul>` (Unordered List)

- Dùng để định nghĩa một danh sách không có thứ tự, tức là các phần tử trong danh sách không cần sắp xếp theo thứ tự cụ thể.
- Trình duyệt sẽ hiển thị các phần tử trong danh sách dưới dạng các dấu chấm tròn (bullet points) theo mặc định.



## Thẻ `<li>` (List Item)

- Dùng để định nghĩa một phần tử trong danh sách.
- Mỗi phần tử `<li>` phải nằm bên trong một thẻ danh sách cha như `<ul>` hoặc `<ol>`

Danh sách có thứ tự trong HTML được tạo bằng thẻ `<ol>` (Ordered List) kết hợp với các phần tử danh sách là thẻ `<li>` (List Item).

## Thẻ `<ol>`

- Dùng để định nghĩa danh sách mà các phần tử có thứ tự rõ ràng (1, 2, 3,... hoặc A, B, C,...).
- Trình duyệt sẽ tự động đánh số thứ tự cho từng phần tử bên trong.

## Thẻ `<li>`

- Đại diện cho mỗi mục trong danh sách.
- Mỗi thẻ `<li>` phải nằm trong thẻ `<ol>` để tạo thành danh sách có thứ tự.-

-Cách tạo danh sách lồng nhau (Nested Lists) trong HTML rất đơn giản, bạn lồng một danh sách bên trong phần tử `<li>` của danh sách khác.

## Cách làm

- Trong một thẻ `<li>` của danh sách cha, bạn có thể thêm một thẻ danh sách mới `<ul>` hoặc `<ol>` làm danh sách con.
- Điều này cho phép tạo ra cấu trúc danh sách đa cấp, ví dụ danh sách có thứ tự bên trong danh sách không thứ tự hoặc ngược lại.

## Bài 7: Tạo Bảng biểu (Tables)

Các thẻ cơ bản trong HTML để tạo bảng gồm:

### 1. Thẻ `<tr>` (Table Row)

- Dùng để định nghĩa một hàng trong bảng.
- Mỗi hàng chứa các ô, có thể là ô tiêu đề hoặc ô dữ liệu.

- Thẻ này bao quanh các thẻ `<th>` hoặc `<td>`.

## 2. Thẻ `<th>` (Table Header)

- Đại diện cho một ô tiêu đề trong bảng.
- Thông thường các ô tiêu đề nằm trong hàng đầu tiên hoặc đầu cột để mô tả ý nghĩa của cột hoặc hàng.
- Nội dung trong `<th>` thường được hiển thị in đậm và căn giữa mặc định.

## 3. Thẻ `<td>` (Table Data)

- Đại diện cho một ô dữ liệu thông thường trong bảng.
- Các thẻ `<td>` nằm trong thẻ `<tr>`, chứa nội dung dữ liệu cụ thể tương ứng với hàng và cột.

## Ví dụ cấu trúc bảng đơn giản

Thuộc tính `colspan` và `rowspan` trong bảng HTML dùng để gộp các ô lại với nhau, tạo ra các ô rộng hơn theo chiều ngang hoặc chiều dọc.

## Thuộc tính `colspan` (Gộp cột)

- Được dùng trong thẻ `<td>` hoặc `<th>` để chỉ định số lượng cột mà ô đó sẽ chiếm chỗ ngang qua.
- Cú pháp: `colspan="n"`, trong đó `n` là số cột cần gộp.

## Thuộc tính `rowspan` (Gộp hàng)

- Dùng trong thẻ `<td>` hoặc `<th>` để chỉ định số lượng hàng mà ô đó sẽ trải dài xuống.
- Cú pháp: `rowspan="n"`, trong đó `n` là số hàng cần gộp.

## HỌC PHẦN 3: TƯƠNG TÁC VÀ CẤU TRÚC (Bài 8 - 11)

### Bài 8: Biểu mẫu (Forms) - Phần 1: Cơ bản

-Thẻ `<form>` trong HTML dùng để tạo biểu mẫu (form) thu thập dữ liệu từ người dùng.

## Thuộc tính `action`

- Xác định URL của trang web hoặc tập tin trên máy chủ sẽ nhận và xử lý dữ liệu form khi người dùng gửi (submit) biểu mẫu.
- Nếu không có thuộc tính này, dữ liệu form sẽ được gửi lại chính trang hiện tại.

## Thuộc tính `method`

- Xác định phương thức HTTP dùng để gửi dữ liệu form đến máy chủ.
- Có hai giá trị chính:
  - `get`: Dữ liệu được gửi dưới dạng tham số URL (nhìn thấy trên thanh địa chỉ), thích hợp với dữ liệu không nhạy cảm và khi muốn chia sẻ URL có dữ liệu.
  - `post`: Dữ liệu được gửi trong phần thân (body) của HTTP request, an toàn hơn và không giới hạn kích thước dữ liệu.

-Thẻ dùng để dán nhãn (label) cho ô nhập liệu trong HTML là thẻ `<label>`.

## Ưu điểm và chức năng của thẻ `<label>`

- Thẻ `<label>` định nghĩa nhãn cho các phần tử nhập liệu như `<input>`, `<textarea>`, `<select>`.
- Khi người dùng click vào nội dung của thẻ `<label>`, con trỏ chuột sẽ tự động chuyển đến ô nhập liệu tương ứng, giúp cải thiện trải nghiệm người dùng.
- Hỗ trợ truy cập cho người dùng dùng trình đọc màn hình, tăng khả năng tiếp cận trang web.

## Cách sử dụng thẻ `<label>`

- Dùng thuộc tính `for` trong thẻ `<label>` để liên kết với thẻ nhập liệu bằng cách đặt giá trị của `for` trùng với `id` của thẻ nhập liệu.

-Thẻ `<input>` trong HTML được sử dụng để tạo các ô nhập liệu trong biểu mẫu với nhiều loại dữ liệu khác nhau, được xác định bởi thuộc tính `type`.

## Các loại (type) phổ biến của thẻ `<input>`

`+ , type="text"`

Tạo ô nhập liệu dạng văn bản một dòng, dùng cho nhập tên, địa chỉ, v.v.

`+ , type="password"`

Tạo ô nhập liệu mật khẩu, các ký tự nhập vào sẽ được che dấu.

`+ , type="email"`

Tạo ô nhập liệu email, giúp kiểm tra định dạng email khi nhập.

`+ , type="submit"`

Tạo nút bấm gửi dữ liệu form tới server. Khi nhấn nút này, biểu mẫu sẽ được gửi đi.

## Bài 9: Biểu mẫu (Forms) - Phần 2: Nâng cao

Các loại phổ biến khác của thẻ `<input>` trong HTML gồm:

- `type="checkbox"`  
Tạo ô vuông cho phép người dùng chọn hoặc bỏ chọn nhiều lựa chọn độc lập.
- `type="radio"`  
Tạo nút tròn cho phép người dùng chọn một giá trị duy nhất trong một nhóm các lựa chọn. Các input radio cùng `name` thuộc một nhóm.
- `type="date"`  
Tạo ô nhập dữ liệu cho phép chọn ngày tháng qua bộ chọn ngày tích hợp của trình duyệt.

Ô nhập liệu nhiều dòng trong HTML được tạo bằng thẻ `<textarea>`.

### Đặc điểm của thẻ `<textarea>`

- Cho phép người dùng nhập văn bản dài và xuống dòng nhiều lần, khác với thẻ `<input>` chỉ nhập một dòng.
- Có thẻ mở và thẻ đóng, nội dung giữa hai thẻ là văn bản mặc định hiển thị trong ô nhập liệu.
- Kích thước có thể điều chỉnh bằng các thuộc tính:
  - `rows`: số dòng hiển thị.
  - `cols`: số ký tự chiều rộng.

Nút bấm trong HTML thường được tạo bằng các thẻ sau:

## 1. Thẻ `<button>`

- Dùng để tạo nút bấm có thể chứa văn bản hoặc hình ảnh bên trong.
- Có thể đặt thuộc tính `type` là:
  - `button`: nút bấm bình thường, thường dùng với JavaScript cho các hành động tùy chỉnh.
  - `submit`: nút bấm gửi dữ liệu form.
  - `reset`: nút bấm để đặt lại dữ liệu form về trạng thái ban đầu.

## 2. Thẻ `<input>` với `type` liên quan đến nút

- `type="submit"`: tạo nút gửi form.
- `type="button"`: tạo nút bấm bình thường không gửi form, dùng cho tương tác JavaScript.
- `type="reset"`: tạo nút đặt lại form.

-Danh sách chọn (Dropdown) trong HTML được tạo bằng thẻ `<select>` kết hợp với các thẻ `<option>` để định nghĩa các lựa chọn bên trong.

## Thẻ `<select>`

- Được sử dụng để tạo danh sách thả xuống, người dùng có thể chọn một hoặc nhiều tùy chọn từ danh sách.
- Phổ biến trong các biểu mẫu để hạn chế lựa chọn, giúp giao diện gọn gàng hơn so với nhiều ô nhập liệu riêng biệt.
- Thuộc tính phổ biến:
  - `name`: tên của danh sách, dùng để gửi dữ liệu sau khi form submit.
  - `id`: liên kết với thẻ `<label>` để tạo nhãn cho dropdown.
  - `multiple`: cho phép chọn nhiều mục cùng lúc.
  - `size`: số lượng mục hiển thị cùng lúc khi không dùng thả xuống.

## Thẻ `<option>`

- Định nghĩa từng mục lựa chọn trong dropdown.
- Thuộc tính `value` để đặt giá trị gửi đi khi lựa chọn được chọn.
- Thuộc tính `selected` để mặc định lựa chọn ban đầu.

## Bài 10: HTML Ngữ nghĩa (Semantic HTML)

Các thẻ "vô nghĩa" dùng để nhóm nội dung trong HTML thường là:

## Thẻ `<div>`

- Là thẻ cấp khối (block-level) không mang nghĩa ngữ nghĩa cụ thể nào.
- Được dùng để nhóm các phần tử HTML khác lại với nhau theo khối, phục vụ mục đích bố cục, tạo khung, hoặc áp dụng style chung qua CSS.
- Không cung cấp thông tin ngữ nghĩa cho nội dung, chỉ giúp tổ chức trang web về mặt cấu trúc trình bày.

## Thẻ `<span>`

- Là thẻ nội tuyến (inline-level) cũng không mang nghĩa ngữ nghĩa.
- Dùng để nhóm các phần tử hoặc đoạn văn bản nhỏ trong cùng một dòng nhằm áp dụng style hoặc thao tác riêng biệt.
- Thường dùng để chỉnh sửa hoặc đánh dấu một phần nhỏ nội dung trong khối văn bản lớn.

Những thẻ này không giúp trình duyệt hay bộ máy tìm kiếm hiểu ý nghĩa nội dung, nên được gọi là thẻ "vô nghĩa" (non-semantic). Cần kết hợp chúng với các thẻ ngữ nghĩa như `<header>`, `<section>`, `<article>`, `<nav>` để xây dựng trang web có cấu trúc rõ ràng và thân thiện với SEO cũng như người dùng.

Các thẻ ngữ nghĩa (semantic tags) trong HTML giúp định nghĩa rõ ràng cấu trúc và ý nghĩa của nội dung trên trang web, hỗ trợ trình duyệt, công cụ tìm kiếm và người dùng hiểu được từng phần nội dung. Dưới đây là giới thiệu các thẻ ngữ nghĩa phổ biến:

- `<header>`  
Đại diện cho phần đầu trang hoặc đầu của một phần nội dung. Thường chứa tiêu đề, logo, hoặc thanh điều hướng chính.
- `<nav>`  
Định nghĩa khu vực chứa các liên kết điều hướng, như menu chính hoặc menu phụ của trang web.
- `<main>`  
Đại diện cho phần nội dung chính của trang. Mỗi trang chỉ có một thẻ `<main>` để tập trung nội dung quan trọng nhất.
- `<article>`  
Đại diện cho một nội dung độc lập và tự chứa, có thể là một bài báo, một blog post, hoặc tin tức.

- **<section>**  
Là phần nội dung có liên quan chặt chẽ, thường có tiêu đề, dùng để chia nhỏ nội dung thành các phần.
- **<aside>**  
Đại diện cho nội dung phụ hoặc bổ sung như sidebar, quảng cáo, hoặc chú thích liên quan nhưng không phải nội dung chính.
- **<footer>**  
Là phần chân trang hoặc chân một phần nội dung, thường chứa thông tin về tác giả, bản quyền, liên kết bổ sung.

Sử dụng HTML ngữ nghĩa mang lại nhiều lợi ích, đặc biệt với SEO và khả năng tiếp cận:

## 1. Tối ưu SEO

- Các công cụ tìm kiếm như Google dựa vào cấu trúc ngữ nghĩa để hiểu nội dung trang rõ ràng và chính xác hơn.
- Thẻ ngữ nghĩa giúp xác định các phần quan trọng như tiêu đề, điều hướng, nội dung chính, bài viết, chân trang, từ đó nâng cao thứ hạng tìm kiếm.
- Cấu trúc HTML rõ ràng giúp công cụ lập chỉ mục hiệu quả, đồng thời hỗ trợ tạo rich snippets và featured snippets thu hút người dùng.

## 2. Cải thiện khả năng tiếp cận (Accessibility)

- Các trình đọc màn hình và thiết bị hỗ trợ người khuyết tật dựa vào thẻ ngữ nghĩa để diễn giải nội dung trang.
- Sử dụng thẻ phù hợp giúp người dùng khiếm thị hoặc gặp khó khăn về nghe nhìn có trải nghiệm truy cập tốt hơn.

## 3. Mã nguồn rõ ràng, dễ bảo trì

- Cấu trúc semantic giúp lập trình viên và nhà phát triển dễ hiểu bố cục và chức năng từng phần trong mã nguồn.
- Việc bảo trì, nâng cấp và tối ưu trang web trở nên đơn giản, nhanh chóng hơn.