

Website cơ bản - 02

Nguyễn Anh Tuấn

KTECH
COLLEGE



Nội dung bài giảng

1 HTML

2 CSS

3 Javascript

4 HTTP Method

HTML

HTML

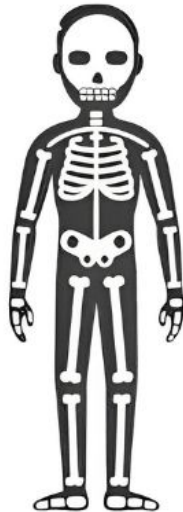
- **HTML (HyperText Markup Language**, hay còn gọi là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như **CSS** và các ngôn ngữ kịch bản giống như **JavaScript**.
- Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện.
- HTML mô tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.



HTML



HTML the Skeleton



CSS



CSS the Skin



JS



Javascript the Brain

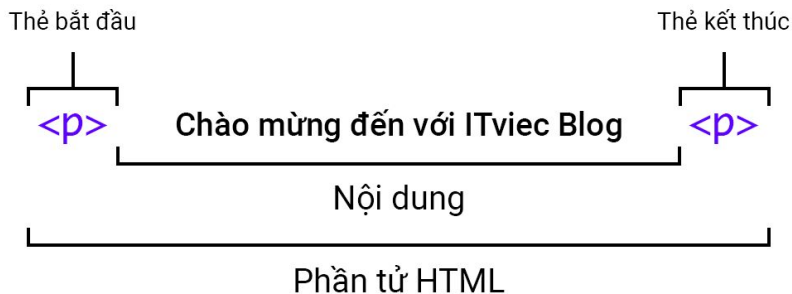


HTML

- **HTML** có chức năng giúp người dùng có thể định dạng, thiết kế cấu trúc các thành phần của một trang web hay ứng dụng, bao gồm tiêu đề (headings), liên kết (links), bảng (table), hình ảnh, đoạn văn bản và nhiều yếu tố khác.
- Một tài liệu **HTML** được hình thành từ các phần tử HTML (**HTML Elements**) và được quy định bằng các cặp thẻ (**tags và attribute**). Các cặp thẻ này nhận biết bởi một dấu ngoặc nhọn và bao gồm thẻ mở và thẻ đóng. Ví dụ, tạo một đoạn văn bản bằng cặp tag mở **<p>** và đóng **</p>**:

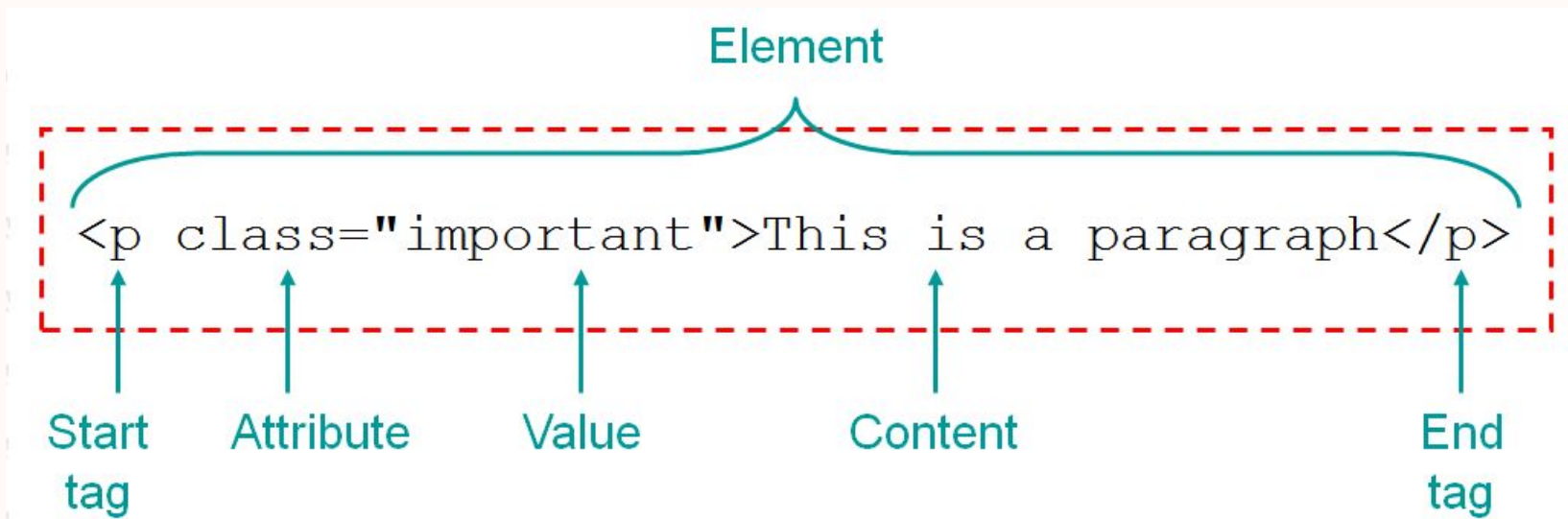
<p>Cách thêm văn bản trong HTML**</p>**

<p>HTML là gì?**</p>**



HTML

- Các thẻ **HTML** cũng có các thuộc tính hay đặc trưng có thể cung cấp tác dụng hữu ích như định dạng, style,... Một thuộc tính (attribute) được sử dụng để xác định đặc trưng của phần tử HTML và được đặt bên trong phần tử đó.



Công cụ lập trình HTML

- Có rất nhiều IDE hỗ trợ lập trình HTML, trong đó có 2 công cụ gọn nhẹ và phổ biến:
 - **Notepad++**
 - **Ưu điểm**
 - Nhỏ gọn.
 - Load nhanh và chạy không tốn nhiều tài nguyên hệ thống.
 - Tự động hoàn thành các từ và hàm.
 - Tùy chọn plug-in để mở rộng chức năng.
 - **Nhược điểm**
 - Có thể ít hữu ích hơn đối với các ngôn ngữ như Java.
 - **Visual studio code**
 - **Ưu điểm**
 - Miễn phí và Mã nguồn mở.
 - Đa dạng trong Tích hợp ngôn ngữ và Frameworks.
 - Hiệu Suất Tốt và Tiêu Thụ Ít Tài Nguyên Hệ Thống.
 - **Nhược điểm**
 - Không phải là một IDE đầy đủ cho các dự án lớn và phức tạp.
 - Cần cài đặt tiện ích mở rộng cho một số tính năng cụ thể.

Công cụ lập trình HTML

- Ngoài các IDE hỗ trợ lập trình HTML, lập trình viên có thể tham khảo thêm trang web:
<https://www.w3schools.com/html/>
 - **Ưu điểm:**
 - Dễ làm quen, tiếp cận.
 - Công cụ test trực quan dựa trên code nhập vào.
 - Nhiều ví dụ tham khảo.
 - Rất cơ bản cho những người mới bắt đầu mới HTML, CSS và Javascript.
 - **Khuyết điểm:**
 - Không hỗ trợ học tập nâng cao kiến thức chuyên sâu.



Cấu trúc của một trang HTML

<!-- DOCTYPE để khai báo ra version của chuẩn html -->

<!-- html ở đây ngầm định là html5, phiên bản mới nhất thời điểm hiện tại -->

<!DOCTYPE html>

<!-- Toàn bộ nội dung của 1 trang web sẽ được đặt trong thẻ <html></html> -->

<html>

<!-- Phần header giống như bì thư, có địa chỉ, thông tin người gửi người nhận, tiêu đề, ... -->

<head>

<title>Page Title</title>

</head>

<!-- Phần body là nội dung bức thư, cũng là nội dung sẽ hiển thị trực tiếp trên website -->

<body>

<h1>This is a Heading</h1>

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

</body>

</html>

Cấu trúc của một trang HTML

- **Lưu ý:**

- Hiện nay các trình duyệt đã hỗ trợ tự động tạo thẻ `<html>`, `<head>`, `<body>`, nên không bắt buộc phải viết đủ cấu trúc của 1 trang HTML, chỉ cần viết nội dung bên trong body.
- Tuy nhiên, nên viết đầy đủ cấu trúc để tạo ra được 1 trang web hoàn chỉnh, thực tế. Việc không viết đủ cấu trúc chỉ nên dành cho mục đích học tập.



Một số thẻ phổ biến

- Thẻ **<div>** (division)
 - Thể hiện 1 vùng hay 1 không gian trong trang web.

```
<div style="background-color:#FFF4A3; width:150px; height:50px">
```

```
<h2>London</h2>
```

```
</div>
```

```
<div style="background-color:#FFC0C7; width:200px; height:60px">
```

```
<h2>Oslo</h2>
```

```
</div>
```

```
<div style="background-color:#D9EEE1; width:250px; height:70px; display:inline;">
```

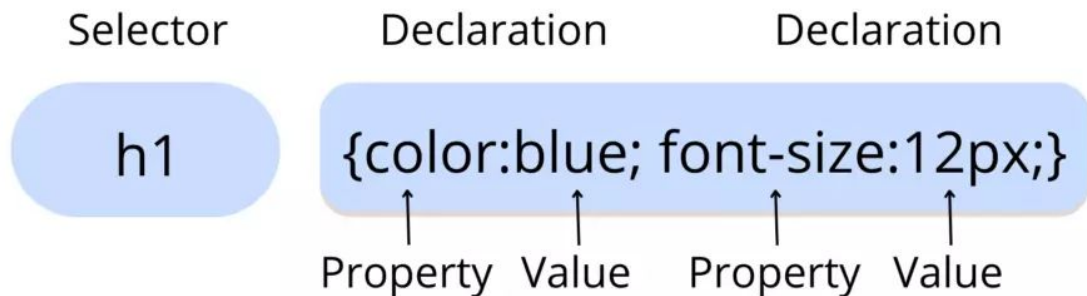
```
<h2>Rome</h2>
```

```
</div>
```

CSS

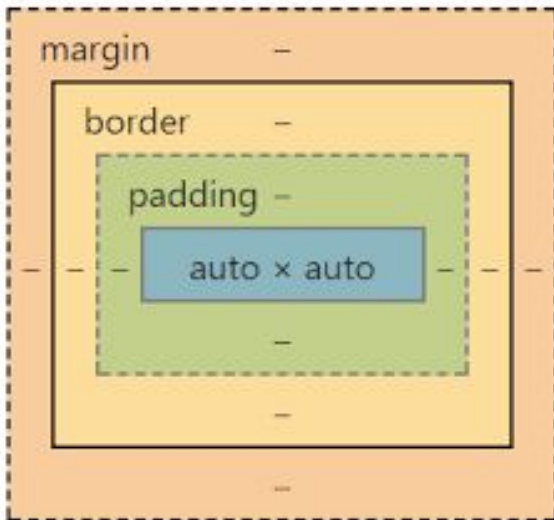
CSS

- **CSS** là viết tắt của **Cascading Style Sheets**, là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML).
- Phương thức hoạt động của **CSS** là tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.
- Mỗi tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.



Bố cục và cấu trúc một đoạn CSS

- Bố cục CSS thường chủ yếu dựa vào hình hộp (**box model**) và mỗi hộp đều chiếm những khoảng trống trên website với các thuộc tính như:
 - **Padding:** Gồm không gian xung quanh nội dung (ví dụ: xung quanh đoạn văn bản).
 - **Border:** Là đường viền nằm ngay bên ngoài phần đệm.
 - **Margin:** Là khoảng cách xung quanh bên ngoài của phần tử.

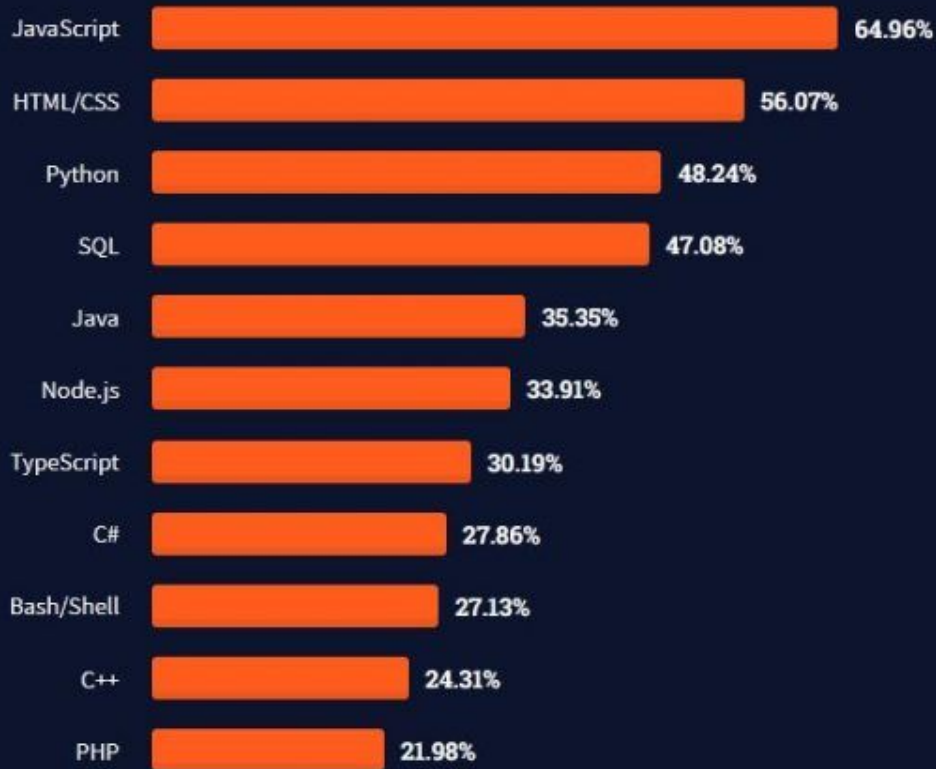


Cách tổ chức CSS

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #body {
        background-color: powderblue;
      }
      .b1 {
        background-color: white;
      }
    </style>
  </head>
  <body id="body">
    <h1 class="b1">This is a heading</h1>
    <p class="b1">This is a paragraph.</p>
  </body>
</html>
```


Javascript

Javascript



Javascript

- **JavaScript** là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web.
- Là ngôn ngữ kịch bản phía máy khách, **JavaScript** là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của **JavaScript**.



Javascript

```
<!DOCTYPE html>
<html>
  <head></head>
  <body></body>
  <foot>
    <script>
      var ten = "what the hell?"
      document.write(ten)
    </script>
  </foot>
</html>
```

Javascript

```
<!DOCTYPE html>
<html>
  <head></head>
  <body></body>
  <foot>
    <script>
      <!-- input -->
      var ten = window.prompt("Nhập tên vào đây: ")
      <!-- xử lý logic -->
      ten = "Tôi là " + ten
      <!-- output -->
      document.write(ten)
    </script>
  </foot>
</html>
```

Javascript

```
<!DOCTYPE html>
<html>
  <head></head>
  <body></body>
  <foot>
    <script>
      var tuoi = window.prompt("Nhập tuổi vào đây: ")
      if (tuoi <= 18) {
        document.write("Chưa đủ tuổi đi nghỉ vụ")
      } else {
        document.write("Đã đủ tuổi đi nghỉ vụ")
      }
    </script>
  </foot>
</html>
```

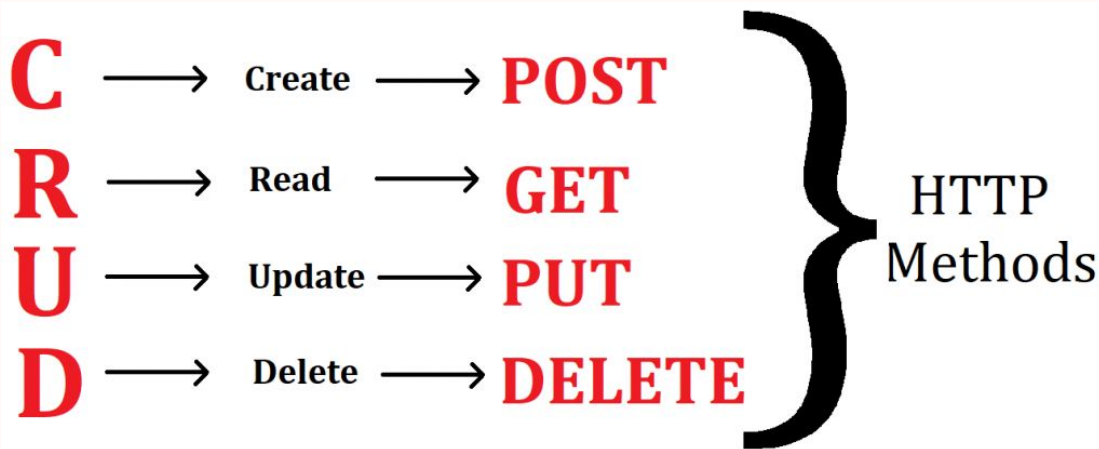
Javascript

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <p id="demo"></p>
  </body>
  <foot>
    <script>
      var tuoi = window.prompt("Nhập tuổi vào đây: ")
      if (tuoi <= 18) {
        document.getElementById("demo").innerHTML = "Chưa đủ tuổi đi nghĩa vụ"
      } else {
        document.getElementById("demo").innerHTML = "Đã đủ tuổi đi nghĩa vụ"
      }
    </script>
  </foot>
</html>
```

HTTP Method

HTTP Method

- Các phương thức HTTP còn được gọi là HTTP verbs, chịu trách nhiệm xác định các hành động có thể được thực hiện trên các tài nguyên được xác định bởi mã định danh tài nguyên URI(Uniform Resource Identifiers)
- HTTP định nghĩa 1 số phương thức, mỗi phương thức phục vụ một mục đích cụ thể trong việc xử lý các yêu cầu(request) và phản hồi(response) tài nguyên.
- Các phương thức HTTP phổ biến nhất bao gồm: **GET, POST, PUT, DELETE**, PATCH, HEAD, OPTIONS, CONNECT và TRACE.



HTTP Method

- **HTTP GET:**

- Phương thức này dùng để lấy tài nguyên từ server.

GET /api/users?id=123

- **HTTP PUT:**

- Phương thức này được dùng để cập nhật hoặc tạo một tài nguyên(resource) trên server, các request giống nhau sẽ cập nhật cùng một tài nguyên.

PUT /api/users/123

Content-Type: application/json

```
{  
  "name": "Updated Name",  
  "email": "updated@example.com"  
}
```

HTTP Method

- **HTTP POST:**

- Phương thức này thường được dùng để gửi dữ liệu cần xử lý đến máy chủ.
- Phương thức POST không thêm dữ liệu vào URL mà gửi dữ liệu trong request body. Điều này cho giúp nó phù hợp để gửi dữ liệu lớn hoặc dữ liệu có thông tin nhạy cảm.
- Phương thức POST được sử dụng khi muốn gửi 1 số dữ liệu đến máy chủ, ví dụ như cập nhật thông tin, gửi dữ liệu của 1 form, v.v. Trong trường hợp tạo mới, nếu gửi 2 request POST giống nhau thì tài nguyên sẽ được tạo 2 lần trên server.

POST /api/users

Content-Type: application/json

```
{  
  "name": "John Doe",  
  "email": "john@example.com"  
}
```

HTTP Method

- **HTTP DELETE:**

- Phương thức này thường được dùng để xóa tài nguyên trên server, nhiều request giống nhau thì sẽ xóa cùng 1 tài nguyên. Ví dụ: bạn gửi nhiều request xóa user theo ID bằng 123 thì tất cả các request này chỉ thực hiện việc xóa với user có ID = 123.

DELETE /api/users/123

- **HTTP PATCH:**

- Phương thức này được dùng để sửa đổi 1 phần(partial) cho tài nguyên. Phương thức này hiệu quả hơn PUT khi chỉ cập nhật các trường cụ thể của resource.

PATCH /api/users/123

Content-Type: application/json

```
{  
  "email": "newemail@example.com"  
}
```

ROAD TO KOREA

Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx