

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**CÔNG NGHỆ WEB**

**Đề tài : Xây Dựng Thiết Kế Website Nghe Nhạc**

**Nhóm 4 \_ 20CN5**

**Giảng viên hướng dẫn : Nguyễn Thị Hạnh**

**Nhóm Sinh viên thực hiện:**

1. **Nguyễn Đức Thắng**
2. **Nguyễn Thành Trung**

**Hà Nội \_2023**

**Lớp : 20VL**

**Giảng viên giảng dạy:**

Giảng viên giảng dạy : Lương Thị Bích Phượng

HÀ NỘI, 10/2023

MỤC LỤC

[1. Giới thiệu về HTML 4](#_Toc149056980)

[**1.1** **Hướng dẫn về cú pháp và các phần tử HTML cơ bản.** 4](#_Toc149056981)

[**1.2** **Hướng dẫn tạo một tệp HTML cơ bản** 6](#_Toc149056982)

[**1.3** **Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết** 7](#_Toc149056983)

[2. Cơ bản về CSS 8](#_Toc149056984)

[**2.1 Hướng dẫn cách thêm kiểu cho trang web bằng CSS.** 8](#_Toc149056985)

[**2.2 Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn.** 9](#_Toc149056986)

[**2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao.** 11](#_Toc149056987)

[3. JavaScript Cơ bản 13](#_Toc149056988)

[3.1. Giới thiệu cú pháp JavaScript cơ bản 13](#_Toc149056989)

[**3.2. Hướng dẫn tạo biến , hàm và sử dụng sự kiện** 15](#_Toc149056990)

[**3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn.** 17](#_Toc149056991)

[4. HTML Forms và Input Elements 18](#_Toc149056992)

[**4.1. Hướng dẫn cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi.** 18](#_Toc149056993)

[**4.2. Tạo kiểu cho biểu mẫu bằng CSS.** 22](#_Toc149056994)

[**4.3. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu (vd: kiểm tra dữ liệu hợp lệ).** 25](#_Toc149056995)

[**a. Truy cập đến form dùng javascript** 25](#_Toc149056996)

[**b.Truy cập đến các control trong form để lấy value** 25](#_Toc149056997)

[**c. Kiểm tra dữ liệu trong Form với JavaScript** 26](#_Toc149056998)

[5. Quy trình phát triển web 27](#_Toc149056999)

[**5.1. Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web.** 27](#_Toc149057000)

[**5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát triển, kiểm thử, và triển khai.** 29](#_Toc149057001)

[**5.3 : Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm, bao gồm quản lý dự án và phát triển.** 30](#_Toc149057002)

[6. Thực hành xây dựng trang web đơn giản: 31](#_Toc149057003)

[**6.1. Xây dựng trang web đơn giản từ đầu bằng việc kết hợp HTML, CSS và JavaScript.** 31](#_Toc149057004)

[**6.2. Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL): Bài thi** 33](#_Toc149057005)

[7. Responsive Web Design: 33](#_Toc149057006)

[**7.1. Giới thiệu về cách tạo trang web phản hồi và thân thiện với nhiều thiết bị.** 33](#_Toc149057007)

[**7.2. Sử dụng các kỹ thuật CSS media queries để điều chỉnh giao diện cho các màn hình khác nhau** 35](#_Toc149057008)

[**7.3 Hướng dẫn cách tạo bố cục đáng linh hoạt cho các thiết bị khác nhau.** 37](#_Toc149057009)

[8.HTML5 và CSS3 40](#_Toc149057010)

[**8.1 Giới thiệu về các tính năng mới trong HTML5 và CSS3.** 40](#_Toc149057011)

[**8.2 Hướng dẫn cách sử dụng các phần tử HTML5 như video, âm thanh, canvas, và đồ họa vector** 42](#_Toc149057012)

[**8.3 Sử dụng CSS3 để thêm các hiệu ứng và chuyển động vào trang web, bao gồm cả CSS transitions và animations.** 43](#_Toc149057013)

[9.Thiết kế trang web thân thiện với SEO 44](#_Toc149057014)

[**9.1 Giới thiệu về SEO (Search Engine Optimization) và tại sao nó quan trọng đối** với trang web 44](#_Toc149057015)

[**9.2 Hướng dẫn cách tối ưu hóa tiêu đề trang, thẻ meta, và nội dung để cải thiện thứ hạng tìm kiếm.** 45](#_Toc149057016)

[1. Tối ưu thẻ tiêu đề và thẻ meta 45](#_Toc149057017)

[2. Tối ưu thẻ alt 46](#_Toc149057018)

[3. Tối ưu các từ khóa phù hợp 46](#_Toc149057019)

[4. Tối ưu nội dung 47](#_Toc149057020)

[**9.3 Sử dụng công cụ tối ưu hóa SEO để theo dõi hiệu suất và tối ưu hóa trang web.** 47](#_Toc149057021)

[2.1. Ahrefs: Công cụ SEO keyword 48](#_Toc149057022)

[2.2. BuzzStream 48](#_Toc149057023)

[2.3. Google PageSpeed Insights 49](#_Toc149057024)

[2.4. Google Webmaster Tools 49](#_Toc149057025)

[2.5. Google Analytics 49](#_Toc149057026)

[2.6. Copyscape 49](#_Toc149057027)

[2.7. Công cụ Yoast SEO 50](#_Toc149057028)

[10.Quản lý phiên bản và Git 50](#_Toc149057029)

[**10.1 Giới thiệu về hệ thống quản lý mã nguồn và Git.** 50](#_Toc149057030)

[**10.2 Hướng dẫn tạo kho lưu trữ (repository) Git và làm việc với các lệnh cơ bản như git clone, git pull, git push, và git commit.** 51](#_Toc149057031)

[**10.3 Thảo luận về việc làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án.** 52](#_Toc149057032)

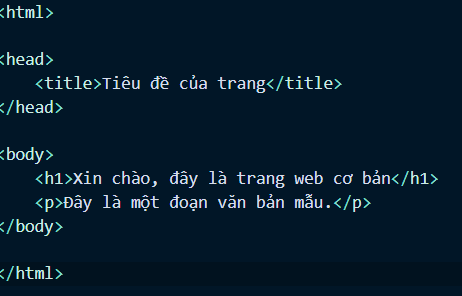
[TÀI LIỆU THAM KHẢO 53](#_Toc149057033)

# 1. Giới thiệu về HTML

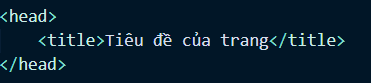
* 1. **Hướng dẫn về cú pháp và các phần tử HTML cơ bản.**
* Khai báo tài liệu HTML: Tài liệu HTML bắt đầu bằng dòng khai báo sau:



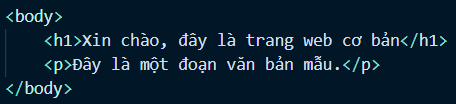
* Khai báo tài liệu HTML: Tài liệu HTML bắt đầu bằng dòng khai báo sau:



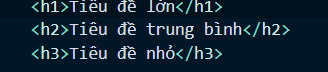
* Phần đầu (Head): Thẻ <head>...</head> chứa thông tin về tài liệu như tiêu đề, tập tin CSS, tập tin JavaScript, và các siêu liên kết.



* Phần thân (Body): Thẻ <body>...</body> chứa nội dung chính của trang web. html



* Tiêu đề (Headings): Thẻ từ <h1>...</h1> đến <h6>...</h6> dùng để định dạng tiêu đề với <h1> là tiêu đề lớn nhất và <h6> là tiêu đề nhỏ nhất.



* Đoạn văn bản (Paragraphs): Thẻ <p>...</p> dùng để tạo đoạn văn bản.



* Hình ảnh (Images): Để chèn hình ảnh, bạn sử dụng thẻ <img>. Thuộc tính src là đường dẫn đến hình ảnh và alt là văn bản mô tả hình ảnh (được sử dụng khi hình ảnh không thể hiển thị).



* Siêu liên kết (Hyperlinks): Để tạo liên kết đến trang web khác, bạn sử dụng thẻ <a>. Thuộc tính href chứa đường dẫn đến trang web mục tiêu.



* Bình luận (Comments): Bạn có thể thêm bình luận trong mã nguồn HTML bằng <!-- và -->.



* 1. **Hướng dẫn tạo một tệp HTML cơ bản**

Bước 1: Mở trình soạn thảo văn bản: Bạn có thể sử dụng các trình soạn thảo văn bản phổ biến như Notepad (trên Windows), Visual Studio Code, Sublime Text, hoặc bất kỳ trình soạn thảo nào bạn ưa thích.

Bước 2: Tạo file với đuôi .html

Bước 3: Trong file vừa tạo, thêm phần nội dung, văn bản, hình ảnh và các phần khác của trang.



Bước 4: Lưu tệp.

Bước 5: Sau khi bạn đã lưu tệp HTML, hãy mở trình duyệt web và kéo và thả tệp HTML vào trình duyệt hoặc sử dụng một trình duyệt để mở tệp.

* 1. **Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết**

Để tạo một trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết, bạn có thể sử dụng các thẻ HTML cơ bản như <html>, <head>, <body>, <h1>, <p>, <img>, và <a>.

Ví dụ:



# 2. Cơ bản về CSS

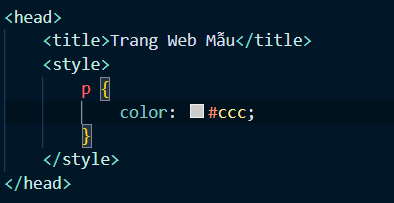
## **2.1 Hướng dẫn cách thêm kiểu cho trang web bằng CSS.**

Có 3 cách kết nối style CSS với trang web:

Cách 1: Kết nối bằng cách sử dụng thẻ <link>



Cách 2: Kết nối bằng cách sử dụng thẻ <style> trong phần <head>



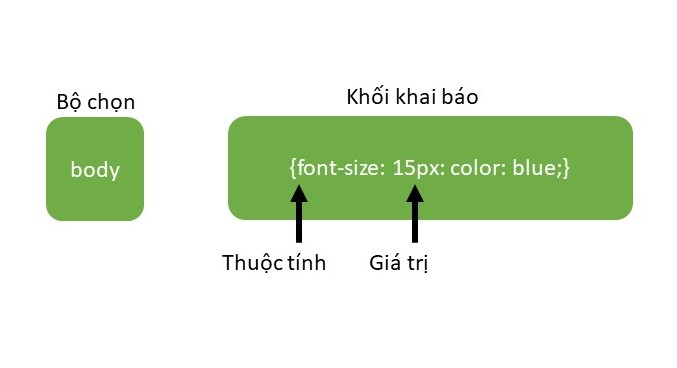
Cách 3: Kết nối bằng cách sử dụng thuộc tính style trực tiếp trên phần tử HTML



## **2.2 Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn.**

**\* Cú pháp CSS**

Quy tắc CSS được thiết lập bao gồm một bộ chọn (selector) và một khối khai báo (declaration).. Bộ chọn trỏ tới phần tử HTML mà bạn muốn định kiểu. Khối khai báo chứa một hoặc nhiều khai báo ngăn cách nhau bởi dấu chấm phảy. Mỗi khai báo bao gồm một tên thuộc tính và một giá trị, ngăn cách nhau bởi dấu hai chấm.



**\* Bộ chọn của CSS – CSS Selectors**

**a. Bộ chọn dựa vào tên thẻ – Element Selector**

Ví dụ .body{}, .h1{},…

Kiểu vùng chọn này là đơn giản nhất, đó là nó sẽ chọn toàn bộ các phần tử trên tài liệu HTML dựa vào tên thẻ có trong tài liệu rồi áp dụng CSS. Với kiểu sử dụng vùng chọn này thì toàn bộ các thẻ HTML trong website được chọn trong CSS đều biến đổi theo, tức là bạn không thể sử dụng kiểu chọn tên thẻ để viết CSS cho một khu vực độc lập.

**b. Bộ chọn dựa vào ID**

Vùng chọn dựa vào ID (tên định danh) nghĩa là bạn có thể chọn một phần tử cụ thể dựa vào giá trị của thuộc tính id trong thẻ HTML. Sở dĩ vùng chọn id được sử dụng để chọn một phần tử cụ thể là vì trên một trang tài liệu HTML thì mỗi phần tử phải mang một id riêng biệt không trùng nhau.

Id được thiết lập dựa vào thuộc tính id trong thẻ HTML và bất cứ thẻ nào cũng có thể sử dụng id. Khi viết tên id vào CSS thì nó phải có dấu thăng (#tên-id) đặt trước tên id để phân biệt với các loại vùng chọn khác. Ví dụ:

**c. Bộ chọn dựa vào Class**

Class (lớp) cũng rất được sử dụng phổ biến như id nhưng một điểm khác biệt của class là một class có thể được sử dụng cho nhiều phần tử trên một trang tài liệu HTML, còn id thì chỉ được sử dụng một lần duy nhất cho một phần tử.

Class được khai báo trong một phần tử HTML bởi thuộc tính class như <h1 class="tên-class">. Khi khai báo vùng chọn class trong CSS, thì tên class phải được đặt sau dấu chấm (.tên-class). Ví dụ

**d. Bộ chọn theo thứ cấp**

Kiểu vùng chọn này bạn cũng sẽ sử dụng rất thường xuyên, đặc biệt là khi tiến hành viết CSS cho website đó là chọn phần tử theo thứ cấp. Nghĩa là với vùng chọn này, bạn có thể chọn một phần tử con trong một phần tử mẹ nào đó.



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

**\* Các thuộc tính màu sắc:**

***- Color:*** dùng để thiết lập màu sắc cho văn bản.

***- Opacity:*** dùng để thiết lập mức độ trong suốt của 1 phần tử.

**\* Các thuộc tính màu nền và đường biên:**

- Background

- Background-attachment

- Background-color: thiết lập màu nền của 1 phần tử.

- Background-image

- Background-position

**\* Các thuộc tính text:**

***- Thuộc tính text-align:*** được sử dụng để căn chỉnh văn bản có thể căn chỉnh văn bản sang trái, phải hoặc chính giữa.

Nếu thuộc tính text-align được thiết lập giá trị là "justify", các dòng sẽ được kéo dài sao cho chiều rộng bằng nhau.

***- Thuộc tính text-decoration:*** được sử dụng để tạo hoặc xóa các dấu gạch ngang khỏi văn bản.

Giá trị text-decoration: none thường được sử dụng để xóa các dấu gạch chân khỏi các liên kết.

***- Thuộc tính text-transform:*** được sử dụng để chỉ định, thiết lập chữ hoa hoặc chữ thường cho text và còn được sử dụng để chuyển đổi text thành chữ in hoa hoặc chữ thường, hoặc viết hoa chữ cái đầu tiên của mỗi từ.

***- Thuộc tính text-indent:*** được sử dụng để chỉ định thụt dòng đầu tiên trong đoạn văn bản.

***- Thuộc tính letter-spacing:*** được sử dụng để chỉ định khoảng cách giữa các ký tự trong một từ.

***- Thuộc tính line-height:*** được sử dụng để chỉ định khoảng cách giữa các dòng trong đoạn văn bản.

***- Thuộc tính direction:*** được sử dụng để thay đổi hướng văn bản của một phần tử.

***- Thuộc tính word-spacing:*** được sử dụng để chỉ định khoảng cách giữa các từ trong một câu.

***- Thuộc tính text-shadow:*** được sử dụng để tạo hiệu ứng chữ đổ bóng cho text.

# 3. JavaScript Cơ bản

## 3.1. Giới thiệu cú pháp JavaScript cơ bản

**a. Khái niệm JavaScript**

- Javascript là ngôn ngữ lập trình web và có thể thêm mới và update cả HTML và CSS.

- Javascript không chỉ gói gọn trong các ứng dụng front end trên trình duyệt mà còn có thể phát triển cả ứng dụng cho desktop, mobile hay server.

- Javascript được hỗ trợ trên tất cả các trình duyệt như Firefox, Chrome, IE… Thậm chí hỗ trợ các trình duyệt trên thiết bị di động.

* jQuery – thư viện Javascript để đơn giản hóa việc xử lý DOM.
* ReactJS – thư viện phát triển ứng dụng web hiện đại.
* NodeJS – là nền tảng phát triển ứng dụng phía server bằng javascript.

**b. Cú pháp JavaScript cơ bản**

- Một chương trình máy tính là một tập danh sách các hướng dẫn (statements) để máy tính thực thi. Và Javascript cũng vậy nhưng khác một điều là các tập lệnh này cho trình duyệt thực hiện thay vì máy tính.

**\* Cách khai báo biến**

- Javascript hỗ trợ 3 cách để khai báo một biến, đó là sử dụng các từ khóa: let, var và const.

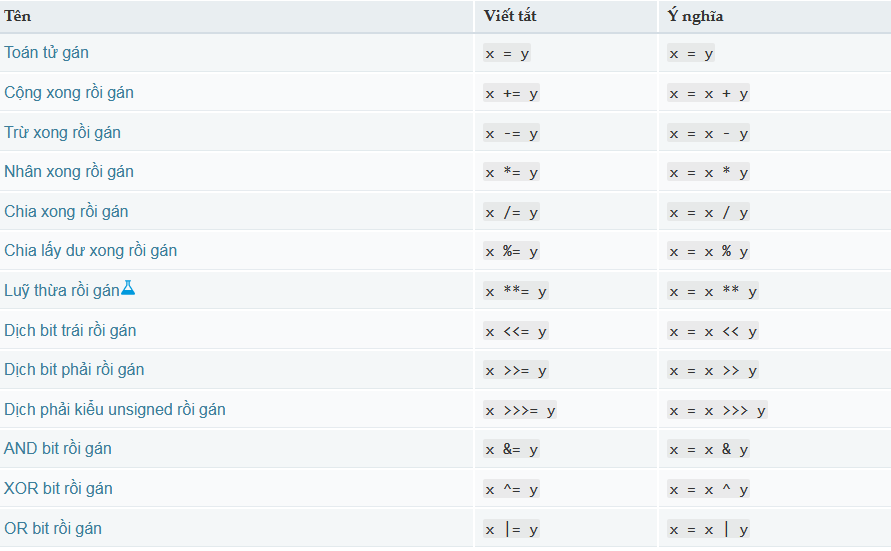
* Const: được dùng để khai báo hằng số, giá trị của nó không thay đổi trong suốt chương trình. Nó giống với từ khóa *final* của Java vậy.
* var: biến được khai báo với từ khóa *var* sẽ được truy cập ở bất kể đâu trong chương trình.
* let: từ khóa này giúp cho biến chỉ có thể được truy xuất bên trong phạm vi block quanh nó (block được định nghĩa bởi dấu ngoặc nhọn {} ).

**\* Kiểu dữ liệu của biến**

- Theo tiêu chuẩn ES7, có 7 kiểu dữ liệu:

* Kiểu boolean
* Kiểu null
* Kiểu undefined
* Kiểu số (int)
* Kiểu chuỗi (String)
* Kiểu Symbol (mới trong ECMAScript 6)
* Kiểu đối tượng (Object)

**\* Toán tử trong JavaScript**



## **3.2. Hướng dẫn tạo biến , hàm và sử dụng sự kiện**

**a. Tạo biến JavaScript**

- Khai báo và sử dụng biến trong JavaScript với var và let ,quy tắc đặt tên biến và phép toán gán ,khai báo hằng số với const trong js

-Biến dùng để lưu trữ giá trị các dữ liệu hay các đối tượng

-Giá trị của biến có thể được thay đổi cập nhật trong quá trình ứng dụng hoạt động (tên biến,khai báo và khởi tạo biến …)

- Quy tắc đặt tên biến :

+ Tên biến được tạo ra bởi các ký tự chữ , số , \_ và $

+ Tên không được phép bắt đầu bằng số ( chỉ bắt đầu bằng ký tự chữ , $ , \_ )

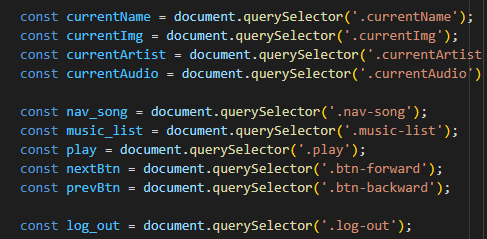
+ Không được chứa các ký hiệu đặc biệt như toán học , logic (như +,-,\*,>,<,…)

+ Không chứa khoảng trắng

+ Không được đặt tên biến trùng với các từ khóa dành riêng cho ngôn ngữ JavaScript liệt kê ở bảng sau

- Khái báo và khởi tạo biến

Lấy ví dụ về một phần bài tập nhóm làm ví dụ ta có :



**b. Tạo hàm JavaScript**

function renderContent(music) {

    let htmls = music.map((song) => {

        return `

                <div class="m-4" style="width: 200px;">

                  <div style="position: relative;">

                      <img src="${song.image}" class="rounded-3 img-fluid" style="object-fit: cover; width: 200px; height: 200px;">

                        <a href="#">

                          <div class="song-hover">

                            <i class="h1 bi bi-play" style="left: 50%;top: 50%; position: absolute; transform: translate(-50%, -50%);"></i>

                            </div>

                        </a>

                    </div>

                <div>

                    <div class="lead text-white">

                        ${song.title}

                    </div>

                    <small class="text-white">${song.artist}</small>

                    </div>

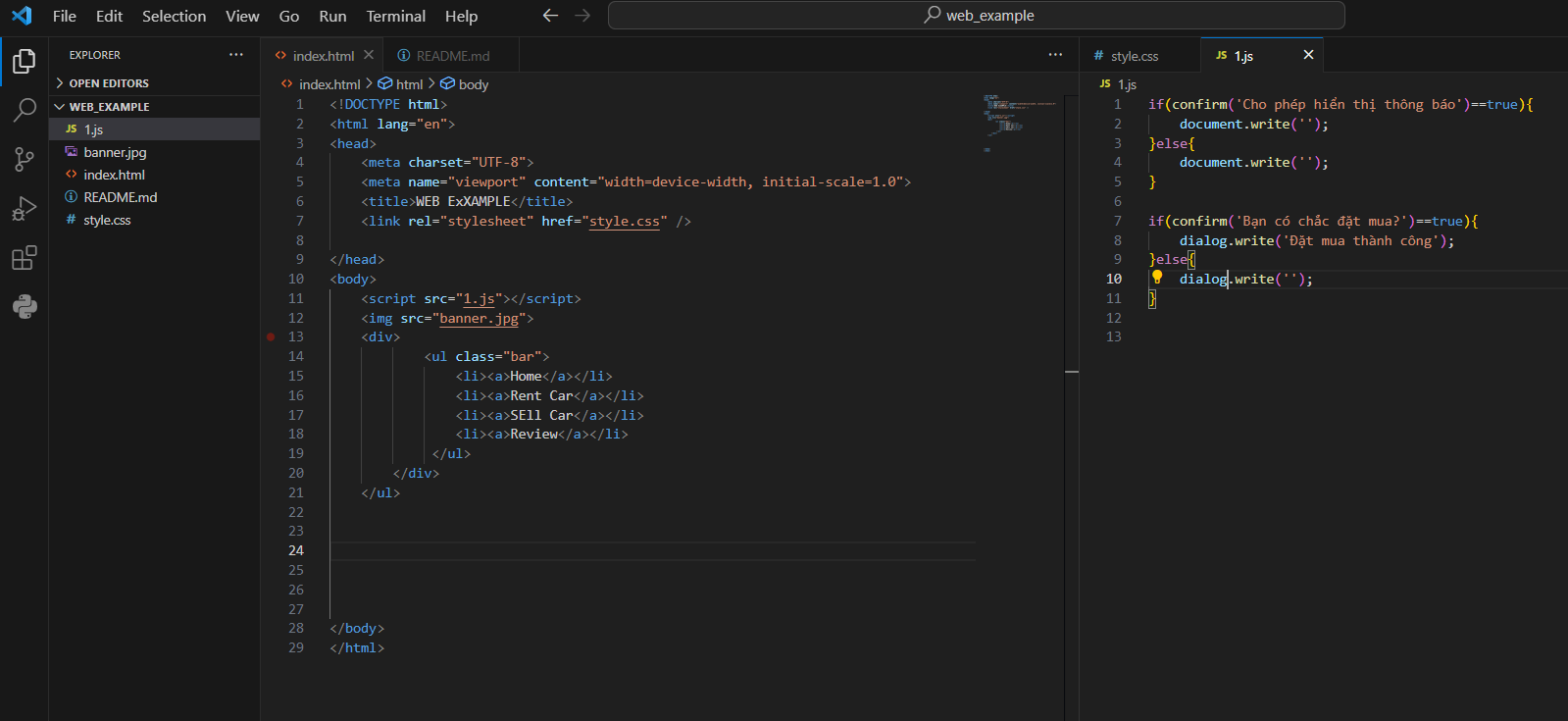
                </div>

        `

    })

Hàm được tạo ra có nghĩa là tên thuộc tính của hàm chứa tên đối tượng. Nó rất hữu ích khi xem cuộc gọi: trong quá trình gỡ lỗi hoặc đọc thông báo lỗi.

## **3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn.**



- Ví dụ: sử dụng js để kiểm tra form và hiện thông báo



# 4. HTML Forms và Input Elements

## **4.1. Hướng dẫn cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi.**

- Phần tử HTML <form> xác định một biểu mẫu dùng để thu thập thông tin người dùng nhập vào. Các phần tử form là các kiểu phần tử nhập khác nhau như trường nhập văn bản, hộp kiểm, nút radio, nút gửi,v.v… Dưới đây là các thanh phần :

**Cấu trúc form**

<form action="process.php" method="get">

<!--Các phần tử form-->

</form>

Trong đó:

* <form> ... </form>: Thẻ khai báo một form.
* action: Thuộc tính khai báo đường dẫn xử lý dữ liệu sau khi người dùng submit form.
* method: Thuộc tính khai báo phương thức xử lý dữ liệu bao gồm POST và GET.

Lưu ý: Phần action và method qua nội dung php bạn sẽ được nắm rõ hơn để khai báo. Còn trong quá trình xây dựng html tạm thời bạn cứ để trống.

**Các phần tử form thường dùng**

**Input text – nhập đoạn text ngắn**

Đây là phần tử form dùng nhập text có số lượng ký tự ngắn như tên người dùng, địa chỉ, tiêu đề bài viết – sản phẩm…

Ví dụ bên dưới chúng ta tạo ra input nhập username

<form action="" method="">

Username: <input type="text" name="username">

</form>

**Input password – nhập mật khẩu**

Để lập những thông tin cần bảo mật như mật khẩu trong form có type = "password" giúp chúng ta nhập thông tin một cách an toàn.

<form action="" method="">

Password: <input type="password" name="password">

</form>

**Input submit – gửi dữ liệu lên server**

Khi đã nhập dữ liệu vào form để gửi nó lên server chúng ta cần khai báo input type = "submit". Khi người dùng nhấn vào nút submit dữ liệu sẽ được gửi lên server để xử lý.

Bên dưới là demo form đăng nhập bao gồm username, password, và một nút bấm đăng nhập.

<form action="" method="">

Username: <input type="text" name="username"><br>

Password: <input type="password" name="password"><br>

<input type="submit" name="btn\_login" value="Đăng nhập">

</form>

**Input email – nhập địa chỉ email**

Trong quá trình đăng ký tài khoản hoặc các form đăng ký nhận bản tin, giảm giá thì email là một phần không thể thiếu.

Để nhập liệu email chúng ta sử dụng input type = "email"

<form action="" method="">

<input type="email" name="email">

</form>

**Input date – nhập ngày tháng**

<form action="" method="">

<input type="date" name="birthday">

</form>

**Textarea – nhập đoạn văn**

Textarea thường dùng để nhập liệu đoạn văn dài như mô tả bài viết, nội dung bài viết, nội dung sản phẩm…

<form action="" method="">

<textarea name="post\_content" cols="30" rows="10"></textarea>

</form>

**Trong đó**

* cols: Độ rộng của ô nhập liệu
* rows: Độ cao của ô nhập liệu

**Input number – nhập số lượng**

Trong một số trường hợp dữ liệu nhập vào ở dạng số nguyên bạn có thể sử dụng type = "number". Phần này thường dùng trong chọn số lượng sản phẩm trong giỏ hàng.

<form action="" method="">

<input type="number" name="num\_order" min="1" value="1" max="10">

</form>

Trong đó

* min: Thuộc tính khai báo giá trị bé nhất của dữ liệu.
* max: Thuộc tính khai báo gía trị lớn nhất của dữ liệu.
* value: Thuộc tính khai báo giá trị mặc định.

**Checkbox – hộp chọn nhiều**

Trong trường hợp muốn khảo sát thông tin từ người dùng về một chủ đề có nhiều lựa chọn. Người dùng có thể chọn một hoặc nhiều option thì chúng ta sử dụng input = "checkbox".

Ví dụ bên dưới giúp người dùng khai báo kỹ năng của họ bao gồm html, css, php.

Lưu ý: Các checkbox có cùng một chủ đề bạn cần sử dụng có cùng giá trị của thuộc tính name

<form action="" method="">

<input type="checkbox" name="skill" value="html"> Html<br>

<input type="checkbox" name="skill" value="css"> Css<br>

<input type="checkbox" name="skill" value="php"> Php<br>

</form>

**Radiobox – hộp chọn một trong số**

Khi muốn lấy thông tin của người dùng chỉ được chọn một trong số rất nhiều lựa chọn chúng ta sử dụng input type="radio".

Ví dụ bên dưới giúp người dùng chọn giới tính của mình.

<form action="" method="">

<input type="radio" name="gender" value="male">Nam<br>

<input type="radio" name="gender" value="female">Nữ<br>

</form>

**Selectbox – chọn qua thanh xổ**

Một chức năng thường dùng select chính là thanh chọn vị trí tỉnh/thành phố đang ở.

<form action="" method="">

<select name="city">

<option value="">Hà Nội</option>

<option value="">Hải Phòng</option>

<option value="">Đà Nẵng</option>

<option value="">TP.HCM</option>

<option value="">Tỉnh khác</option>

</select>

</form>

## **4.2. Tạo kiểu cho biểu mẫu bằng CSS.**

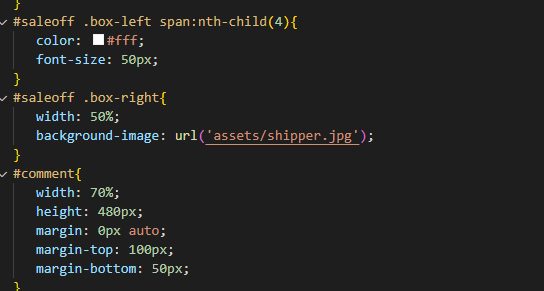
**a. Khái niệm**

- Biểu mẫu (Form) là đối tượng giúp ích cho việc nhập và hiển thị dữ liệu, thông tin một cách trực quan, có hệ thống, có thể thống kê dễ thuận tiện cho việc điều khiển một ứng dụng, hoặc thực hiện một chức năng nào đó.

**b. Tạo kiểu cho biểu mẫu HTML bằng CSS**

- Sử dụng thuộc tính *width* để xác định rõ chiều rộng của trường đầu vào:

+ Ví dụ: áp dụng cho tất cả các phần tử <input> :



- Định dạng *padding* cho trường đầu vào:

+ Sử dụng thuộc tính padding để thêm khoảng trống bên trong trường văn bản.

+ Khi có nhiều trường đầu vào khác nhau có thể thêm margin, để thêm khoảng trống bên ngoài cho trường đầu vào.

+ Đặt thuộc tính box-sizing là border-box để đảm bảo phần padding và border cuối cùng được đặt trong tổng chiều rộng và chiều cao của các phần tử.

- Định dạng đường viền cho trường đầu vào:

+ Sử dụng thuộc tính *border* để thay đổi kích thước và màu sắc đường viền

+ Sử dụng thuộc tính *border-radius* để bo góc tròn

+ Nếu chỉ muốn thêm đường viền dưới cùng thì sử dụng thuộc tính *border-bottom:*

- Định dạng màu cho trường đầu vào:

+ Sử dụng thuộc tính *background-color* để thêm màu nền cho trường đầu vào, và thuộc tính *color* thay đổi màu văn bản:

- Thêm biểu tượng/ảnh vào bên trong trường đầu vào:

+ Sử dụng thuộc tính *background-image* và thuộc tính vị trí *background-position* để thêm icon vào trong trường đầu vào. Chú ý: thêm *padding-left* để dành khoảng trống cho biểu tượng:

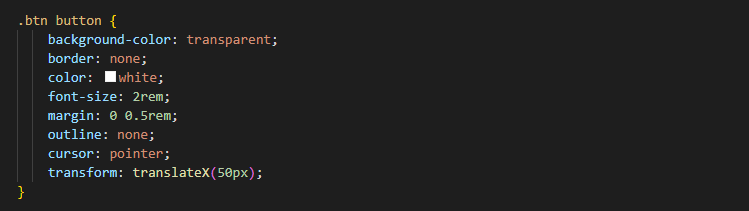
- Thêm hiệu ứng tìm kiếm cho trường đầu vào:

+ Sử dụng thuộc tính *transition* trong CSS để tạo hiệu ứng chiều rộng cho trường tìm kiếm khi người dùng click chuột vào trường nhập văn bản.

- Tạo kiểu cho Textareas:

+ Sử dụng thuộc tính *resize* để ngăn các vùng văn bản bị thay đổi kích thước ( tắt “grabber” ở góc dưới cùng bên phải):

- Định dạng button:

.

## **4.3. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu (vd: kiểm tra dữ liệu hợp lệ).**

### **a. Truy cập đến form dùng javascript**

Bằng cách sau :

+ Truy xuất đến form theo chỉ số : document.form[ chỉ số]

+ Truy xuất đến form theo tên : document.form[‘tênform’]

+ Truy xuất đến form theo cú pháp ngắn gọn : document.tênform

+ Ví dụ :

<form name="timkiem">

<input name="tukhoa" value="Từ khóa" />

</form>

<form name="dangnhap">

<input name="un" value="teonv" />

<input name="pw" value="123" />

</form>

<script>

console.log(document.timkiem.tukhoa.value); //Từ khóa

console.log(document.forms[1].un.value); //teonv

console.log(document.forms['dangnhap'].pw.value); //123

</script>

### **b.Truy cập đến các control trong form để lấy value**

+getElementsByName: documnet.getElementsByName(‘namecủaControl’)[chỉ số].value;

+ Truy cập thông qua form :

Document.tênForm.tênControl.value;

+ Ví dụ :

<form name="tk">

<input type="text" name="tukhoa" id="kw" value="Từ khóa">

<button type="submit">Tìm kiếm</button>

</form> <hr>

<script>

t = document.tk.tukhoa.value; //Từ khóa

t = document.forms[0].tukhoa.value; //Từ khóa

t = document.getElementById("kw").value; //Từ khóa

</script>

### **c. Kiểm tra dữ liệu trong Form với JavaScript**

- Là kiểm tra xem thông tin có được điền vào form đúng yêu cầu hay chưa. Ví dụ tuổi hay số lượng sản phẩm phải là kiểu số nguyên, ngàysinh được được là chữ, mã sinh viên phải dài 7 ký tự …. Nhờ kiểm tra và báo lỗi, user được thông báo và nhập lại cho đúng, khi đó thông tin được xử lý và lưu trữ mới mới có giá trị

- Thực hiện kiểm tra dữ liệu trong form với javascript : Hàm gọi trong sự kiện onSubmit của tag form để kiểm tra dữ liệu nếu không hợp lệ thì trả về false ,còn nếu hợp lệ thì trả về true.

- Ví dụ :

<form name="frmlogin" action="xuly.php" **onsubmit="kiemtra()"**>

<div>

<label>Username</label> <input type="text" name="u" id="un">

</div>

<div>

<label>Password</label> <input type="password" name="p" id="pw">

</div>

<div>

<button type="submit">Đăng nhập</button>

</div>

<h4 id="baoloi"></h4>

</form>

# 5. Quy trình phát triển web

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

**\* Quy trình phát triển web :**

**- B1: Thu thập thông tin**

Giai đoạn này tập trung tìm hiểu và nghiên cứu về yêu cầu khách hàng để quyết định các bước tiếp theo sẽ thực hiện như thế nào. Nhiệm vụ quan trọng nhất trong bước này là hiểu rõ về mục đích của website, các chức năng website sẽ cung cấp, nhóm người dùng mục tiêu của website và thu thập đầy đủ các yêu cầu từ phía khách hàng, người dùng. Trả lời những câu hỏi trên sẽ giúp chúng ta lựa chọn được phương án tốt nhất để triển khai dự án

**- B2: Lập kế hoạch**

Ở bước này, lập trình viên sẽ đưa cho khách hàng cái nhìn tổng quan về website bằng các sitemap, wireframe hoặc bảng biểu dựa trên các thông tin đã thu thập được ở bước 1.

**- B3: Thiết kế giao diện**

Trong giai đoạn thiết kế, trang web sẽ trực quan hơn với các nút, hình ảnh, video,... sẽ được tạo , tất cả thông tin được thu thập trong giai đoạn đầu tiên là rất quan trọng. Các yêu cầu của khách hàng và trải nghiệm người dùng phải luôn được ghi nhớ trong khi thực hiện thiết kế.

**- B4 : Viết tài liệu đặc tả SRS**

SRS là tài liệu được sử dụng để mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống. Tài liệu này sẽ hỗ trợ đưa ra các tính năng của hệ thống hoặc dùng cho việc đọc hiểu hệ thống của phía khách hàng.

Đây là một tài liệu quan trọng cho đội phát triển (system analyst, business, analyst, code) và kiểm thử (tester). Ngoài ra, dựa vào các yêu cầu trong SRS, ta có thể đánh giá được số lượng scope, thời gian hoàn thành và các chi phí trong quá trình phát triển sản phẩm một cách nhanh chóng và dễ dàng hơn.

**- B5: Viết mã nguồn**

Bắt tay vào tạo trang web. Trước tiên khi bắt tay vào viết mã nguồn, cần lựa chọn công nghệ, ngôn ngữ lập trình phù hợp cho dự án.

**- B6: Kiểm thử , đánh giá và ra mắt sản phẩm**

Quá trình kiểm thử là quá trình thường xuyên nhất của một quy trình. Mọi liên kết nên được kiểm tra để đảm bảo rằng không có liên kết nào bị lỗi hay gián đoạn. Nên kiểm tra mọi biểu mẫu, mọi câu lệnh và kiểm tra lỗi chính tả của toàn bộ trang web. Sử dụng các tiêu chuẩn web chung để kiểm tra xem mã nguồn có đảm bảo sự tương thích giữa các trình duyệt hay không.

**- B7 : Bảo trì**

Điều quan trọng cần lưu ý, hay coi trang web là một dịch vụ hơn là một sản phẩm. Nên đảm bảo mọi thứ đều hoạt động tốt, mọi người đều hài lòng và luôn chuẩn bị sẵn sàng để thực hiện các thay đổi trong trường hợp khác.

## **5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát triển, kiểm thử, và triển khai.**

- Công nghệ web thường tuân theo quy trình phát triển phần mềm chung và các bước quan trọng trong quy trình phát triển công nghệ web là:

+ Phân tích yêu cầu: Đây là giai đoạn đầu tiên và quan trọng nhất trong quy trình phát triển. Tại giai đoạn này, cần hiểu rõ yêu cầu của khách hàng, xác định tính năng và chức năng cần có trong ứng dụng web.

+ Thiết kế giao diện: Sau khi có được yêu cầu của khách hàng cần thiết kế giao diện người dùng (UI/UX) cho ứng dụng web. Điều này bao gồm việc xác định các trang và luồng điều hướng giữa chúng.

+ Phát triển: Ở bước này cần triển khai mã nguồn để xây dựng chức năng của ứng dụng web. Có thể sử dụng các ngôn ngữ lập trình phù hợp như HTML, CSS, JavaScript, PHP, Ruby, Python, Java,… để phát triển ứng dụng.

+ Kiểm thử: Sau khi hoàn thành phát triển, chúng ta cần tiến hành kiểm thử ứng dụng. Kiểm thử bao gồm kiểm tra tính năng, tương thích, hiệu suất và bảo mật của ứng dụng web.

+ Triển khai: Khi kiểm thử thành công, triển khai ứng dụng web lên môi trường hoạt động thực tế. Điều này có thể là máy chủ chạy trên internet hoặc một môi trường đám mây như AWS, Google Cloud...

+ Bảo trì và nâng cấp: Sau khi triển khai, cần duy trì và cải tiến ứng dụng web theo yêu cầu của khách hàng. Bảo trì và nâng cấp bao gồm khắc phục lỗi, bảo mật và cập nhật tính năng mới cho ứng dụng.

- Ngoài ra, quy trình phát triển công nghệ web còn có thể bao gồm các bước khác như thu thập phản hồi từ người dùng, tối ưu hóa hiệu suất và tăng cường bảo mật. Các bước này có thể thay đổi phụ thuộc vào quy mô và phạm vi của dự án quản lý dự án và phát triển.

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

**Quản lý dự án:**

Trưởng nhóm dự án: Chịu trách nhiệm về lập kế hoạch, giám sát và điều phối công việc của toàn bộ nhóm. Đảm bảo dự án được triển khai đúng tiến độ và chất lượng.

**Phân tích:**

Nhân viên phân tích dự án: Nghiên cứu, phân tích và đặc tả yêu cầu của dự án. Liên hệ và làm việc với khách hàng để hiểu rõ các yêu cầu và mong muốn.

**Thiết kế:**

* Nhân viên thiết kế giao diện: Tạo ra thiết kế giao diện dự án dựa trên yêu cầu và sự hỗ trợ từ phân tích dự án.
* Nhân viên thiết kế hệ thống: Thiết kế kiến trúc hệ thống và quyết định về cấu trúc, giao thức và công nghệ phù hợp.

**Phát triển:**

* Nhân viên lập trình: Xây dựng và triển khai phần mềm, sáng tạo giải pháp kỹ thuật dựa trên yêu cầu và thiết kế.
* Nhân viên kiểm thử: Thực hiện các kiểm thử phần mềm, đảm bảo rằng phần mềm hoạt động đúng và đáp ứng yêu cầu.

**Quản lý chất lượng:**

Nhân viên quản lý chất lượng: Theo dõi chất lượng và hiệu suất của phần mềm, thực hiện các kiểm tra và đánh giá kỹ thuật.

**Quản lý tiến độ:**

Nhân viên quản lý dự án: Theo dõi tiến độ dự án, quản lý nguồn lực và giải quyết các vấn đề phát sinh trong quá trình thực hiện dự án.

**Hỗ trợ và bảo trì:**

* Nhân viên hỗ trợ: Cung cấp hỗ trợ kỹ thuật, giải quyết sự cố và đào tạo người dùng cuối.
* Nhân viên bảo trì: Đảm bảo phần mềm hoạt động ổn định sau khi triển khai, thực hiện bảo trì và cập nhật.

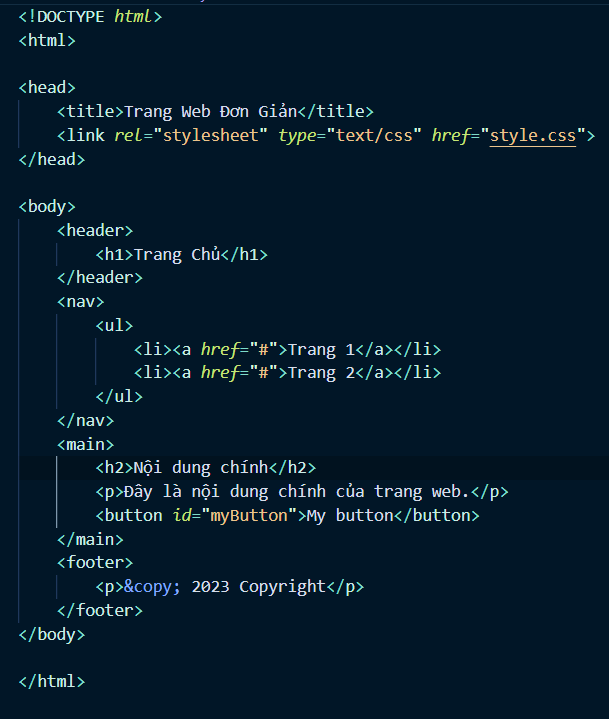
# 6. Thực hành xây dựng trang web đơn giản:

## **6.1. Xây dựng trang web đơn giản từ đầu bằng việc kết hợp HTML, CSS và JavaScript.**

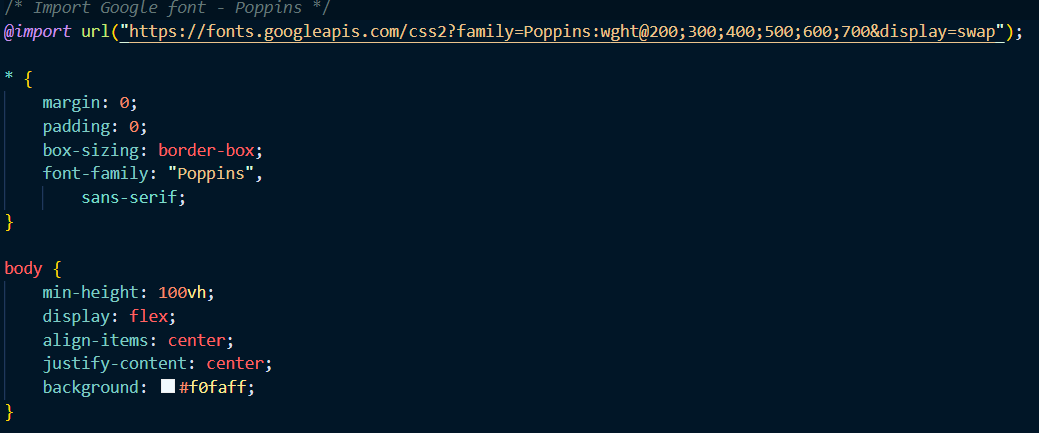
Bước 1: Xác định cấu trúc trang Bắt đầu bằng việc xác định cấu trúc cơ bản của trang web của bạn. Thông thường, một trang web sẽ có các phần sau:

* Head: Chứa các thông tin về trang web như tiêu đề, thẻ <meta>, và các tệp CSS và JavaScript liên kết.
* Menu: Đây là nơi bạn đặt các liên kết đến các trang con hoặc các phần của trang web.
* Content: Phần chính của trang web, nơi hiển thị nội dung chính, hình ảnh, văn bản, v.v.
* Footer: Chứa thông tin liên hệ hoặc bất kỳ thông tin nào bạn muốn đặt ở cuối trang.

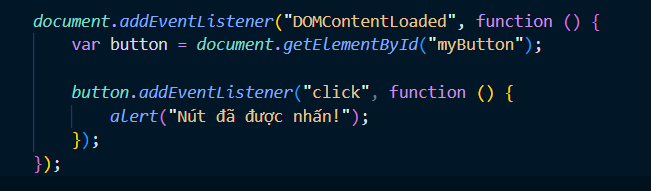
Bước 2: Sử dụng HTML Sử dụng HTML để tạo cấu trúc trang web. Dưới đây là một ví dụ đơn giản:



Bước 3: Sử dụng CSS Sử dụng CSS để tạo kiểu cho trang web của bạn. Tạo một tệp style.css để định dạng giao diện. Dưới đây là một ví dụ đơn giản:



Bước 4: Sử dụng JavaScript Thêm tính năng cơ bản bằng JavaScript. Dưới đây là một ví dụ đơn giản:



Bước 5: Kết hợp tất cả lại Liên kết các tệp HTML, CSS và JavaScript vào tài liệu HTML chính để hiển thị trang web của bạn.

## **6.2. Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL): Bài thi**

# 7. Responsive Web Design:

## **7.1. Giới thiệu về cách tạo trang web phản hồi và thân thiện với nhiều thiết bị.**

Tạo một trang web phản hồi và thân thiện với nhiều thiết bị là một phần quan trọng của phát triển web hiện đại. Điều này đảm bảo rằng trang web của bạn sẽ được hiển thị đẹp và hoạt động trên nhiều thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động và máy tính bảng. Dưới đây là một số nguyên tắc và kỹ thuật để tạo trang web phản hồi và thân thiện với nhiều thiết bị:

1. Sử dụng HTML5 và CSS3:

Sử dụng các thẻ HTML5 như <header>, <nav>, <main>, <footer>,... để tạo cấu trúc trang web mạnh mẽ và dễ dàng kiểm soát. Sử dụng CSS3 để tạo kiểu cho trang web và sử dụng tính năng mới như media queries để điều chỉnh giao diện dựa trên kích thước màn hình.

1. Media Queries:

Sử dụng media queries để thay đổi kiểu dáng và bố cục của trang web dựa trên kích thước màn hình.

1. Thiết kế linh hoạt:

Hãy thiết kế trang web của bạn để có thể thích nghi với nhiều kích thước màn hình khác nhau thay vì chỉ tập trung vào một kích thước cố định.

1. Ảnh phản hồi:

Sử dụng ảnh phản hồi (responsive images) để tối ưu hóa hiển thị ảnh trên nhiều thiết bị. Sử dụng thuộc tính srcset và thẻ <picture> để điều chỉnh kích thước và chất lượng ảnh dựa trên kích thước màn hình.

1. Thử nghiệm trên nhiều thiết bị:

Sử dụng công cụ kiểm tra trình duyệt và thiết bị để đảm bảo rằng trang web của bạn hoạt động mượt mà trên tất cả các loại thiết bị và trình duyệt khác nhau.

1. Tối ưu hóa tốc độ tải trang:

Sử dụng nén và tối ưu hóa hình ảnh, sử dụng tệp CSS và JavaScript siêu nhẹ, và sử dụng bộ nhớ cache để giảm thời gian tải trang.

1. Kiểm tra thường xuyên:

Kiểm tra trang web trên nhiều thiết bị và thường xuyên cập nhật nó để đảm bảo tính phản hồi và hiệu suất tốt.

1. Thử nghiệm với các trình duyệt khác nhau:

Đảm bảo trang web của bạn hoạt động tốt trên nhiều trình duyệt khác nhau bằng cách kiểm tra và sửa lỗi đối với từng trình duyệt.

Tạo một trang web phản hồi và thân thiện với nhiều thiết bị đòi hỏi sự cân nhắc và công sức, nhưng nó đảm bảo rằng trải nghiệm người dùng sẽ tốt trên mọi nền tảng và thiết bị.

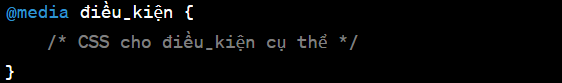
## **7.2. Sử dụng các kỹ thuật CSS media queries để điều chỉnh giao diện cho các màn hình khác nhau**

CSS media queries là một công cụ mạnh mẽ trong phát triển web, cho phép bạn điều chỉnh giao diện của trang web dựa trên điều kiện như kích thước màn hình, tỷ lệ khung hình, chế độ in, vị trí của thiết bị và nhiều yếu tố khác. Chúng cho phép bạn tạo trang web phản hồi (responsive) và thân thiện với nhiều thiết bị khác nhau, bao gồm máy tính để bàn, điện thoại di động và máy tính bảng.

Dưới đây, nhóm em sẽ giải thích chi tiết cách sử dụng media queries và ví dụ về cách bạn có thể sử dụng chúng để tùy chỉnh giao diện cho các màn hình khác nhau:

**Cú pháp cơ bản của Media Queries**:

Cú pháp cơ bản của media queries bao gồm:

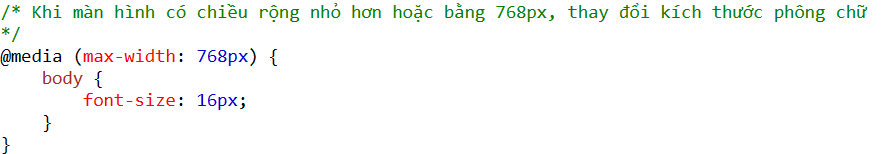


* @media: Đây là quy tắc khai báo media queries.
* điều\_kiện: Điều kiện xác định khi nào CSS bên trong sẽ được áp dụng.

**Media Query dựa trên kích thước màn hình:**

Một trong những cách phổ biến để sử dụng media queries là dựa trên kích thước màn hình.

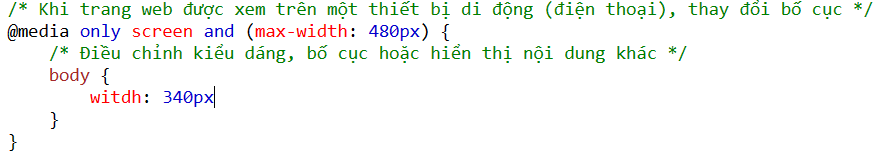
Ví dụ:



Trong ví dụ này, CSS bên trong media query sẽ chỉ được áp dụng khi chiều rộng của màn hình nhỏ hơn hoặc bằng 768px.

**Media Query dựa trên loại thiết bị:**

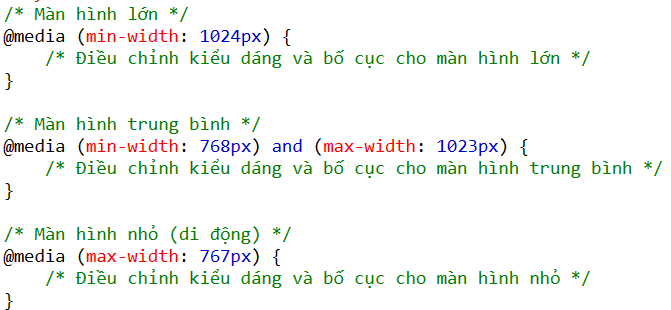
Bạn cũng có thể sử dụng media queries để xác định loại thiết bị. Ví dụ:



Ở đây, media query chỉ áp dụng cho các thiết bị di động (điện thoại) với chiều rộng màn hình tối đa là 480px.

**Kết hợp Media Queries:**

Bạn có thể kết hợp nhiều điều kiện trong media queries để tạo giao diện đa dạng cho nhiều loại thiết bị:



Các điều kiện trong các media queries có thể được tùy chỉnh để phù hợp với nhu cầu thiết kế của bạn.

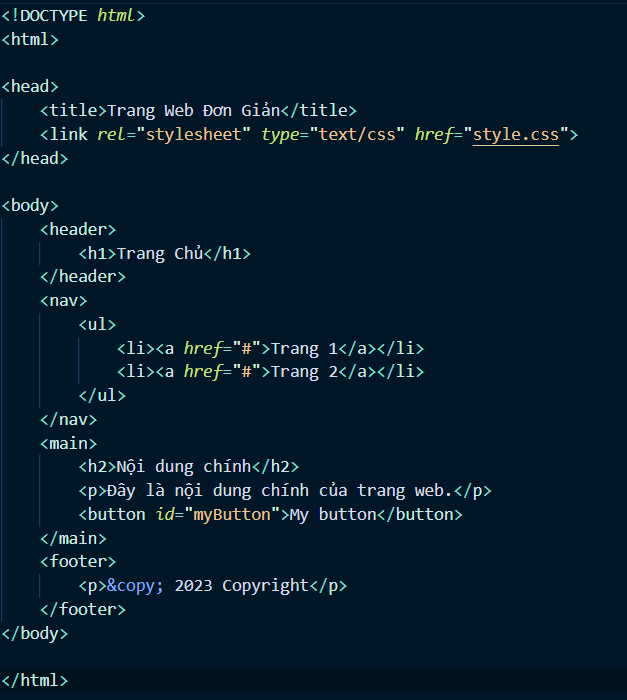
Media queries giúp bạn tạo trang web phản hồi và thân thiện với nhiều thiết bị khác nhau bằng cách đảm bảo rằng giao diện sẽ hiển thị một cách tốt nhất trên từng loại màn hình hoặc thiết bị.

## **7.3 Hướng dẫn cách tạo bố cục đáng linh hoạt cho các thiết bị khác nhau.**

Để tạo một bố cục đáng linh hoạt cho các thiết bị khác nhau, bạn có thể sử dụng CSS và media queries để thích nghi với kích thước màn hình và loại thiết bị. Dưới đây là hướng dẫn cụ thể về cách tạo bố cục đáng linh hoạt:

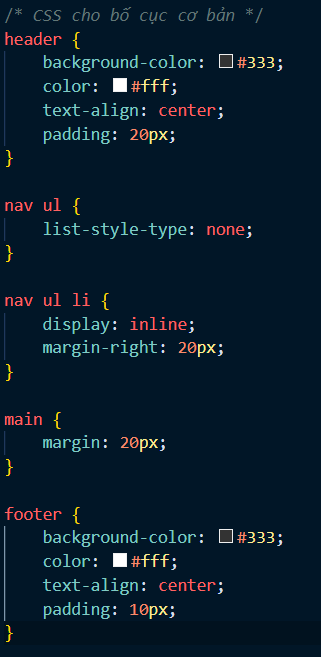
**Bước 1**: Thiết lập cơ sở

Trước tiên, hãy xác định một cơ sở CSS cho trang web của bạn. Đảm bảo rằng bạn đã sử dụng các thẻ HTML5 để tạo cấu trúc trang và áp dụng CSS để định dạng các phần tử trang. Dưới đây là một ví dụ cơ bản:



**Bước 2**: Sử dụng CSS cho Bố cục Cơ bản

Trong tệp style.css, bạn có thể áp dụng CSS để tạo bố cục cơ bản cho trang web. Đảm bảo rằng bạn đã sử dụng các thẻ HTML và các lớp hoặc ID để chọn các phần tử mà bạn muốn tùy chỉnh. Dưới đây là một ví dụ đơn giản:



**Bước 3**: Sử dụng Media Queries

Sử dụng media queries để điều chỉnh bố cục cho các thiết bị khác nhau hoặc kích thước màn hình khác nhau. Dưới đây là ví dụ sử dụng media queries để thay đổi bố cục cho các màn hình có chiều rộng nhỏ hơn 768px:



Trong ví dụ này, bố cục của header và nav sẽ thay đổi khi màn hình có chiều rộng nhỏ hơn 768px. nav sẽ chuyển thành hiển thị dọc và các mục li trong nav sẽ được căn giữa.

**Bước 4**: Kiểm Tra và Tinh Chỉnh

Cuối cùng, hãy kiểm tra trang web của bạn trên nhiều thiết bị và kích thước màn hình khác nhau để đảm bảo rằng bố cục đáng linh hoạt hoạt động đúng cách và đảm bảo rằng trải nghiệm người dùng là tốt nhất.

Tóm lại, để tạo bố cục đáng linh hoạt cho các thiết bị khác nhau, bạn cần sử dụng CSS để tạo bố cục cơ bản, sau đó sử dụng media queries để thay đổi bố cục dựa trên kích thước màn hình hoặc loại thiết bị.

# 8.HTML5 và CSS3

## **8.1 Giới thiệu về các tính năng mới trong HTML5 và CSS3.**

HTML5 và CSS3 đã mang đến nhiều tính năng mới và cải tiến mạnh mẽ cho phát triển web. Dưới đây là một số tính năng quan trọng của cả hai:

HTML5:

* Thẻ mới: HTML5 đã giới thiệu nhiều thẻ mới để làm cho cấu trúc trang web trở nên dễ đọc hơn và hiệu quả hơn, bao gồm <header>, <nav>, <article>, <section>, <footer>, và nhiều thẻ khác.
* Multimedia: HTML5 hỗ trợ video và âm thanh trực tiếp bằng thẻ <video> và <audio> mà không cần sử dụng plugins bên ngoài như Flash.
* Canvas: <canvas> cho phép tạo và vẽ đồ họa, biểu đồ và hoạt hình trong trình duyệt.
* Web Storage: Hỗ trợ Local Storage và Session Storage để lưu trữ dữ liệu nhanh chóng trên máy tính của người dùng mà không cần server.
* Geolocation: HTML5 cho phép trang web xác định vị trí địa lý của người dùng thông qua Geolocation API.
* Drag and Drop: Hỗ trợ kéo và thả phần tử trên trang web bằng sử dụng API Drag and Drop.
* Form Input Types: HTML5 giới thiệu các loại mới cho các trường đầu vào như email, url, date, number, và nhiều loại khác giúp kiểm tra dữ liệu dễ dàng hơn.

CSS3:

* CSS Selectors: CSS3 cung cấp các bộ chọn mạnh mẽ như pseudo-classes (:nth-child, :hover, :not) và pseudo-elements (::before, ::after) để lựa chọn và định dạng các phần tử cụ thể.
* Transitions và Animations: CSS3 cho phép tạo hiệu ứng chuyển động và hoạt hình mà không cần sử dụng JavaScript. Bạn có thể sử dụng transition và @keyframes để tạo các hiệu ứng động.
* Box Shadow và Border Radius: CSS3 cho phép thêm đổ bóng và bo góc cho các phần tử, giúp tạo ra giao diện trông hiện đại hơn.
* Flexbox: CSS3 giới thiệu mô hình Flexbox cho việc quản lý bố cục và căn chỉnh phần tử dễ dàng hơn.
* Grid Layout: CSS3 cung cấp khung lưới (grid layout) cho việc bố trí các phần tử trên trang web một cách mạnh mẽ và linh hoạt.
* Custom Fonts: CSS3 cho phép sử dụng phông chữ tùy chỉnh bằng cách sử dụng @font-face, giúp thiết kế trang web trở nên độc đáo hơn.
* Media Queries: Media queries trong CSS3 cho phép điều chỉnh giao diện dựa trên kích thước màn hình, giúp tạo ra trang web phản hồi và thân thiện với nhiều thiết bị.

Những tính năng mới trong HTML5 và CSS3 đã đánh dấu sự tiến bộ lớn trong phát triển web, cho phép phát triển trang web hiện đại và thú vị với nhiều tính năng và giao diện tốt hơn.

## **8.2 Hướng dẫn cách sử dụng các phần tử HTML5 như video, âm thanh, canvas, và đồ họa vector**

Sử dụng các phần tử HTML5 như <video>, <audio>, <canvas>, và đồ họa vector là một phần quan trọng của việc tạo trải nghiệm tương tác và thú vị trên trang web của bạn. Dưới đây là hướng dẫn cách sử dụng chúng:

**Sử dụng phần tử <video>:**

Thêm video vào trang HTML: Sử dụng phần tử <video> để nhúng video vào trang web. Đặt thuộc tính src là URL của video và thêm các thuộc tính khác như controls (để hiển thị điều khiển video) và autoplay (nếu bạn muốn video tự động phát).

Ví dụ : <video src="video.mp4" controls autoplay></video>

**Sử dụng phần tử <audio>:**

Sử dụng phần tử <audio> tương tự như <video>, nhưng cho âm thanh thay vì video.

Ví dụ: <audio src="audio.mp3" controls autoplay></audio>

**Sử dụng phần tử <canvas>:**

Phần tử <canvas> cho phép bạn vẽ đồ họa trực tiếp trên trang web bằng JavaScript. Dưới đây là một ví dụ cơ bản:

Tạo phần tử <canvas> trong HTML: <canvas id="myCanvas" width="400" height="200"></canvas>

**Sử dụng đồ họa vector**:

Đồ họa vector thường được tạo ra bằng các công cụ thiết kế vector như Adobe Illustrator hoặc Inkscape và sau đó nhúng vào trang web dưới dạng hình ảnh vector SVG. Để nhúng SVG vào trang web, bạn chỉ cần sử dụng thẻ <img> hoặc đặt SVG trực tiếp trong tài liệu HTML.

Ví dụ: <img src="my-vector-image.svg" alt="Vector Image">

<svg width="100" height="100">

<circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />

</svg>

## **8.3 Sử dụng CSS3 để thêm các hiệu ứng và chuyển động vào trang web, bao gồm cả CSS transitions và animations.**

Sử dụng CSS3 để thêm hiệu ứng và chuyển động vào trang web là một cách tuyệt vời để làm cho trang web của bạn trở nên hấp dẫn và tương tác. Dưới đây là cách sử dụng CSS3 để tạo các hiệu ứng và chuyển động, bao gồm CSS transitions và animations:

**CSS Transitions**:

CSS transitions cho phép bạn tạo hiệu ứng chuyển đổi từ trạng thái này sang trạng thái khác khi một phần tử thay đổi. Dưới đây là ví dụ về cách sử dụng CSS transitions:

1.Định rõ thuộc tính muốn chuyển đổi: Để bắt đầu, bạn cần xác định thuộc tính CSS mà bạn muốn áp dụng chuyển đổi. Ví dụ, bạn muốn chuyển đổi màu nền của một nút khi di chuột qua.

Ví dụ: .button {

background-color: #3498db;

transition: background-color 0.3s ease;

}

2.Áp dụng chuyển đổi khi trạng thái thay đổi: Sử dụng pseudoclass (:hover, :active, :focus,...) để xác định khi chuyển đổi sẽ xảy ra.

Ví dụ: .button:hover {

background-color: #e74c3c;

}

**CSS Animations**:

CSS animations cho phép bạn tạo các hiệu ứng chuyển động phức tạp hơn bằng cách định nghĩa các keyframes. Dưới đây là ví dụ về cách sử dụng CSS animations:

1.Định nghĩa keyframes: Đầu tiên, định nghĩa các keyframes cho chuyển động bằng cách sử dụng @keyframes. Ví dụ:

@keyframes fadeIn {

0% { opacity: 0; }

100% { opacity: 1; }

}

2.Áp dụng animation vào phần tử:

Sử dụng thuộc tính animation để áp dụng animation vào phần tử. Trong ví dụ này, chúng ta sẽ áp dụng animation fadeIn trong 2 giây với một lần lặp.

Ví dụ: .element {

animation: fadeIn 2s;

}

3.Chỉ định sự chuyển động:

Để xác định lặp đi lặp lại hoặc thời gian trễ, bạn có thể sử dụng các thuộc tính như animation-iteration-count, animation-delay, và animation-direction.

Ví dụ: .element {

animation: fadeIn 2s infinite alternate;

}

Sử dụng CSS3 transitions và animations có thể tạo ra trang web với hiệu ứng và chuyển động thú vị, tạo trải nghiệm người dùng tốt hơn và làm cho giao diện trang web trở nên hấp dẫn.

# 9.Thiết kế trang web thân thiện với SEO

## **9.1 Giới thiệu về SEO (Search Engine Optimization) và tại sao nó quan trọng đối** với trang web

Search Engine Optimization (SEO) là một loạt các thủ tục và chiến lược được áp dụng để cải thiện sự hiển thị và xếp hạng của trang web của bạn trên các trang kết quả tìm kiếm của các công cụ tìm kiếm như Google, Bing, Yahoo, và nhiều công cụ tìm kiếm khác. Mục tiêu chính của SEO là tối ưu hóa trang web của bạn để nó xuất hiện ở vị trí cao hơn trong kết quả tìm kiếm tự nhiên (organic search results) cho các từ khóa liên quan đến nội dung trang web của bạn.

Dưới đây là một số lý do tại sao SEO quan trọng đối với trang web:

* Tăng Lưu Lượng Truy Cập: SEO giúp trang web của bạn xuất hiện ở vị trí cao hơn trên kết quả tìm kiếm, điều này đồng nghĩa với việc nhiều người hơn sẽ thấy trang web của bạn và truy cập nó.
* Cải Thiện Tỷ lệ Chuyển Đổi: Khi bạn tối ưu hóa trang web cho từ khóa liên quan và cung cấp nội dung chất lượng, người dùng truy cập trang web của bạn thường có khả năng cao hơn để thực hiện các hành động mà bạn muốn họ thực hiện, chẳng hạn như mua hàng, đăng ký, hoặc liên hệ.
* Tăng Nhận Diện Thương Hiệu: Vị trí cao trong kết quả tìm kiếm thường được liên kết với uy tín và đáng tin cậy. Điều này giúp cải thiện nhận diện thương hiệu của bạn.
* Cạnh tranh Hiệu Quả: SEO giúp bạn cạnh tranh với các đối thủ trong lĩnh vực của bạn. Nếu trang web của bạn xuất hiện ở vị trí cao hơn trong kết quả tìm kiếm, bạn có lợi thế cạnh tranh lớn hơn.
* Tối ưu hóa Trải Nghiệm Người Dùng: SEO không chỉ liên quan đến từ khóa và liên kết, mà còn liên quan đến cải thiện trải nghiệm của người dùng trên trang web của bạn. Điều này có thể cải thiện tỷ lệ giữ chân và tương tác của người dùng.
* Cung Cấp Thông Tin Chính Xác: SEO giúp bạn cung cấp thông tin chính xác và cụ thể cho người tìm kiếm, giúp họ tìm thấy thông tin một cách dễ dàng và nhanh chóng.
* Cắt Giảm Chi Phí Quảng Cáo: So với quảng cáo trực tiếp, SEO có thể làm giảm chi phí tiếp thị, đặc biệt khi trang web của bạn đứng ở vị trí cao trên kết quả tìm kiếm.

SEO là một phần quan trọng trong việc xây dựng và duy trì hiện diện trực tuyến của bạn. Nó giúp trang web của bạn thấy được và tạo cơ hội tương tác với khách hàng tiềm năng.

## **9.2 Hướng dẫn cách tối ưu hóa tiêu đề trang, thẻ meta, và nội dung để cải thiện thứ hạng tìm kiếm.**

### 1. Tối ưu thẻ tiêu đề và thẻ meta

Thẻ tiêu đề và mô tả meta rất quan trọng. Có thể bạn nghĩ thẻ tiêu đề và mô tả meta bước đầu cung cấp cho người đọc ý tưởng hoặc tóm tắt chung nhất về nội dung nhưng đã khi nào bạn tự hỏi tại sao Google lại chú ý đến cả hai thẻ này?

Đó là bởi nếu như với người đọc, tiêu đề và mô tả meta cung cấp cho người đọc thông tin tóm tắt nhất về nội dung thì với Google, chúng cũng xác định mức độ liên quan giữa nội dung của bạn đối với tìm kiếm của người dùng.

Do đó, bạn nên tối ưu thẻ tiêu đề và thẻ mô tả meta của mình, bao gồm các từ khóa mục tiêu và đảm bảo rằng độ dài của chúng hoàn hảo cho các công cụ tìm kiếm. Nếu một trong hai yếu tố quá dài, nó có thể bị cắt trong các trang kết quả, dẫn tới cản trở người dùng có thể xem nội dung.

### 2. Tối ưu thẻ alt

Thẻ alt – thẻ hình ảnh sẽ cung cấp ngữ cảnh cho những người dùng không thể nhìn thấy hình ảnh trên trang web của bạn, đồng thời cũng cung cấp ngữ cảnh cho Google.

Vì Google không thể nhìn thấy hình ảnh, do đó, văn bản thay thế là cần thiết để giúp Google hiểu ngữ cảnh của hình ảnh và xác định xem chúng có liên quan đến nội dung hay không.

Việc tối ưu thẻ alt không chỉ quan trọng cho Google để hiểu hình ảnh mà việc cung cấp văn bản thay thế là một yếu tố quan trọng trong trải nghiệm người dùng. Thẻ alt sẽ cho người dùng biết hình ảnh của bạn là gì nếu hình ảnh không thể hiển thị chính xác, nhất là với những người đang sử dụng trình đọc màn hình.

### 3. Tối ưu các từ khóa phù hợp

Nếu bạn đang tự hỏi làm thế nào để cải thiện SEO on-page thì việc tối ưu hóa từ khóa chắc chắn là không thể bỏ qua.

Từ khóa là một trong những tín hiệu lớn nhất cho Google hiểu rằng nội dung của bạn có liên quan, điều đó cũng đồng nghĩa, tối ưu hóa từ khóa cũng giúp thương hiệu cải thiện thứ hạng tìm kiếm. Trước khi viết nội dung, hãy đảm bảo nghiên cứu từ khóa đủ sâu và đủ rộng để quyết định chọn từ khóa có giá trị nhất.

Nếu đã viết nội dung và đang tìm cách cải thiện nỗ lực SEO trên trang bằng cách tối ưu từ khóa, bạn chỉ cần tìm cách kết hợp nhiều từ khóa có giá trị cao vào nội dung hiện có, có thể bằng cách:

* Thêm nội dung về một chủ đề có liên quan để nhắm mục tiêu từ khóa chất lượng cao
* Thay các từ khóa có giá trị thấp, từ khoá ngắn bằng những từ khóa có giá trị hơn
* Thêm từ khóa giá trị vào nội dung hiện có

### 4. Tối ưu nội dung

Cách duy nhất để hiển thị trong kết quả tìm kiếm là tạo nội dung. Điều quan trọng là phải lên lịch nội dung để theo đúng định hướng tạo cũng như nhất quán với việc triển khai nội dung.

Dưới đây là lý do tại sao nội dung rất quan trọng để tăng kết quả SEO:

* Google rất thích các trang web liên tục xuất bản nội dung mới
* Càng tạo nhiều nội dung, bạn càng có thể nhắm mục tiêu nhiều từ khóa hơn
* Càng cung cấp nhiều nội dung cho người dùng, thì họ càng có nhiều cơ hội tìm thấy bạn
* Càng tạo ra nhiều nội dung, bạn càng có thể khai thác lợi thế của nhiều nền tảng khác nhau

Nói về sự đa dạng của nội dung, bạn không chỉ cần gắn chặt với một định dạng nội dung. Trên thực tế, Google thích xem nhiều loại nội dung hấp dẫn người dùng theo những cách khác nhau. Chẳng hạn thay vì chỉ bằng văn bản, có thể thử:

* Video
* Infographic
* Podcast
* Hướng dẫn

Những loại nội dung khác nhau này có thể giúp bạn thu hút khán giả của mình theo những cách khác nhau và cũng cho phép dễ dàng chia sẻ nội dung trên các nền tảng khác nhau.

## **9.3 Sử dụng công cụ tối ưu hóa SEO để theo dõi hiệu suất và tối ưu hóa trang web.**

1. **Công cụ SEO**

Công cụ SEO là những công cụ kỹ thuật hỗ trợ cho người làm SEO kiểm tra, đo lường, và cải thiện kết quả của việc tối ưu hóa website. Cụm từ này tiếng Anh gọi là SEO Tool, thường là các phần mềm, chạy độc lập hoặc tích hợp trên các website của nhà cung cấp.

Việc sử dụng cũng khá đơn giản. Cách thường thấy là chỉ cần tải xuống và cài đặt trên trình duyệt là có thể sử dụng luôn, chẳng hạn: SeoQuake, Mozbar, Alexa. Với một số khác như Google Analytics, Yandex Metrica thì bạn cần đăng ký tài khoản, sau đó tạo và gắn một số đoạn mã (code) vào trang web là xong.

Câu hỏi mà nhiều người quan tâm: sử dụng những thứ đó có phải trả tiền không?

Nhiều trong số công cụ SEO hiện tại đang miễn phí. Bạn có thể sử dụng chúng mà không phải trả tiền. Mặc dù miễn phí, chúng vẫn cực kỳ hữu ích, tiêu biểu nhất là Google Analytics, Mozbar...

Bên cạnh đó, sẽ có những công cụ chỉ miễn phí cho gói cơ bản, và tính tiền cho gói dịch vụ cao cấp hơn. Ngoài ra, bạn sẽ thấy có những công cụ phải trả phí, thường đi kèm với 1 khoảng thời gian sử dụng thử.

Tựu chung lại, căn cứ vào chi phí, các công cụ này có thể xếp vào 3 nhóm:

* Miễn phí (Free): hoàn toàn miễn phí
* Miễn phí + trả phí (Freemium): miễn phí gói cơ bản, tính phí với gói cao cấp
* Mất phí (Premium): tính phí

2. Những công cụ SEO phổ biến hiện nay

### 2.1. Ahrefs: Công cụ SEO keyword

Ahrefs là một trong các công cụ SEO trực tuyến được khuyên sử dụng nhất. Nó chỉ đứng thứ 2 sau Google khi trở thành trình lấy dữ liệu trang Web lớn nhất.

Các chuyên gia SEO không thể có đủ tính năng Kiểm toán Web của Ahref vì đây là một công cụ phân tích SEO tốt nhất. Công cụ nêu bật những phần nào trong trang Web của bạn yêu cầu cải tiến để giúp đảm bảo xếp hạng tốt nhất của bạn.

### 2.2. BuzzStream

Có nhiều chức năng khó tin mà các công cụ SEO này cung cấp, bao gồm:

Nghiên cứu, phân tích một cách nhanh chóng và hiệu quả. bạn sẽ tăng tốc độ nghiên cứu, theo dõi các contact, và tự động thu thập thông tin cần thiết.

Gửi thông điệp tiếp xúc. Một phần đông của việc xây dựng sự liên kết là cài đặt mối quan hệ. bạn có thể sử dụng BuzzStream để gửi các tin nhắn xây dựng mối tương quan cá nhân hóa, và tạo ra kết quả.

Quản lý nhiều dự án xây dựng liên kết của bạn. Các dự án xây dựng liên kết tốt nhất được quản lý và tổ chức hiệu quả từ khi bắt đầu đến cuối. BuzzStream có công cụ để theo dõi tất cả các phương diện của chiến dịch SEO đạt kết quả tốt

BuzzStream được sử dụng bởi các cá nhân, các công ty nhỏ, các doanh nghiệp lớn và mọi người ở giữa. công ty cho biết rõ ràng rằng các nhà xây dựng liên kết tốt nhất thế giới dựa vào các công cụ SEO của nó. BuzzStream muốn bạn làm như vậy.

### 2.3. Google PageSpeed Insights

Kiểm tra về tốc độ và năng lực sử dụng các trang Web của bạn trên nhiều thiết bị. Chỉ cần nhập URL và công cụ này sẽ kiểm tra thời gian tải và hiệu năng cho máy tính và cả điện thoại di động.

Trong đó nó còn có cơ hội nắm rõ ràng ưu nhược điểm của Web. Kết quả trên thiết bị di động cũng gắn với trải nghiệm người sử dụng, theo các tiêu chí như mục đích máy và kích cỡ phông chữ.

### 2.4. Google Webmaster Tools

Chưa bàn đến việc dùng công cụ ở Đông, Tây, hay bất kì chỗ quái nào, Google Webmaster Tools luôn là công cụ bạn phải cài đặt ngay khi thành lập 1 Web (xem chỉ dẫn thiết lập tại đây).

Với Google Webmaster Tools, điều mà hầu hết mọi người đều biết (hoặc chỉ biết), đó là submit sitemap lên trên đó để Google đơn giản index Web của chúng ta hơn. Tuy nhiên, Google Webmaster Tools còn cực nhiều điều mà mọi người cần biết nếu như muốn SEO tốt.

### 2.5. Google Analytics

Google Analytics – một trong số các công cu SEO phổ biến nhất hiện nay: Thống kê các chỉ số về lượt click, đối tượng mục tiêu người dùng, thời gian hiện tại có bao nhiêu khách truy cập Website,.. bài viết nào đang được ưa thích nhất, thời gian ở lại trang của người dung.

Các chỉ số điều được liệt kệ một các chi tiết theo bảng và cột: như số lượng, phần%… Rất trực quan và dễ nhìn.

### 2.6. Copyscape

Kiểm duyệt nội dung trùng lặp. Nhập URL của một blogpost hoặc Website, và Copyscape có thể cho bạn biết ở những đâu xuất hiện cùng một nội dung nào đấy.

Kết quả mà bạn nhận được có thể phải được theo dõi cẩn thận để cam kết việc SEO của bạn thống nhất.

### 2.7. Công cụ Yoast SEO

Sau khi nghiên cứu, phân tích từ khóa xong suôi. Bước tiếp sẽ là lên thông tin và viết bài với những bộ từ khóa đó.

Chúng ta hay gọi là viết bài chuẩn SEO.

đơn giản là tập hợp một danh sách kỹ thuật bạn có thể ứng dụng trong bài đăng như:

Từ khóa phải có trong URL, title, meta description, heading 1, H2,..

Mật độ keyword sẽ đâu đấy phải là 1-5%

Trong bài viết sẽ phải có đường link nội bộ

Ảnh có atl chứa từ khóa

# **10.Quản lý phiên bản và Git**

## **10.1 Giới thiệu về hệ thống quản lý mã nguồn và Git.**

Hệ thống quản lý mã nguồn và Git là một trong những công cụ quan trọng trong phát triển phần mềm, cho phép các nhà phát triển làm việc cùng nhau để theo dõi và quản lý mã nguồn của dự án. Dưới đây là một sự giới thiệu về hệ thống quản lý mã nguồn và Git:

* Hệ thống quản lý mã nguồn (Source Code Management - SCM): Hệ thống quản lý mã nguồn là một phần quan trọng của quá trình phát triển phần mềm, cho phép lưu trữ, theo dõi và quản lý mã nguồn của một dự án. SCM giúp quản lý sự thay đổi trong mã nguồn, thuận tiện cho việc làm việc nhóm và duyệt mã nguồn. Các SCM phổ biến khác bao gồm Subversion (SVN) và Mercurial.
* Git: Git là một hệ thống quản lý mã nguồn phân tán (Distributed Version Control System - DVCS) phổ biến nhất hiện nay. Được phát triển bởi Linus Torvalds vào năm 2005, Git giúp quản lý và theo dõi sự thay đổi trong mã nguồn một cách hiệu quả và linh hoạt. Dưới đây là một số điểm quan trọng về Git:
* Hệ thống phân tán: Mỗi nhà phát triển có bản sao đầy đủ của toàn bộ lịch sử mã nguồn trên máy của họ. Điều này cho phép làm việc offline và giúp giảm rủi ro mất dữ liệu.
* Nhánh (Branch): Git cho phép tạo và quản lý nhiều nhánh khác nhau của dự án. Nhánh giúp phát triển và thử nghiệm tính năng mới mà không ảnh hưởng đến phiên bản chính.
* Ghi lại lịch sử: Git lưu trữ toàn bộ lịch sử thay đổi, cho phép xem lại, so sánh và khôi phục phiên bản cũ.
* Hợp nhất (Merge) và Cảnh báo xung đột (Conflict Resolution): Git cung cấp công cụ để hợp nhất các nhánh và xử lý xung đột khi có nhiều phiên bản thay đổi đối chọi.
* Nền tảng phong cách làm việc phi tuyến tính (Non-linear Workflow): Git cho phép làm việc theo nhiều mô hình làm việc khác nhau, chẳng hạn như phát triển song song, quản lý phiên bản phát hành và nhiều khả năng khác.
* GitHub và GitLab: GitHub và GitLab là các dịch vụ hoặc nền tảng dựa trên Git, cho phép lưu trữ mã nguồn trực tuyến, làm việc nhóm, theo dõi vấn đề và tích hợp liên kết với dự án phát triển.

Git đã trở thành một công cụ quản lý mã nguồn quan trọng cho cộng đồng phát triển phần mềm và được sử dụng rộng rãi trong các dự án từ nhỏ đến lớn.

## **10.2 Hướng dẫn tạo kho lưu trữ (repository) Git và làm việc với các lệnh cơ bản như git clone, git pull, git push, và git commit.**

1. Tạo kho lưu trữ Git (Repository): Để tạo một kho lưu trữ Git mới, bạn có thể sử dụng Git cục bộ hoặc sử dụng dịch vụ như GitHub hoặc GitLab để tạo kho lưu trữ trực tuyến. Dưới đây là cách tạo một kho lưu trữ Git cục bộ:

* Mở cửa sổ dòng lệnh (Command Prompt hoặc Terminal).
* Điều hướng đến thư mục nơi bạn muốn tạo kho lưu trữ.
* Sử dụng lệnh sau để tạo kho lưu trữ: git init.
* Kho lưu trữ sẽ được tạo trong thư mục hiện tại.

2. Clone kho lưu trữ từ xa: Để sao chép một kho lưu trữ từ xa vào máy tính của bạn, bạn sử dụng lệnh git clone. Thay đổi <repository\_url> thành URL của kho lưu trữ bạn muốn sao chép: git clone <repository\_url>.

3. Commit thay đổi: Sau khi bạn đã thay đổi mã nguồn, bạn cần commit để lưu trữ các thay đổi đó trong kho lưu trữ. Sử dụng các lệnh sau:

* Để thêm tất cả các tệp đã thay đổi vào phạm vi commit: git add .
* Để commit các thay đổi đã thêm vào phạm vi commit: git commit -m "Nội dung commit"

4. Pull từ kho lưu trữ từ xa: Khi bạn làm việc cùng với nhóm và muốn đồng bộ hóa mã nguồn từ kho lưu trữ từ xa về máy tính của bạn, sử dụng lệnh git pull: git pull.

5. Push lên kho lưu trữ từ xa: Để đưa các thay đổi bạn đã commit lên kho lưu trữ từ xa (ví dụ: GitHub hoặc GitLab), sử dụng lệnh: git push.

Lưu ý rằng trước khi push, bạn cần phải commit tất cả các thay đổi của bạn. Hãy cẩn thận và xem xét thay đổi của mình trước khi commit và push để đảm bảo rằng bạn không đẩy lên các thay đổi không mong muốn.

## **10.3 Thảo luận về việc làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án.**

Làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án là một phần quan trọng của quá trình phát triển phần mềm. Dưới đây, chúng ta sẽ thảo luận về các khía cạnh quan trọng trong việc làm việc cùng nhóm và quản lý phiên bản dự án bằng Git:

* Hợp nhất (Merge) và Nhánh (Branch):
* Nhánh (Branch): Một số lượng lớn các tính năng và sửa lỗi có thể được phát triển đồng thời. Mỗi tính năng hoặc sửa lỗi có thể được phát triển trên một nhánh riêng. Các nhánh này cho phép ngăn chặn xung đột giữa các thay đổi và cho phép làm việc độc lập trước khi hợp nhất vào nhánh chính (thường là master hoặc main).
* Hợp nhất (Merge): Khi một tính năng hoặc sửa lỗi đã hoàn thành và kiểm tra, nó có thể được hợp nhất vào nhánh chính thông qua quá trình merge hoặc rebase. Sử dụng cẩn thận để đảm bảo rằng không có xung đột và tính năng đã hoạt động một cách đúng đắn.
* Quản lý Xung Đột (Conflict Resolution): Trong trường hợp xung đột xảy ra khi hợp nhất các nhánh, Git sẽ yêu cầu bạn giải quyết xung đột. Điều này xảy ra khi hai người hoặc nhiều người cùng sửa chữa cùng một dòng mã hoặc tệp. Quá trình giải quyết xung đột yêu cầu sự hợp tác và kiên nhẫn trong việc tự động hoặc thủ công giải quyết xung đột.
* Dự án Phát Hành (Release Management): Git cho phép quản lý các phiên bản dự án và việc phát hành phần mềm dễ dàng hơn. Bằng cách tạo nhãn (tags) trên các phiên bản ổn định, bạn có thể theo dõi lịch sử phiên bản dự án và quản lý việc phát hành phiên bản cho người dùng cuối.
* Công cụ Quản lý Dự án (Project Management Tools): Khi làm việc trong môi trường Git, bạn có thể kết hợp Git với các công cụ quản lý dự án như GitHub Projects, GitLab Boards, hay các ứng dụng quản lý công việc như JIRA. Điều này giúp quản lý các công việc, tiến độ dự án và sự gắn kết giữa các nhiệm vụ và thay đổi mã nguồn.
* Quản lý Lịch sử và Theo Dõi Thay Đổi (History and Change Tracking): Một trong những lợi ích lớn của Git là khả năng theo dõi lịch sử thay đổi. Mỗi lần commit được tạo, nó sẽ được lưu trữ trong lịch sử mã nguồn. Điều này giúp bạn hiểu được khi nào, ai và tại sao thực hiện một thay đổi cụ thể.
* Bảo Mật và Quản lý Quyền Truy Cập (Security and Access Control): Git cung cấp các công cụ để quản lý quyền truy cập và bảo mật. Bạn có thể xác định ai có quyền commit, push, hay pull từ kho lưu trữ. Các dịch vụ như GitHub và GitLab cung cấp quản lý quyền truy cập cấp cao hơn.

Tóm lại, làm việc cùng nhóm trong môi trường Git giúp quản lý và phát triển dự án một cách hiệu quả hơn. Quản lý phiên bản, hợp nhất, giải quyết xung đột, quản lý lịch sử, và sử dụng các công cụ quản lý dự án sẽ giúp đảm bảo dự án phát triển suôn sẻ và theo kế hoạch.

# TÀI LIỆU THAM KHẢO

- Link tài liệu:

<https://timoday.edu.vn/bai-3-css-selectors/>

https://thachpham.com/web-development/html-css/vung-chon-css-co-ban.html

- Link tài liệu: [Tổng hợp các thuộc tính của CSS | Tìm ở đây (timoday.edu.vn)](https://timoday.edu.vn/tong-hop-cac-thuoc-tinh-cua-css/#text)

**-** Link tài liệu: [Javascript là gì? Cú pháp Javascript cơ bản - VNTALKING](https://vntalking.com/cu-phap-javascript-co-ban-nhung-buoc-di-dau-tien.html#javascript-la-gi)

Tài liệu tham khảo: https://www.w3schools.com/

<https://www.codecademy.com/catalog>

Tài liệu tham khảo: A Complete Guide to Flexbox (https://csstricks.com/snippets/css/a-guide-to-flexbox/), A Complete Guide to Grid

(<https://css-tricks.com/snippets/css/complete-guide-grid/>)

Tài liệu tham khảo: http://html5doctor.com/

https://css-tricks.com/

https://www.pluralsight.com/courses/html5-css3-

fundamentals

Tài liệu tham khảo: https://moz.com/beginners-guide-to-seo

https://yoast.com/seo-blog/

<https://yoast.com/seo-blog/>

Tài liệu tham khảo: https://git-scm.com/book/en/v2

<https://www.atlassian.com/git>