TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

Logo

Description automatically generated

**BÀI TẬP LỚN**

**MÔN: CÔNG NGHỆ WEB**

**NHÓM 5: XÂY DỰNG TRANG WEB XEM PHIM**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | Nguyễn Thị Hạnh |
| Thành viên nhóm: | Bùi Mạnh Hùng  Đỗ Thông Minh  Lê Bá Vũ |

Contents

[Chương I : Tổng quan về Công nghệ Web 3](#_Toc146642502)

[1.1 Lịch sử và phát triển : 3](#_Toc146642503)

[1.2 Các yếu tố quan trọng : 4](#_Toc146642504)

[1.3 Phát triển Web Fornt-End và Back-End : 4](#_Toc146642505)

[1.4 Công nghệ liên quan : 4](#_Toc146642506)

[1.5 Bảo mật và quyền riêng tư : 5](#_Toc146642507)

[1.6 Cộng đồng và tài nguyên : 5](#_Toc146642508)

[Chương II : Giới thiệu về HTML 5](#_Toc146642509)

[2.1 Tổng quan về HTML : 5](#_Toc146642510)

[2.2 Hướng dẫn tạo một tệp HTML cơ bản : 6](#_Toc146642511)

[2.3 Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết trong HTML : 7](#_Toc146642512)

[2.4 Hướng dẫn cú pháp và các phần tử HTML cơ bản : 9](#_Toc146642513)

[2.4.1 Cú pháp cơ bản của HTML : 9](#_Toc146642514)

[2.4.2 Các thành phần HTML cơ bản : 9](#_Toc146642515)

[Chương III : Cơ bản về CSS : 11](#_Toc146642516)

[3.1 Tổng quan về CSS : 11](#_Toc146642517)

[3.2 Hướng dẫn thêm kiểu cho trang web bằng CSS : 11](#_Toc146642518)

[3.2.1 Tạo tệp CSS : 11](#_Toc146642519)

[3.2.2 Liên kết tệp CSS với trang HTML : 11](#_Toc146642520)

[3.2.3 Viết mã CSS : 11](#_Toc146642521)

[3.3 Tạo các lớp và áp dụng bằng cách sử dụng lựa chọn : 12](#_Toc146642522)

[3.3.1 Tạo lớp : 12](#_Toc146642523)

[3.3.2 Sử dụng lựa chọn bằng lớp trong CSS : 12](#_Toc146642524)

[3.3.3 Áp dụng lớp vào nhiều phần tử : 12](#_Toc146642525)

[3.3.4 Sử dụng nhiều lớp cho một phần tử : 13](#_Toc146642526)

[3.3.5 Áp dụng bằng các sử dụng lựa chọn : 13](#_Toc146642527)

[3.3.6 Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc độ rộng và chiều cao : 13](#_Toc146642528)

[Chương IV : JavaScript cơ bản : 14](#_Toc146642529)

[4.1 Giới thiệu cú pháp JavaScript cơ bản : 14](#_Toc146642530)

[4.1.1 Biến và kiểu dữ liệu : 14](#_Toc146642531)

[4.1.2 Câu lệnh và biểu thức : 15](#_Toc146642532)

[4.1.3 Điều khiển luồng : 15](#_Toc146642533)

[4.1.4 Hàm : 16](#_Toc146642534)

[4.1.5 Đối tượng : 16](#_Toc146642535)

[4.1.6 Sự kiện : 16](#_Toc146642536)

[4.1.7 Bất đồng bộ và AJAX : 16](#_Toc146642537)

[4.1.8 Các Thư Viện và Framework : 17](#_Toc146642538)

[4.2 Hướng dẫn tạo biến, hàm và sử dụng sự kiện : 17](#_Toc146642539)

[4.2.1 Tạo biến : 17](#_Toc146642540)

[4.2.2 Tạo hàm : 17](#_Toc146642541)

[4.2.3 Sử dụng sự kiện : 18](#_Toc146642542)

[4.3 Ví dụ đơn giản hiển thị thông báo khi được ấn nút trong JavaScript : 19](#_Toc146642543)

[Chương V : HTML Forms và Input Elements : 20](#_Toc146642544)

[5.1 Tổng quan về HTML Forms và Input Elements : 20](#_Toc146642545)

[5.2 Từng bước tạo biểu mẫu đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, nút gửi : 20](#_Toc146642546)

[5.3 Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu để kiểm tra dữ liệu hợp lệ : 21](#_Toc146642547)

[CHƯƠNG VI : Quy trình phăt triển web. 22](#_Toc146642548)

[6.1 Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web. 22](#_Toc146642549)

[6.1.1 Lập kế hoạch (Planning): 22](#_Toc146642550)

[6.1.2 Thiết kế (Design): 22](#_Toc146642551)

[6.1.3 Phát triển (Development): 23](#_Toc146642552)

[6.1.4 Kiểm tra và đánh giá (Testing and Evaluation): 23](#_Toc146642553)

[6.1.5 Triển khai (Deployment): 23](#_Toc146642554)

[6.1.6 Quản lý và bảo trì (Management and Maintenance): 23](#_Toc146642555)

[6.2 Một số ví dụ về các bước trong quy trình phát triển web: 24](#_Toc146642556)

[6.2.1 Phân tích yêu cầu (Requirements Analysis): 24](#_Toc146642557)

[6.2.2 Thiết kế giao diện (UI/UX Design): 24](#_Toc146642558)

[6.2.3 Phát triển (Development): 24](#_Toc146642559)

[6.2.4 Kiểm tra và Kiểm tra (Testing and Quality Assurance): 24](#_Toc146642560)

[6.2.5 Triển khai (Deployment): 25](#_Toc146642561)

# Chương I : Tổng quan về Công nghệ Web

## **1.1 Lịch sử và phát triển**

Lĩnh vực công nghệ web đã phát triển rất nhanh kể từ khi World Wide Web (WWW) được phát minh bởi Tim Berners-Lee vào những năm 1990. Web ban đầu chỉ là một nền tảng dành cho việc chia sẻ văn bản, nhưng nó đã tiến hóa thành một môi trường đa phương tiện phức tạp hơn với ứng dụng web động, thương mại điện tử, truyền thông xã hội và nhiều tiện ích khác.

## **1.2 Các yếu tố quan trọng**

* JavaScript: Là ngôn ngữ lập trình dùng để tạo các tính năng tương tác và động trên trang web.
* Server-Side Programming: Các ngôn ngữ như PHP, Python, Ruby và Java thường được sử dụng để phát triển ứng dụng web phía máy chủ.
* Cơ Sở Dữ Liệu: Hệ quản trị cơ sở dữ liệu (DBMS) như MySQL, PostgreSQL, MongoDB, và Redis được sử dụng để lưu trữ và quản lý dữ liệu trang web.
* API (Application Programming Interface): API cho phép các ứng dụng tương tác với nhau và chia sẻ dữ liệu.
* HTML (HyperText Markup Language): Là ngôn ngữ đánh dấu cơ bản để tạo và định dạng nội dung trên web.
* CSS (Cascading Style Sheets): Được sử dụng để tùy chỉnh giao diện và kiểu dáng của trang web

## **1.3 Phát triển Web Fornt-End và Back-End**

* HTML (HyperText Markup Language): Là ngôn ngữ đánh dấu cơ bản để tạo và định dạng nội dung trên web.
* CSS (Cascading Style Sheets): Được sử dụng để tùy chỉnh giao diện và kiểu dáng của trang web.

## **1.4 Công nghệ liên quan**

* Responsive Web Design: Tạo ra trang web có thể hiển thị đẹp trên các thiết bị và kích thước màn hình khác nhau.
* Single Page Applications (SPAs): Ứng dụng web mà tất cả trang chuyển đổi và tải dữ liệu bằng AJAX, không yêu cầu tải lại trang.
* Progressive Web Apps (PWAs): Ứng dụng web kết hợp các tính năng của ứng dụng di động, cho phép hoạt động ngoại tuyến và đánh giá cao trải nghiệm người dùng.
* Serverless Computing: Sử dụng dịch vụ điện toán đám mây để quản lý cơ sở hạ tầng và tập trung vào việc phát triển ứng dụng.

## **1.5 Bảo mật và quyền riêng tư**

* Công nghệ web đối mặt với nhiều thách thức về bảo mật và quyền riêng tư, bao gồm bảo vệ khỏi tấn công mạng, xác thực người dùng, mã hóa dữ liệu, và tuân thủ các quy định về quyền riêng tư như GDPR.

## **1.6 Cộng đồng và tài nguyên**

* Cộng đồng phát triển web rất lớn, với hàng ngàn tài liệu, khóa học trực tuyến, các công cụ phát triển, và cộng đồng trò chuyện trực tuyến để hỗ trợ những người làm việc trong lĩnh vực này.
* Trên tất cả, công nghệ web là một lĩnh vực đa dạng và thú vị, đang không ngừng phát triển và thay đổi để đáp ứng nhu cầu ngày càng cao của người sử dụng internet và doanh nghiệp trực tuyến.

# Chương II : Giới thiệu về HTML

## **2.1 Tổng quan về HTML**

HTML là viết tắt của "HyperText Markup Language," và nó là ngôn ngữ đánh dấu cơ bản được sử dụng để tạo và định dạng các trang web trên Internet. HTML cho phép bạn xây dựng cấu trúc và hiển thị nội dung trên web, bao gồm văn bản, hình ảnh, video, liên kết, biểu đồ và nhiều thành phần khác. Dưới đây là một số điểm quan trọng về HTML

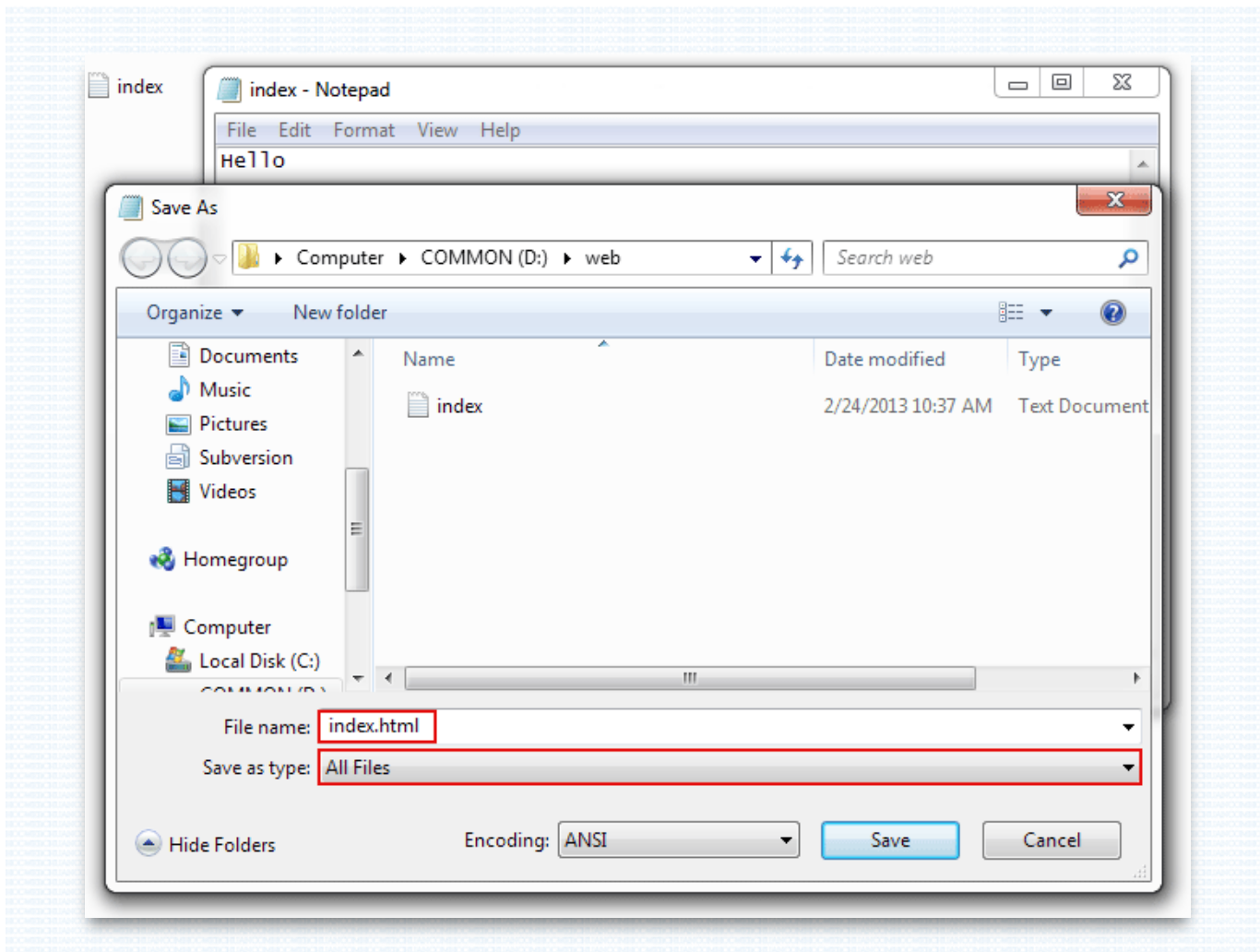
* Cú Pháp Đánh Dấu: HTML sử dụng cú pháp dựa trên các thẻ (tags) và các thuộc tính (attributes) để xác định cấu trúc và định dạng của nội dung. Mỗi thẻ bắt đầu bằng < và kết thúc bằng >. Ví dụ: <p> là thẻ bắt đầu cho một đoạn văn bản, và </p> là thẻ kết thúc.
* Cấu Trúc Cơ Bản: Một trang HTML cơ bản thường bao gồm một phần đầu (<head>) để đặt tiêu đề, tập tin CSS, và các thông tin khác, và một phần thân (<body>) chứa nội dung hiển thị của trang web.
* Thẻ và Thuộc Tính: HTML sử dụng nhiều thẻ để biểu thị các loại nội dung khác nhau. Mỗi thẻ có thể có các thuộc tính để xác định thông tin bổ sung. Ví dụ: <img> thường có thuộc tính src để chỉ định đường dẫn đến hình ảnh và alt để mô tả hình ảnh.
* Liên Kết: Thẻ <a> được sử dụng để tạo liên kết (hyperlink) đến các trang web khác hoặc tài liệu. Thuộc tính href trong thẻ <a> xác định đường dẫn đến trang bạn muốn liên kết đến.
* Danh Sách và Bảng: HTML cho phép bạn tạo danh sách không có thứ tự (<ul>) và danh sách có thứ tự (<ol>) cùng với các bảng (<table>) để tổ chức dữ liệu.
* Phiên Bản HTML: Có nhiều phiên bản của HTML, như HTML4, HTML5, và HTML5.1. HTML5 là phiên bản phổ biến nhất và được sử dụng rộng rãi trên web ngày nay.Tích Hợp Multimedia: HTML5 cung cấp hỗ trợ tích hợp video và audio trực tiếp vào trang web thông qua các thẻ như <video> và <audio>, giúp tạo ra trải nghiệm tương tác và đa phương tiện.
* CSS và JavaScript: HTML thường kết hợp với CSS (Cascading Style Sheets) để tùy chỉnh giao diện và JavaScript để thêm tính năng tương tác vào trang web.

## **2.2 Hướng dẫn tạo một tệp HTML cơ bản**

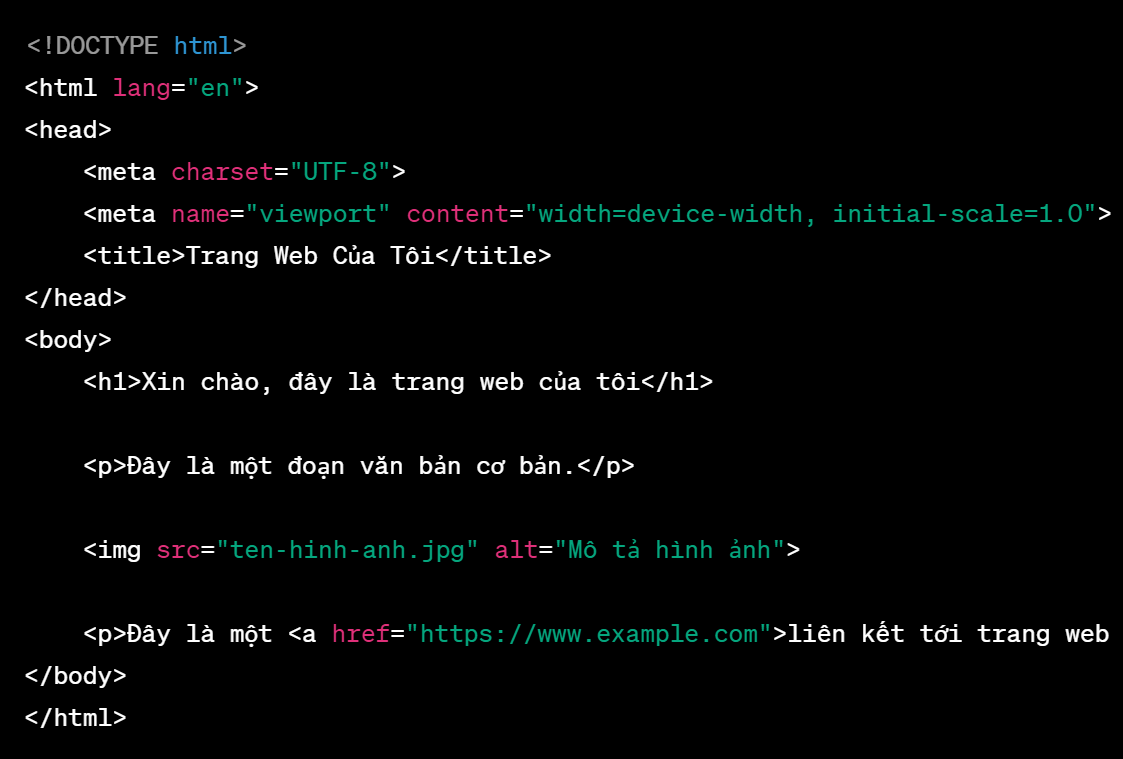
Trước tiên ta tạo một file *index.txt* đơn giản bằng *notepad* trong Window với nội dung sau:

<p>Hello</p>

Sau đó "Save As" lại file với định dạng mới là .html



## **2.3 Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết trong HTML**



Giải thích :

1. <img>: Để chèn hình ảnh, bạn sử dụng thẻ <img>. Trong ví dụ trên, src là đường dẫn tới hình ảnh và alt là văn bản mô tả hình ảnh (hiển thị khi hình ảnh không thể tải được).

2. <a>: Để tạo siêu liên kết, bạn sử dụng thẻ <a>. Trong ví dụ trên, href là đường dẫn tới trang web bạn muốn liên kết đến.

Hãy chắc chắn thay đổi giá trị của src và href để trỏ đúng đường dẫn tới hình ảnh và trang web bạn muốn sử dụng.

Lưu ý rằng bạn cũng có thể tùy chỉnh trang web bằng cách thêm CSS để cải thiện giao diện và trình bày trang web. Để thực hiện điều này, bạn có thể thêm thẻ <style> vào phần <head> của tài liệu HTML hoặc tạo một tệp CSS riêng và liên kết nó với trang HTML của bạn bằng thẻ <link>.

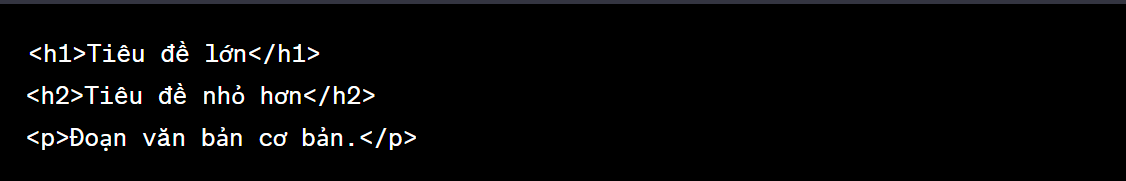
## **2.4 Hướng dẫn cú pháp và các phần tử HTML cơ bản**

### **2.4.1 Cú pháp cơ bản của HTML**

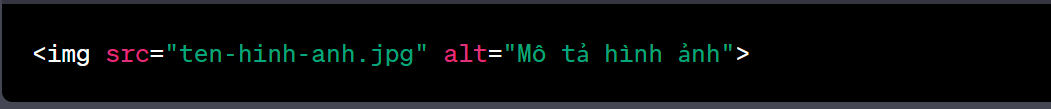
* Thẻ HTML: Một trang web HTML bắt đầu bằng thẻ <!DOCTYPE html> để xác định phiên bản HTML và mở thẻ <html>.
* Thẻ <head>: Phần này chứa các thông tin về trang web như tiêu đề, tập tin CSS, meta tags, và nhiều thứ khác. Thẻ <head> được mở bằng <head> và đóng bằng </head>.
* Thẻ <title>: Được sử dụng để đặt tiêu đề của trang web, nó nằm trong phần <head>. Ví dụ: <title>Tiêu đề trang web</title>.
* Thẻ <meta>: Sử dụng để cung cấp thông tin về tài liệu như mã ký tự, viewport, và các thông tin khác. Ví dụ: <meta charset="UTF-8">.
* Thẻ <body>: Đây là phần chứa nội dung hiển thị của trang web, bao gồm văn bản, hình ảnh, liên kết, và nhiều thứ khác. Thẻ <body> mở bằng <body> và đóng bằng </body>.

### 2.4.2 Các thành phần HTML cơ bản

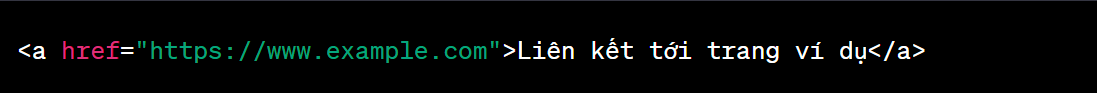
* Văn bản: Để thêm văn bản vào trang web, bạn sử dụng thẻ <p> cho đoạn văn bản đơn giản và <h1>, <h2>,...,<h6> cho các tiêu đề với <h1> là tiêu đề chính và <h6> là tiêu đề nhỏ nhất.



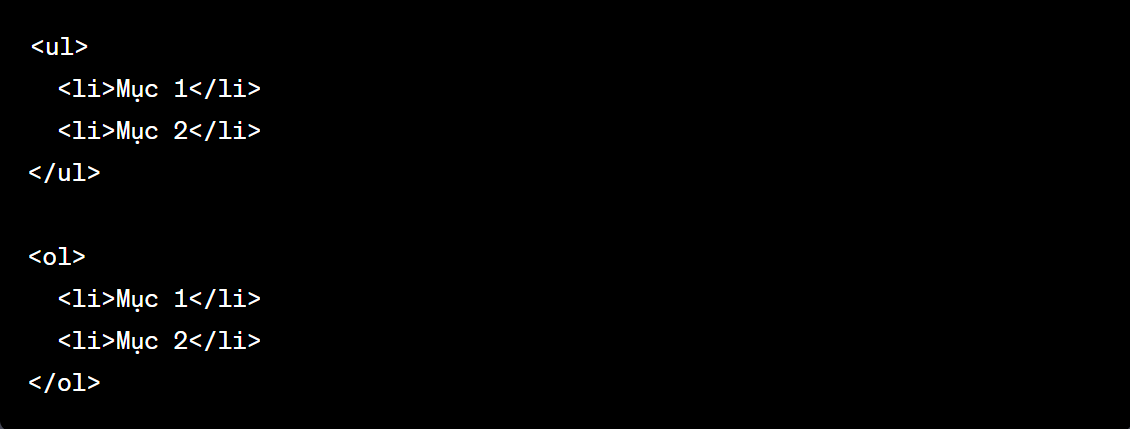
* Hình ảnh: Để chèn hình ảnh, sử dụng thẻ <img> với thuộc tính src để chỉ định đường dẫn đến hình ảnh và alt để mô tả hình ảnh (cho trường hợp hình ảnh không thể hiển thị).



* Siêu liên kết: Để tạo liên kết, bạn sử dụng thẻ <a> với thuộc tính href để chỉ định đường dẫn đến trang web hoặc tài liệu mà bạn muốn liên kết đến.

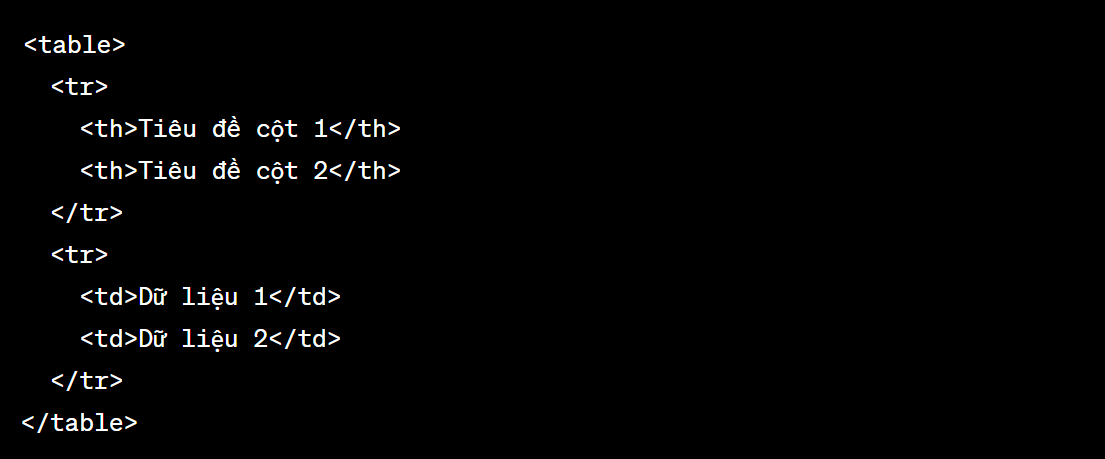


* Danh sách: Để tạo danh sách, bạn có thể sử dụng thẻ <ul> cho danh sách không có thứ tự (unordered list) và <ol> cho danh sách có thứ tự (ordered list). Các mục trong danh sách được định nghĩa bằng thẻ <li>.



* Bảng: Để tạo bảng, bạn sử dụng thẻ <table>, <tr> cho hàng, <th>

cho tiêu đề cột, và <td> cho ô dữ liệu.



# Chương III : Cơ bản về CSS

## **3.1 Tổng quan về CSS**

CSS, hay "Cascading Style Sheets," là một ngôn ngữ đánh dấu được sử dụng để điều chỉnh giao diện và kiểu dáng của các trang web và tài liệu HTML. CSS cho phép bạn tách biệt kiểu dáng và bố cục của trang web từ nội dung, giúp bạn tạo ra các trang web đẹp và dễ bảo trì.

## **3.2 Hướng dẫn thêm kiểu cho trang web bằng CSS**

### **3.2.1 Tạo tệp CSS**

* Bắt đầu bằng việc tạo một tệp CSS riêng biệt. Bạn có thể tạo một tệp có phần mở rộng .css, ví dụ: styles.css.
* Đảm bảo đặt tệp CSS này trong cùng thư mục với trang HTML của bạn hoặc cung cấp đường dẫn đến tệp CSS nếu nó ở một vị trí khác.

### **3.2.2 Liên kết tệp CSS với trang HTML**

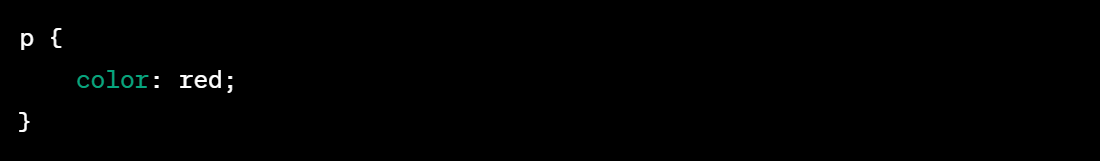
* Trong phần <head> của trang HTML, bạn cần thêm thẻ <link> để liên kết tệp CSS với trang. Thẻ này sẽ trông như sau:



### **3.2.3 Viết mã CSS**

* Trong tệp CSS, bạn có thể bắt đầu viết mã CSS để thêm kiểu cho trang web của bạn. Mã CSS được viết bên trong cặp dấu ngoặc nhọn {} và áp dụng cho các phần tử HTML thông qua các chọn lọc (selectors).

Ví dụ, để thay đổi màu văn bản của tất cả các đoạn văn bản (<p>) trong trang web của bạn thành màu đỏ, bạn có thể sử dụng mã CSS như sau:



* Đây là một số ví dụ về các thuộc tính CSS phổ biến:

color: Để đặt màu văn bản.

font-size: Để đặt kích thước font chữ.

font-family: Để đặt kiểu font chữ.

background-color: Để đặt màu nền.

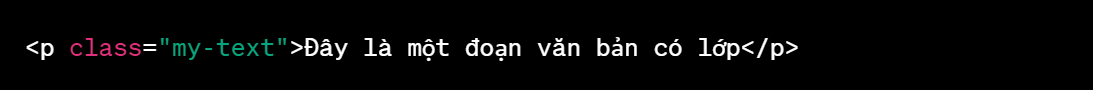
margin, padding: Để điều chỉnh khoảng cách xung quanh phần tử.

border: Để tạo viền xung quanh phần tử.

text-align: Để căn chỉnh văn bản.

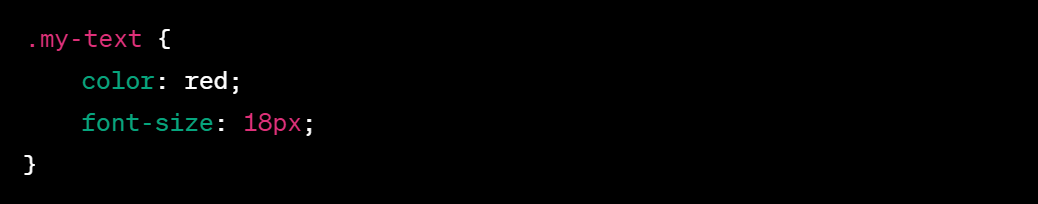
## **3.3 Tạo các lớp và áp dụng bằng cách sử dụng lựa chọn**

### **3.3.1 Tạo lớp**



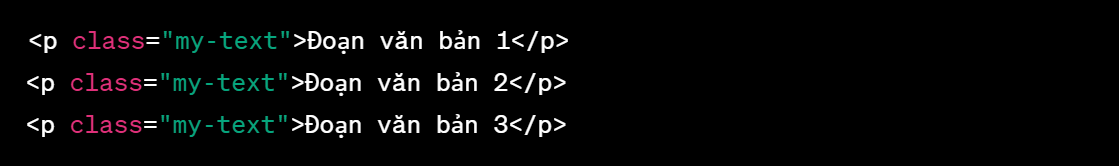
### **3.3.2 Sử dụng lựa chọn bằng lớp trong CSS**

* Trong tệp CSS, bạn có thể sử dụng lựa chọn bằng lớp bằng cách sử dụng dấu chấm (.) trước tên lớp. Ví dụ:



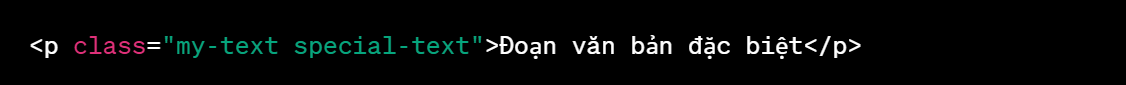
Trong ví dụ này, tất cả các phần tử HTML có lớp là "my-text" sẽ có màu đỏ và kích thước font là 18px.

### **3.3.3 Áp dụng lớp vào nhiều phần tử**



### **3.3.4 Sử dụng nhiều lớp cho một phần tử**

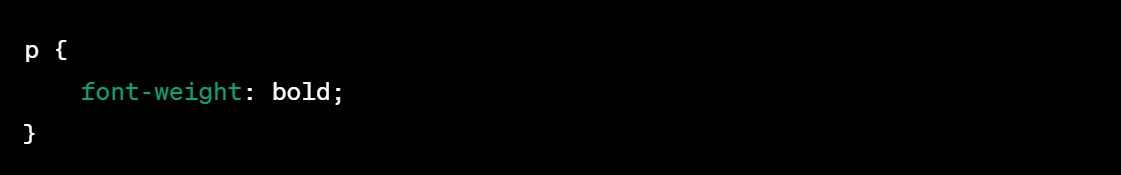
* Bạn có thể sử dụng nhiều lớp cho cùng một phần tử HTML bằng cách sử dụng nhiều thuộc tính class. Ví dụ:



Trong ví dụ này, phần tử <p> sẽ áp dụng cả kiểu từ lớp "my-text" và lớp "special-text" trong tệp CSS.

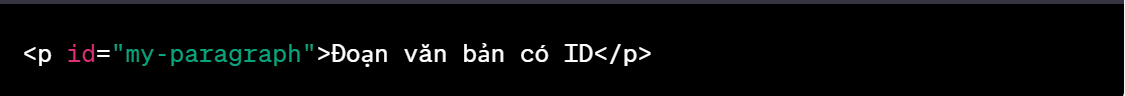
### **3.3.5 Áp dụng bằng các sử dụng lựa chọn**

* Lựa chọn bằng thẻ HTML :

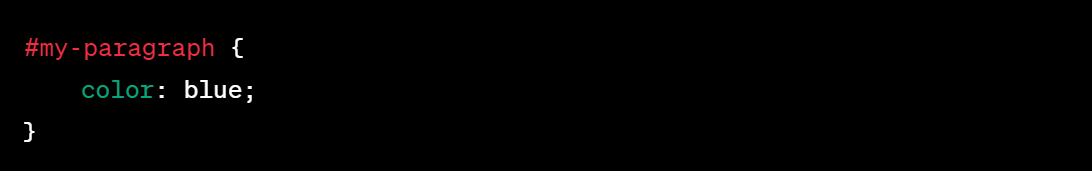


Tất cả các đoạn văn bản trong thẻ <p> sẽ có độ đậm (font-weight) là đậm.

* Lựa chọn bằng ID :



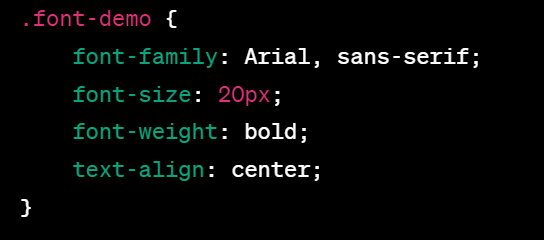
* CSS :



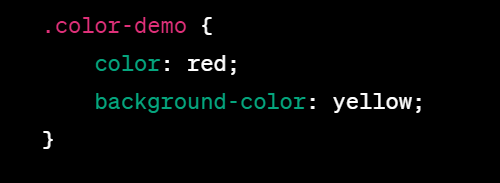
Đoạn văn bản có ID "my-paragraph" sẽ có màu xanh.

### **3.3.6 Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc độ rộng và chiều cao**

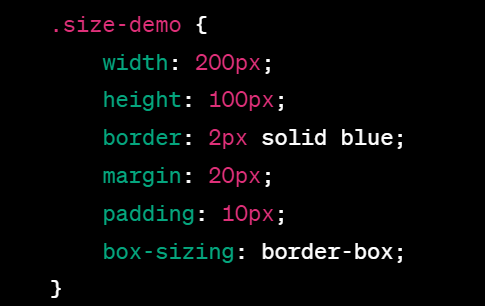
* Thử nghiệm với thuộc tính font chữ :



* Thử nghiệm với thuộc tính màu sắc :



* Thử nghiệm với thuộc tính độ rộng và chiều cao :



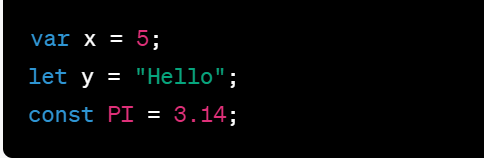
# Chương IV : JavaScript cơ bản

**4.1 Giới thiệu cú pháp JavaScript cơ bản**

JavaScript là một ngôn ngữ lập trình phía máy khách (client-side) phổ biến trong phát triển web. Dưới đây là một giới thiệu về cú pháp JavaScript cơ bản:

**4.1.1 Biến và kiểu dữ liệu**

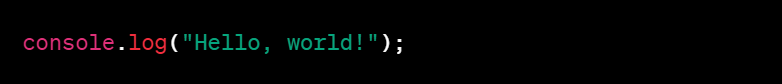
* Biến (Variables): Được sử dụng để lưu trữ dữ liệu. Khai báo biến bằng var, let, hoặc const. Ví dụ:



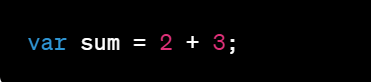
* Kiểu dữ liệu (Data Types): JavaScript có kiểu dữ liệu như số (number), chuỗi (string), boolean (true hoặc false), mảng (array), đối tượng (object), và nhiều kiểu dữ liệu khác.

**4.1.2 Câu lệnh và biểu thức**

* Câu lệnh (Statements): Được sử dụng để thực hiện một hành động. Ví dụ:

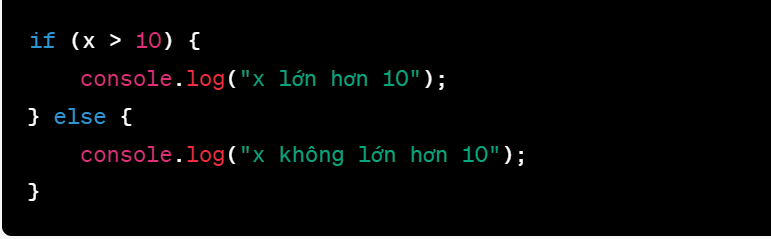


* Biểu thức (Expressions): Được sử dụng để tính toán giá trị. Ví dụ:

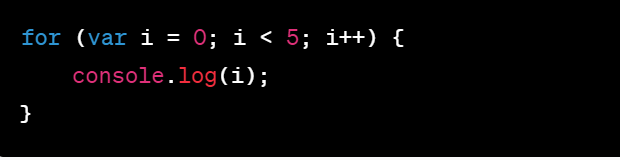


## **4.1.3 Điều khiển luồng**

* Câu lệnh if-else: Sử dụng để kiểm tra điều kiện và thực hiện hành động dựa trên kết quả. Ví dụ:

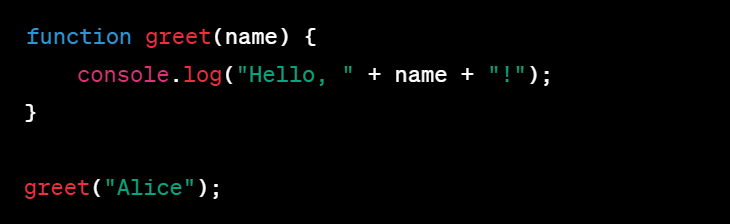


* Vòng lặp (Loops): Sử dụng để lặp qua một dãy lệnh nhiều lần. Ví dụ



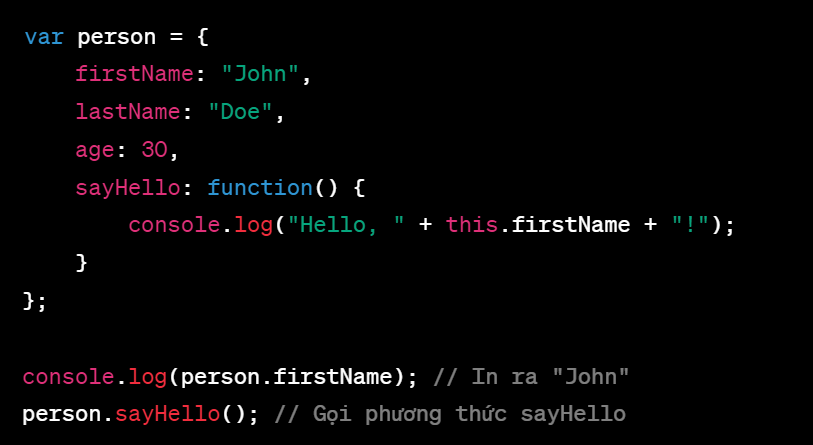
**4.1.4 Hàm**

Hàm (Functions): Được sử dụng để đóng gói một khối mã để thực hiện một nhiệm vụ cụ thể. Ví dụ:



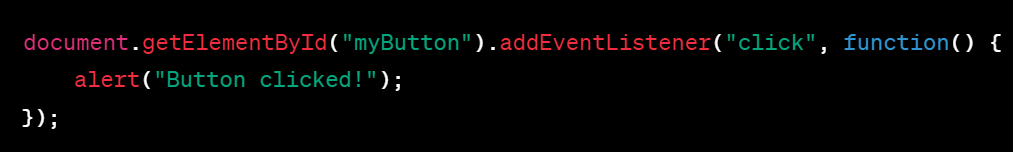
**4.1.5 Đối tượng**

Đối tượng (Objects): Là một tập hợp các thuộc tính và phương thức. Ví dụ:



**4.1.6 Sự kiện**

Sự kiện (Events): Được sử dụng để phản ứng khi người dùng tương tác với trang web. Ví dụ:



**4.1.7 Bất đồng bộ và AJAX**

JavaScript cũng hỗ trợ lập trình bất đồng bộ thông qua Promise và async/await, cho phép thực hiện các hoạt động không chờ đợi, chẳng hạn như tải tài liệu từ máy chủ sử dụng AJAX.

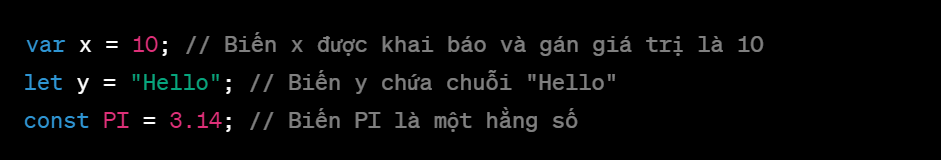
**4.1.8 Các Thư Viện và Framework**

* JavaScript có nhiều thư viện và framework mạnh mẽ như jQuery, React, Angular, và Vue.js, giúp bạn xây dựng ứng dụng web phức tạp và tương tác dễ dàng hơn.
* JavaScript là một ngôn ngữ đa năng và mạnh mẽ, được sử dụng rộng rãi trong phát triển web và nhiều ứng dụng khác. Điều này chỉ là một giới thiệu cơ bản, và có nhiều khía cạnh và tính năng phức tạp hơn để khám phá trong JavaScript.

**4.2 Hướng dẫn tạo biến, hàm và sử dụng sự kiện**

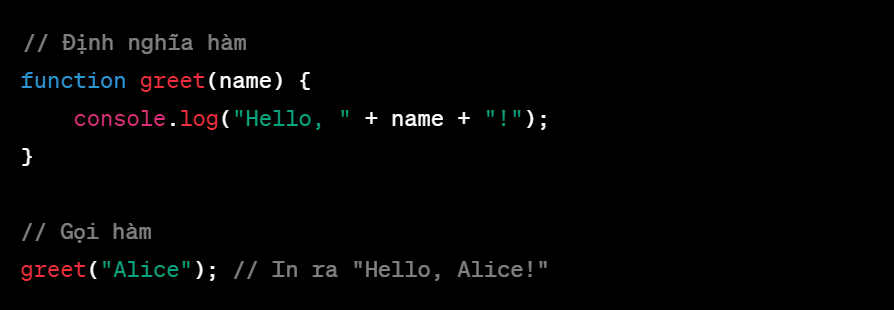
**4.2.1 Tạo biến**

* Biến trong JavaScript có thể được tạo bằng các từ khóa var, let, hoặc const. Dưới đây là ví dụ về cách tạo biến:



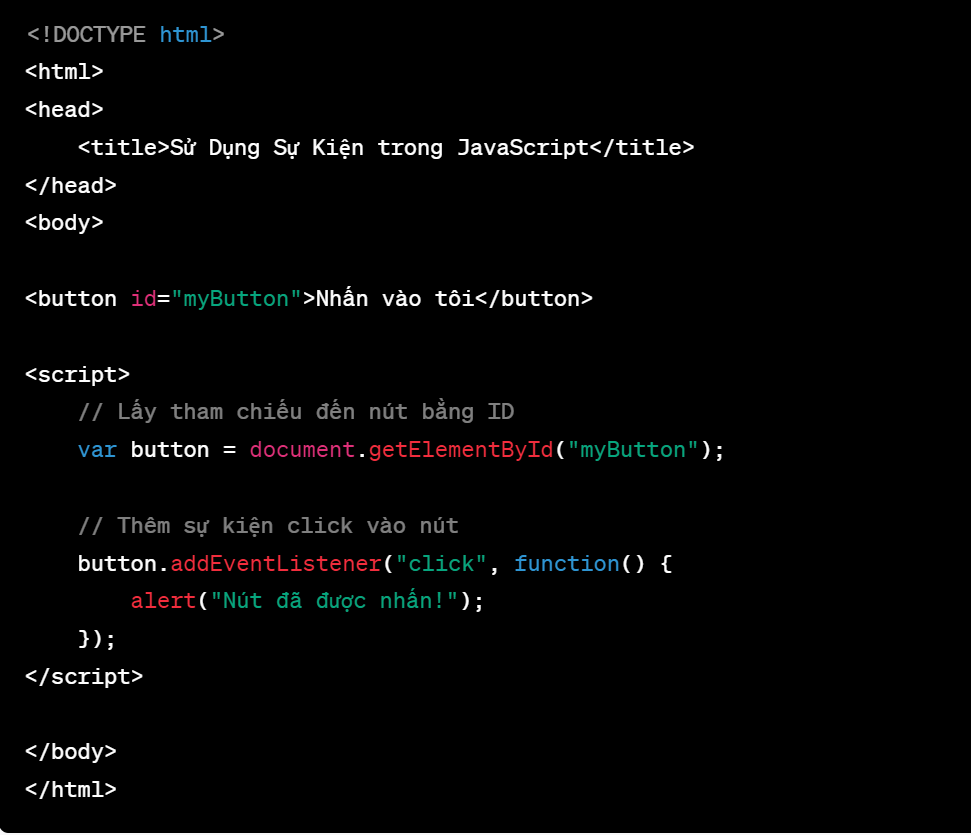
**4.2.2 Tạo hàm**

* Hàm trong JavaScript là một khối mã được đóng gói lại để thực hiện một nhiệm vụ cụ thể. Dưới đây là ví dụ về cách tạo và sử dụng hàm:



### **4.2.3 Sử dụng sự kiện**

* JavaScript cho phép bạn phản ứng với các sự kiện trên trang web. Dưới đây là ví dụ về cách sử dụng sự kiện click trên một nút:



Trong ví dụ này, chúng ta đã lấy tham chiếu đến nút bằng cách sử dụng getElementById và sau đó thêm một sự kiện click bằng addEventListener. Khi nút được nhấn, một thông báo xuất hiện.

## **4.3 Ví dụ đơn giản hiển thị thông báo khi được ấn nút trong JavaScript**



Trong ví dụ này:

* Chương trình đã được tạo một nút có ID là "myButton" bằng cách sử dụng thẻ <button>.
* Sau đó, trong tệp JavaScript, chúng tôi đã sử dụng document.getElementById("myButton") để lấy tham chiếu đến nút.
* Sau đó đã thêm sự kiện "click" cho nút bằng cách sử dụng addEventListener. Khi nút được nhấn, hàm xử lý sự kiện sẽ được gọi.
* Trong hàm xử lý sự kiện, chúng tôi đã sử dụng alert() để hiển thị một thông báo trình duyệt cho người dùng.
* Khi bạn mở trang HTML này và nhấn vào nút, bạn sẽ thấy một hộp thoại thông báo xuất hiện với nội dung "Bạn đã nhấn vào nút!"

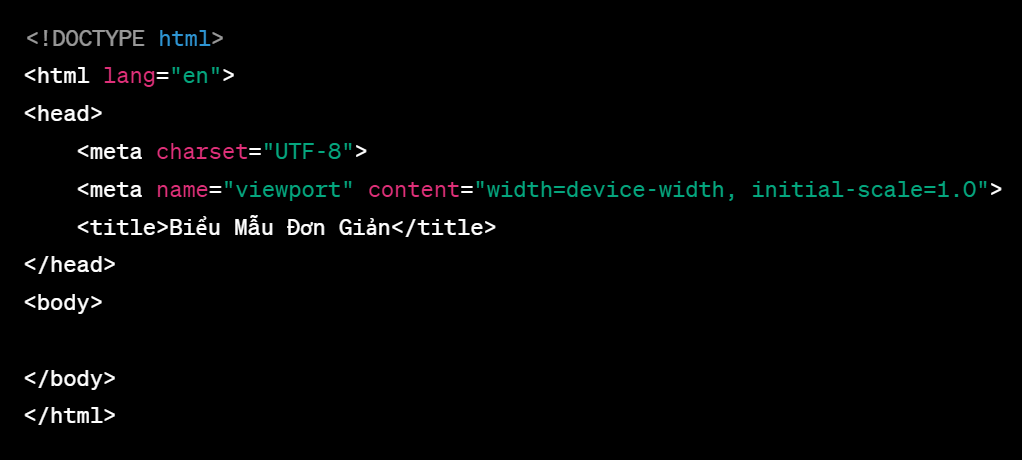
# Chương V : HTML Forms và Input Elements

## **5.1 Tổng quan về HTML Forms và Input Elements**

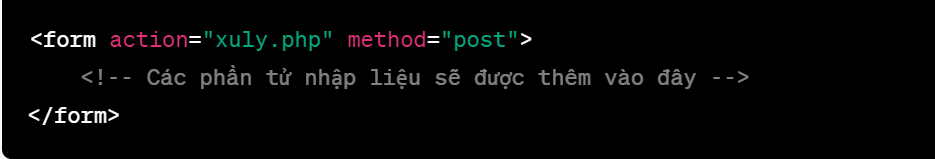
* HTML Forms (biểu mẫu HTML) là một phần quan trọng của trang web, cho phép người dùng tương tác và gửi dữ liệu đến máy chủ web. Một biểu mẫu HTML chứa các trường (fields) hoặc phần tử (elements) như hộp văn bản, nút radio, nút bảng chọn, nút gửi, và nhiều loại trường khác để thu thập thông tin từ người dùng.
* Input Elements (phần tử nhập) là các phần tử HTML được sử dụng trong biểu mẫu (form) để thu thập dữ liệu từ người dùng. Các phần tử này cho phép người dùng nhập và chọn thông tin và sau đó gửi nó đến máy chủ web để xử lý.

## **5.2 Từng bước tạo biểu mẫu đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, nút gửi**

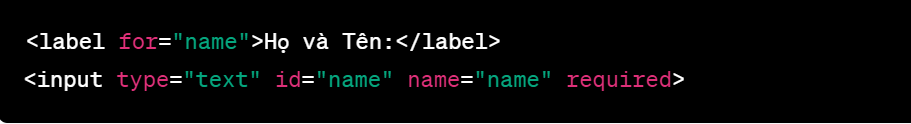
Bước 1 : Tạo một tệp HTML cơ bản



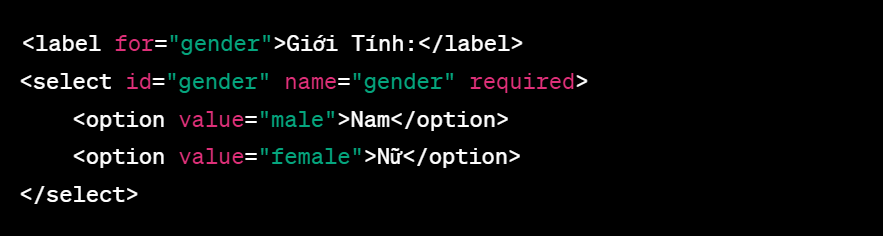
Bước 2 : Tạo biểu mẫu trong thẻ body



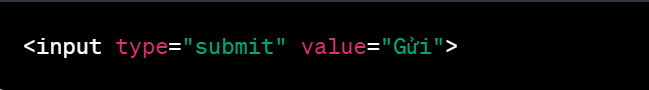
Bước 3 : Thêm ô văn bản (text input)



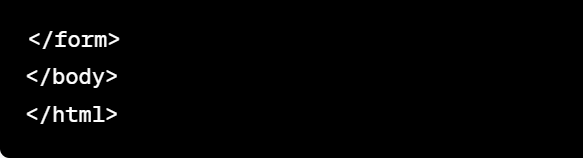
Bước 4 : Thêm ô chọn (select)



Bước 5 : Thêm nút gửi (submit)

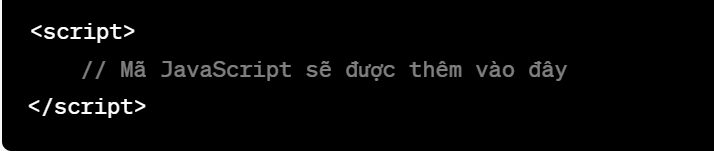


Bước 6 : Hoàn thành tệp HTML

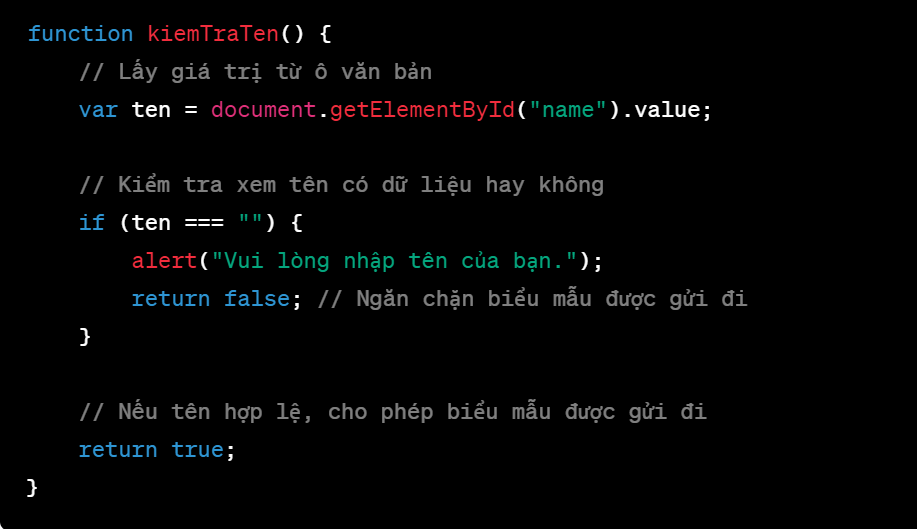


## **5.3 Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu để kiểm tra dữ liệu hợp lệ**

* Thêm JavaScript vào trang HTML: Đầu tiên ta cần thêm một phần JavaScript vào trang HTML của bạn. Ta có thể đặt mã JavaScript bên trong thẻ <script> tại phần cuối thẻ <body> hoặc trong phần <head> của trang.



* Tạo hàm kiểm tra dữ liệu hợp lệ: Bây giờ ta có thể tạo một hàm JavaScript để kiểm tra dữ liệu từ biểu mẫu. Dưới đây là một ví dụ về cách kiểm tra xem ô văn bản chứa tên đã được nhập hay chưa:



# CHƯƠNG VI : Quy trình phăt triển web.

## **6.1 Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web.**

Quy trình phát triển web là một loạt các bước được thực hiện để tạo ra một trang web hoạt động hiệu quả và chất lượng. Dưới đây là một giới thiệu tổng quan về quy trình phát triển web từ lập kế hoạch đến triển khai trang web:

### **6.1.1 Lập kế hoạch (Planning)**

* Xác định mục tiêu: Đầu tiên, bạn cần xác định mục tiêu của trang web, ví dụ: blog cá nhân, cửa hàng trực tuyến, trang web doanh nghiệp, v.v.
* Nghiên cứu thị trường: Tìm hiểu về đối thủ và khách hàng tiềm năng, cũng như xác định yêu cầu của họ.
* Xây dựng budget và lịch trình: Điều này bao gồm xác định nguồn tài chính, thời gian dự kiến và nguồn lực cần thiết.

### **6.1.2 Thiết kế (Design)**

* Thiết kế giao diện (UI/UX): Tạo ra các bản vẽ, mẫu giao diện người dùng (UI) và đảm bảo trải nghiệm người dùng (UX) tốt.
* Chọn màu sắc và hình ảnh: Xác định bảng màu và thu thập hình ảnh, biểu đồ, biểu đồ, v.v. cho trang web.
* Tạo wireframe và prototype: Tạo bản thiết kế tường thuật (wireframe) và bản mẫu (prototype) để thử nghiệm trước.

### **6.1.3 Phát triển (Development)**

* Lập trình và xây dựng cơ sở hạ tầng: Sử dụng ngôn ngữ lập trình và công cụ phát triển để xây dựng trang web.
* Quản lý cơ sở dữ liệu: Tạo và quản lý cơ sở dữ liệu cho ứng dụng web nếu cần.
* Kiểm tra và sửa lỗi: Kiểm tra và sửa các lỗi trong mã nguồn, đảm bảo tính ổn định và an toàn.

### **6.1.4 Kiểm tra và đánh giá (Testing and Evaluation)**

* Kiểm tra chức năng: Kiểm tra các tính năng của trang web để đảm bảo chúng hoạt động đúng cách.
* Kiểm tra tương thích trình duyệt: Đảm bảo trang web hiển thị đúng trên các trình duyệt khác nhau.
* Kiểm tra hiệu suất: Đánh giá hiệu suất trang web và tối ưu hóa tốc độ tải trang.

### **6.1.5 Triển khai (Deployment)**

* Chuẩn bị máy chủ và tên miền: Cài đặt máy chủ web và đăng ký tên miền nếu chưa có.
* Tải lên trực tuyến: Đưa trang web lên máy chủ để nó trở nên truy cập được trên internet. Kiểm tra cuối cùng:
* Kiểm tra trang web sau khi triển khai để đảm bảo tất cả các thành phần hoạt động bình thường.

### **6.1.6 Quản lý và bảo trì (Management and Maintenance)**

* Sao lưu và bảo mật: Thực hiện sao lưu định kỳ và duy trì bảo mật để đảm bảo trang web luôn an toàn.
* Cập nhật và mở rộng: Thêm tính năng mới và cập nhật nội dung khi cần thiết để đáp ứng yêu cầu người dùng mới.
* Kiểm tra định kỳ: Kiểm tra và theo dõi hiệu suất trang web, đảm bảo nó hoạt động mượt mà.

Quy trình phát triển web có thể thay đổi tùy thuộc vào dự án cụ thể và công nghệ sử dụng, nhưng các bước trên cung cấp một cơ sở chung để xây dựng và duy trì một trang web thành công.

## 6**.2 Một số ví dụ về các bước trong quy trình phát triển web**

### **6.2.1 Phân tích yêu cầu (Requirements Analysis)**

* Ví dụ 1: Một công ty du lịch muốn phát triển một trang web đặt tour du lịch trực tuyến. Yêu cầu bao gồm chức năng đặt tour, tích hợp thanh toán trực tuyến và hiển thị thông tin chi tiết về các tour.
* Ví dụ 2: Một trường học muốn xây dựng một trang web học trực tuyến. Yêu cầu bao gồm chức năng đăng nhập học viên, hiển thị nội dung học tập và kiểm tra trực tuyến.

### **6.2.2 Thiết kế giao diện (UI/UX Design)**

* Ví dụ 1: Dự án thiết kế trang web cho một cửa hàng thời trang. Giao diện bao gồm một trang chủ với hình ảnh sản phẩm, danh mục sản phẩm, và giỏ hàng dễ sử dụng.
* Ví dụ 2: Thiết kế giao diện cho một ứng dụng di động về sức khỏe, với thiết kế dựa trên nguyên tắc UX để đảm bảo dễ sử dụng cho người dùng.

### **6.2.3 Phát triển (Development)**

* Ví dụ 1: Sử dụng ngôn ngữ lập trình như JavaScript và framework như React để xây dựng giao diện và tích hợp hệ thống thanh toán trực tuyến.
* Ví dụ 2: Phát triển ứng dụng di động cho hệ điều hành iOS và Android bằng cách sử dụng ngôn ngữ lập trình Swift cho iOS và Java/Kotlin cho Android.

### **6.2.4 Kiểm tra và Kiểm tra (Testing and Quality Assurance)**

* Ví dụ 1: Kiểm tra tích hợp thanh toán trực tuyến để đảm bảo rằng các giao dịch được xử lý đúng cách và an toàn.
* Ví dụ 2: Thực hiện kiểm tra tương thích trình duyệt trên các trình duyệt phổ biến như Chrome, Firefox, và Safari.

### **6.2.5 Triển khai (Deployment)**

* Ví dụ 1: Chuẩn bị máy chủ web và đăng ký tên miền, sau đó triển khai trang web du lịch trực tuyến để khách hàng có thể truy cập và đặt tour.
* Ví dụ 2: Đưa ứng dụng di động lên App Store của Apple và Google Play để người dùng có thể tải về và sử dụng.

Các ví dụ trên giúp minh họa cách các bước trong quy trình phát triển web áp dụng cho các dự án cụ thể và các ngữ cảnh khác nhau. Quy trình này có thể biến đổi và điều chỉnh dựa trên nhu cầu cụ thể của dự án và công nghệ được sử dụng.

## **6.3 Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm, bao gồm quản lý dự án và phát triển.**

Vai trò và trách nhiệm của từng thành viên:

1.Đỗ Thông Minh

2.Lê Bá Vũ

3.Bùi Mạnh Hùng