**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

**TÊN ĐỀ TÀI: THIẾT KẾ WED BÁN HÀNG ĐIỆN THOẠI DI ĐỘNG**

**Giảng viên hướng dẫn: ThS. Lê Văn Phong**

**Sinh viên thực hiện:**  **Nguyễn Chí Vĩ**

**Hà Nội, 2024**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

**TÊN ĐỀ TÀI: THIẾT KẾ WED BÁN HÀNG ĐIỆN THOẠI DI ĐỘNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020741 | Nguyễn Chí Vĩ | 24/12/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

"Chào mừng đến với trang web của chúng tôi - nơi sáng tạo và chuyên nghiệp hóa ý tưởng của bạn! Chúng tôi là đội ngũ thiết kế web đam mê và sáng tạo, cam kết mang đến cho bạn những giải pháp thiết kế web độc đáo và hiệu quả. Với sự kết hợp giữa kiến thức chuyên môn và tinh thần sáng tạo, chúng tôi không chỉ tạo ra những trang web đẹp mắt mà còn đảm bảo tính tương tác, trải nghiệm người dùng tốt nhất. Chúng tôi tin rằng mỗi dự án là một cơ hội để thể hiện tài năng và đem lại giá trị thực sự cho khách hàng. Hãy để chúng tôi biến ý tưởng của bạn thành hiện thực trên không gian số! Dù bạn cần thiết kế web cá nhân, doanh nghiệp hay thương mại điện tử, chúng tôi luôn sẵn lòng hỗ trợ bạn với sự chuyên nghiệp và tận tâm. Hãy cùng chúng tôi khám phá và tạo nên những trải nghiệm web độc đáo!

MỤC LỤC

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆTHỐNG PHẦN MỀM** 7](#_Toc161179252)

[**1.1. Các khái niệm cơ bản** 7](#_Toc161179253)

[**1.2. Phân biệt Web tĩnh, Web động** 8](#_Toc161179254)

[ **Web Tĩnh (Static Web):** 8](#_Toc161179255)

[ **Web Động (Dynamic Web):** 8](#_Toc161179256)

[**1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage,Web Site, WebServer, URL-Uniform Resource Location, Browser, …)** 9](#_Toc161179257)

[**1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …** 10](#_Toc161179258)

[**CHƯƠNG II: HTML &amp; HTML5** 14](#_Toc161179259)

[**2.1. Tổng quan về HTML** 14](#_Toc161179260)

[**2.2. Cấu trúc tổng quát trang HTML** 15](#_Toc161179261)

[**2.3. Các thẻ HTML thông dụng** 16](#_Toc161179262)

[**2.4. Các thẻ tạo biểu mẫu (form)** 18](#_Toc161179263)

[**2.5. Một số thẻ HTML đặc biệt** 19](#_Toc161179264)

[**2.6. HTML5** 21](#_Toc161179265)

[HTML5 là gì? 21](#_Toc161179266)

[Cú pháp HTML5 21](#_Toc161179267)

[Thẻ ngữ nghĩa 21](#_Toc161179268)

[Webform 22](#_Toc161179269)

[**CHƯƠNG III: CSS và CSS3** 23](#_Toc161179270)

[**3.1. CSS là gì?** 23](#_Toc161179271)

[**3.2. Cú pháp CSS** 23](#_Toc161179272)

[**3.3. Áp dụng CSS vào trang HTML** 24](#_Toc161179273)

[**3.4. Selectors** 26](#_Toc161179274)

[**3.4.1. Universal selector** Universal Selector là một loại selector trong CSS cho phép bạn chọn tất cả các phần tử trên trang web. Nó được biểu diễn bằng dấu \*. Khi bạn áp dụng CSS cho Universal Selector, quy tắc sẽ được áp dụng cho mọi phần tử trên trang, không phụ thuộc vào loại, class, hoặc id. 28](#_Toc161179275)

[**3.4.2. Type selector** 29](#_Toc161179276)

[**3.4.3. Identity selector** 29](#_Toc161179277)

[**3.4.4. Class selector** 29](#_Toc161179278)

[**3.4.5. Descendant selector** 30](#_Toc161179279)

[**3.4.6. Child selector** 30](#_Toc161179280)

[**3.4.7. Adjacent sibling selector** Adjacent Sibling Selector trong CSS được sử dụng để chọn phần tử anh em (sibling) trực tiếp và kế tiếp của một phần tử cụ thể. Cú pháp của Adjacent Sibling Selector là sử dụng dấu + giữa các tên phần tử. Ví dụ: 30](#_Toc161179281)

[**3.4.8. Attribute selector** 31](#_Toc161179282)

[**3.4.9. Pseudo class selector** Pseudo-class Selector (Selector Pseudo-class) trong CSS được sử dụng để chọn các phần tử trong một trạng thái hoặc vị trí cụ thể. Cú pháp của Pseudo-class Selector là sử dụng dấu hai chấm : sau tên phần tử. Dưới đây là một số ví dụ: 31](#_Toc161179283)

[**3.4.10. Group selector** Group Selector trong CSS cho phép bạn kết hợp nhiều loại selector để áp dụng cùng một tập hợp quy tắc CSS cho nhiều phần tử. Cú pháp của Group Selector là sử dụng dấu phẩy , để ngăn cách giữa các selector. Dưới đây là một ví dụ: 32](#_Toc161179284)

[**3.5. Đơn vị đo lường CSS** 32](#_Toc161179285)

[**3.6. Kế thừa thuộc tính** 33](#_Toc161179286)

[**3.7. Các nhóm thuộc tính trong CSS** 34](#_Toc161179287)

[- Thuộc tính định dạng font chữ, văn bản (font, text) Thuộc tính định dạng font chữ và văn bản trong CSS giúp bạn kiểm soát các khía cạnh như loại font, kích thước, màu sắc, độ dày, và các thuộc tính liên quan đến văn bản. Dưới đây là một số thuộc tính quan trọng: 34](#_Toc161179288)

[- Thuộc tính định dạng nền (background) 35](#_Toc161179289)

[- Mô hình hộp (box model) 37](#_Toc161179290)

[- Các thuộc tính định margin, padding, border 37](#_Toc161179291)

[- Thuộc tính định dạng cách hiển thị (display) 39](#_Toc161179292)

[- Thuộc tính xác định vị trí (position) 40](#_Toc161179293)

[**3.8. Float &amp; Clear** 41](#_Toc161179294)

[**3.9. Flex** 42](#_Toc161179295)

[**3.10. Grid** 42](#_Toc161179296)

[**3.11. CSS3** - CSS3 là gì? CSS3 là phiên bản thứ ba của Cascading Style Sheets (CSS), một ngôn ngữ mô tả kiểu dáng của trang web. CSS3 đưa ra nhiều cải tiến so với phiên bản trước đó, CSS2, bằng cách thêm vào nhiều tính năng mới và mở rộng khả năng thiết kế của lập trình viên và nhà phát triển web. 43](#_Toc161179297)

[- CSS3 Selectors 44](#_Toc161179298)

[- CSS3 Rounded Corners 45](#_Toc161179299)

[- CSS3 Border Images 45](#_Toc161179300)

[- CSS3 Backgrounds CSS3 cung cấp nhiều tính năng mới cho việc định dạng nền của phần tử HTML. Dưới đây là một số tính năng quan trọng của CSS3 Backgrounds: 46](#_Toc161179301)

[- CSS Gradients 47](#_Toc161179302)

[- CSS3 Shadows 49](#_Toc161179303)

[- CSS3 Fonts 50](#_Toc161179304)

[- CSS3 2D Transforms 52](#_Toc161179305)

[- CSS3 3D Transforms 54](#_Toc161179306)

[- CSS3 Transitions 55](#_Toc161179307)

[- CSS3 Multiple Columns 56](#_Toc161179308)

[- CSS3 Animations 57](#_Toc161179309)

[**3.12. SCSS** 57](#_Toc161179310)

[**3.13. SASS** 58](#_Toc161179311)

[**KẾT LUẬN CHƯƠNG 3** 60](#_Toc161179312)

[**CHƯƠNG 4: THIẾT KẾ WEBISTE THEO ĐỀ TÀI BẠN CHỌN** 62](#_Toc161179313)

[**4.1. Ý tưởng của Website** 62](#_Toc161179314)

[**4.2. Xây dựng bố cục của trang Web** 62](#_Toc161179315)

[**4.3. Thiết kế trang Web bằng HTML và CSS** 63](#_Toc161179316)

[**4.4. Kết quả đạt được** 65](#_Toc161179317)

[**KẾT LUẬN** 66](#_Toc161179318)

**MỤC LỤC HÌNH ẢNH**

**(Nếu có)**

**MỤC LỤC BẢNG**

**(Nếu có)**

**BẢNG CÁC TỪ VIẾT TẮT**

**(Nếu có)**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| 1 | HTML | Hyper Text Markup Language |
| 2 |  |  |

# **CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆTHỐNG PHẦN MỀM**

## **1.1. Các khái niệm cơ bản**

* **Giao diện người dùng (User Interface - UI):** Giao diện người dùng là phần của trang web mà người dùng tương tác với. Nó bao gồm các thành phần như nút, hộp văn bản, menu, hình ảnh và các phần tử trực quan khác.
* **Trải nghiệm người dùng (User Experience - UX):** Trải nghiệm người dùng đề cập đến cảm giác tổng thể của người dùng khi sử dụng trang web, bao gồm sự dễ sử dụng, hiệu suất, tính hấp dẫn và thỏa mãn nhu cầu của họ.
* **Responsive Design:** Thiết kế đáp ứng (responsive design) là việc tạo ra trang web sao cho nó tự động thích ứng với các kích thước màn hình khác nhau của các thiết bị, từ máy tính để bàn đến điện thoại di động và máy tính bảng.
* **Tương tác (Interactivity):** Tương tác là tính năng của trang web cho phép người dùng tương tác với nó thông qua các phản hồi và hành động, bao gồm các biểu mẫu, hộp thoại, và các yếu tố tương tác khác.
* **Thiết kế đồ họa (Graphic Design):** Thiết kế đồ họa là việc tạo ra các yếu tố trực quan trên trang web như hình ảnh, biểu đồ, biểu tượng và màu sắc để tạo ra một trải nghiệm thú vị và hấp dẫn cho người dùng.
* **Phân tích và Đo lường (Analytics):** Phân tích và đo lường là quá trình thu thập dữ liệu về hoạt động của người dùng trên trang web để hiểu hành vi của họ và cải thiện trải nghiệm người dùng và hiệu suất trang web.
* **Tối ưu hóa công cụ tìm kiếm (SEO - Search Engine Optimization):** SEO là các kỹ thuật được áp dụng để tăng cường khả năng tìm thấy của trang web trên các công cụ tìm kiếm như Google, bằng cách cải thiện vị trí của trang web trong kết quả tìm kiếm.
* **Tuân thủ tiêu chuẩn web (Web Standards Compliance):** Tuân thủ tiêu chuẩn web là việc thiết kế và phát triển trang web theo các nguyên tắc và hướng dẫn được đề xuất bởi các tổ chức như W3C (World Wide Web Consortium.

## **1.2. Phân biệt Web tĩnh, Web động**



* **Web Tĩnh (Static Web):**
  + Trang web tĩnh là các trang web mà nội dung được xây dựng và lưu trữ trước, và không thay đổi trong quá trình người dùng truy cập.
  + Các trang web tĩnh thường được viết bằng ngôn ngữ đánh dấu như HTML và CSS.
  + Thường không có khả năng tương tác hoặc thay đổi nội dung dựa trên hành vi của người dùng.
  + Ví dụ: trang web thông tin cơ bản, trang web cá nhân, trang web giới thiệu sản phẩm.



* **Web Động (Dynamic Web):**
  + Trang web động là các trang web mà nội dung được tạo ra dưới thời gian thực hoặc trong quá trình người dùng truy cập.
  + Các trang web động thường sử dụng các ngôn ngữ lập trình phía máy chủ như PHP, Python, Ruby, hoặc JavaScript để tạo nội dung theo yêu cầu.
  + Có khả năng tương tác với người dùng và thay đổi nội dung dựa trên hành vi của họ hoặc dữ liệu được cung cấp.
  + Ví dụ: các trang web thương mại điện tử với chức năng đăng nhập, trang web dự báo thời tiết, trang web xã hội.

Tóm lại, điểm khác biệt chính giữa web tĩnh và web động là khả năng tạo ra và thay đổi nội dung của trang web. Trong khi web tĩnh có nội dung cố định và không thay đổi, web động có khả năng tạo ra và thay đổi nội dung dựa trên yêu cầu và hành vi của người dùng.



## **1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage,Web Site, WebServer, URL-Uniform Resource Location, Browser, …)**



* **Hosting:** Hosting là dịch vụ cung cấp không gian lưu trữ trên máy chủ web để lưu trữ các tập tin, dữ liệu và ứng dụng của một trang web. Các nhà cung cấp dịch vụ hosting cho phép trang web của bạn truy cập trên internet thông qua một địa chỉ IP.
* **Tên miền (Domain Name):** Tên miền là địa chỉ dễ nhớ dùng để xác định một trang web trên internet. Ví dụ: google.com, facebook.com. Tên miền được liên kết với một địa chỉ IP để máy tính có thể tìm thấy trang web tương ứng.
* **Web Server Name:** Đây là tên của máy chủ web, nơi trang web được lưu trữ và phục vụ. Ví dụ: Apache, Nginx là các phần mềm máy chủ web phổ biến.
* **HomePage:** Trang chủ (HomePage) là trang web mặc định mà người dùng thường xem khi truy cập vào một trang web. Nó thường chứa thông tin quan trọng hoặc liên kết đến các phần khác của trang web.
* **Web Site:** Web Site là một tập hợp các trang web liên kết với nhau. Nó bao gồm nội dung, hình ảnh, video và các tài nguyên khác được hiển thị cho người dùng qua internet.
* **Web Server:** Máy chủ web (Web Server) là một máy tính chạy các phần mềm máy chủ web để phục vụ các trang web cho người dùng qua internet. Nó có nhiệm vụ phản hồi yêu cầu từ trình duyệt web và gửi lại các trang web tương ứng.
* **URL (Uniform Resource Locator):** URL là địa chỉ của một trang web trên internet. Nó bao gồm các yếu tố như giao thức (http, https), tên miền, đường dẫn và các tham số tùy chọn.
* **Trình duyệt (Browser):** Trình duyệt là phần mềm được sử dụng để duyệt web và hiển thị các trang web. Ví dụ: Google Chrome, Mozilla Firefox, Microsoft Edge là các trình duyệt phổ biến.





## **1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …**

* **Visual Studio Code (VS Code):** Visual Studio Code là một trình soạn thảo mã nguồn mở, miễn phí và phổ biến được phát triển bởi Microsoft. Nó cung cấp các tính năng hữu ích cho lập trình web như syntax highlighting, auto-completion, debugging, và tích hợp Git.



* **Adobe Dreamweaver:** Adobe Dreamweaver là một ứng dụng phần mềm được thiết kế để giúp người dùng tạo và quản lý các trang web một cách dễ dàng. Nó cung cấp giao diện trực quan và các công cụ kéo và thả cho việc thiết kế trang web, cũng như hỗ trợ mã nguồn cho những người muốn làm việc trực tiếp với mã HTML, CSS, và JavaScript.



* **Sublime Text:** Sublime Text là một trình soạn thảo mã nguồn văn bản nhanh và mạnh mẽ được yêu thích trong cộng đồng lập trình viên. Nó cung cấp các tính năng linh hoạt như multiple selections, command palette, và cộng đồng plugins phong phú.



* **Atom:** Atom là một trình soạn thảo mã nguồn mở được phát triển bởi GitHub. Nó có các tính năng giống như Sublime Text và được tích hợp sâu với các công cụ của GitHub như Git và GitHub.
* 
* **Brackets:** Brackets là một trình soạn thảo mã nguồn miễn phí được phát triển bởi Adobe Systems. Nó được tối ưu hóa cho việc lập trình web, với các tính năng như Live Preview để xem trước trực tiếp trên trình duyệt.



* **Bootstrap Studio:** Bootstrap Studio là một công cụ mạnh mẽ cho việc thiết kế trang web với Bootstrap framework. Nó cung cấp giao diện kéo và thả để dễ dàng tạo các giao diện trực quan và responsive.



* **Adobe XD:** Adobe XD là một công cụ thiết kế trải nghiệm người dùng (UX/UI) được sử dụng để thiết kế các giao diện người dùng cho web và ứng dụng di động. Nó cung cấp các tính năng như prototype, design system, và collaboration.



**kết luận:**

**Thiết kế Web:**

* Thiết kế web là quá trình tạo ra giao diện người dùng và trải nghiệm người dùng trên các trình duyệt web.
* Bao gồm các khía cạnh như giao diện người dùng (UI), trải nghiệm người dùng (UX), responsive design, tương tác, đồ họa, và tối ưu hóa cho công cụ tìm kiếm (SEO).
* Các công cụ và ngôn ngữ phổ biến trong thiết kế web bao gồm HTML, CSS, JavaScript, các trình duyệt web, và các công cụ thiết kế như Visual Studio Code, Adobe Dreamweaver, và Adobe XD.

**Triển khai Hệ thống Phần mềm:**

* Triển khai hệ thống phần mềm là quá trình triển khai các ứng dụng và hệ thống phần mềm vào môi trường sản xuất và vận hành.
* Bao gồm các bước như lựa chọn môi trường hosting, cấu hình hệ thống, triển khai mã nguồn, kiểm thử, và quản lý vận hành.
* Các công cụ và kỹ thuật trong triển khai hệ thống phần mềm bao gồm các dịch vụ hosting, máy chủ web, công cụ kiểm thử, quản lý dự án, và công cụ quản lý mã nguồn như Git.

Kết luận, việc hiểu và áp dụng các nguyên lý và công cụ trong thiết kế web và triển khai hệ thống phần mềm là rất quan trọng để tạo ra các ứng dụng web hiệu quả và có trải nghiệm người dùng tốt. Sự kết hợp giữa các yếu tố thiết kế web đẹp mắt và sáng tạo cùng với quá trình triển khai hệ thống phần mềm chặt chẽ và hiệu quả sẽ tạo nên sản phẩm cuối cùng ấn tượng và thành công trên thị trường.

# **CHƯƠNG II: HTML &amp; HTML5**

**(Hyper Text Markup Language)**

## **2.1. Tổng quan về HTML**



* **Cấu trúc cơ bản của HTML:**
  + HTML sử dụng các thẻ (tags) để đánh dấu các phần tử trong trang web.
  + Mỗi thẻ HTML thường có một thẻ mở và một thẻ đóng, và nội dung của phần tử được đặt giữa chúng. Ví dụ: **<tag> Nội dung </tag>**.
  + Cấu trúc cơ bản của một tài liệu HTML bao gồm thẻ **<html>**, **<head>**, và **<body>**.
* **Thẻ và thuộc tính:**
  + HTML cung cấp một loạt các thẻ để đánh dấu và biểu diễn các phần tử trên trang web như văn bản, hình ảnh, đường liên kết, biểu mẫu, bảng, và nhiều hơn nữa.
  + Mỗi thẻ có thể có các thuộc tính (attributes) để cung cấp thông tin bổ sung về phần tử đó. Ví dụ: **<a href="url">**.
* **Các phần tử cơ bản của HTML:**
  + **<h1>**, **<h2>**, **<h3>**, ... **<h6>**: Đánh dấu tiêu đề cấp độ khác nhau.
  + **<p>**: Đánh dấu đoạn văn bản.
  + **<a>**: Đánh dấu một liên kết.
  + **<img>**: Đánh dấu hình ảnh.
  + **<ul>**, **<ol>**, **<li>**: Đánh dấu danh sách không thứ tự (unordered list) và danh sách có thứ tự (ordered list).
  + **<table>**, **<tr>**, **<td>**: Đánh dấu bảng và các hàng, ô trong bảng.
* **HTML5:**
  + HTML5 là phiên bản mới nhất của HTML, cung cấp các tính năng và thẻ mới để tạo ra các trang web phức tạp và hiện đại hơn.
  + Nó bao gồm các thẻ mới như **<header>**, **<footer>**, **<nav>**, **<article>**, **<section>** để tạo cấu trúc trang web rõ ràng và linh hoạt hơn.
  + HTML5 cũng hỗ trợ các tính năng mới như audio, video, canvas, và geolocation.

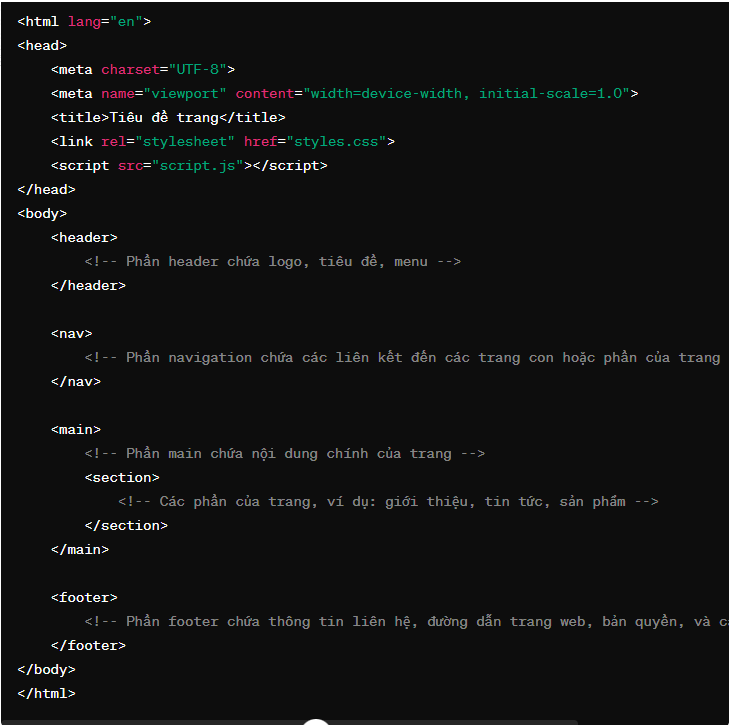
HTML là một ngôn ngữ cơ bản nhưng mạnh mẽ và quan trọng trong việc xây dựng trang web. Nó cung cấp cấu trúc cơ bản và khung nhìn cho việc tạo ra các trang web đa dạng và tương tác.



## **2.2. Cấu trúc tổng quát trang HTML**

* **Doctype Declaration (Khai báo Doctype):**
  + Đây là phần đầu tiên của một tài liệu HTML và nó chỉ định phiên bản của HTML mà trang web sử dụng. Ví dụ: **<!DOCTYPE html>**.
* **Thẻ <html>:**
  + Thẻ **<html>** là thẻ gốc của một tài liệu HTML và chứa toàn bộ nội dung của trang web.
* **Thẻ <head>:**
  + Thẻ **<head>** chứa các thông tin về tài liệu như tiêu đề của trang (**<title>**), các siêu dữ liệu (**<meta>**), các tập tin CSS (**<link>**), các đoạn mã JavaScript (**<script>**), và các tập tin favicon.
* **Thẻ <body>:**
  + Thẻ **<body>** chứa toàn bộ nội dung hiển thị của trang web bao gồm văn bản, hình ảnh, liên kết, các phần tử tương tác và nội dung động.

Dưới đây là một cấu trúc tổng quát của trang HTML:



Trong cấu trúc trên, mỗi phần có một vai trò riêng biệt trong việc tổ chức và hiển thị nội dung trang web. Phần head chứa thông tin metadata và các tài nguyên ngoài, phần body chứa toàn bộ nội dung hiển thị của trang web, bao gồm các phần header, navigation, main content và footer.



## **2.3. Các thẻ HTML thông dụng**



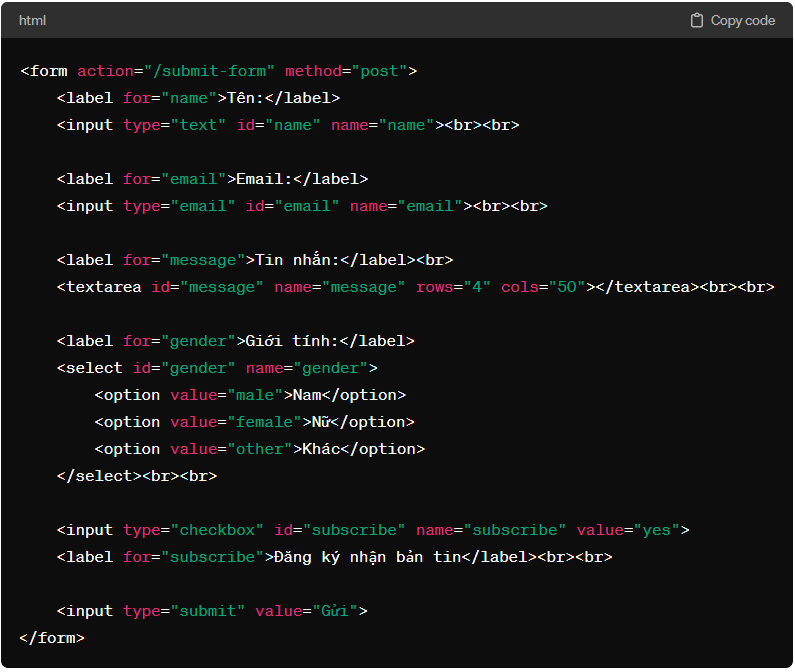
* **<h1>, <h2>, ..., <h6>:**
  + Được sử dụng để đánh dấu các tiêu đề cấp độ khác nhau trên trang web, với **<h1>** là tiêu đề lớn nhất và **<h6>** là tiêu đề nhỏ nhất.
* **<p>:**
  + Được sử dụng để đánh dấu đoạn văn bản.
* **<a>:**
  + Được sử dụng để tạo liên kết (hyperlink) đến các trang web khác hoặc các phần khác của cùng một trang web.
* **<img>:**
  + Được sử dụng để hiển thị hình ảnh trên trang web.
* **<ul>, <ol>, <li>:**
  + **<ul>**: Đánh dấu danh sách không thứ tự (unordered list).
  + **<ol>**: Đánh dấu danh sách có thứ tự (ordered list).
  + **<li>**: Đánh dấu mỗi mục trong danh sách.
* **<div>, <span>:**
  + **<div>**: Được sử dụng để tạo ra các phần tử phân cách và phân loại nội dung trên trang web.
  + **<span>**: Được sử dụng để đánh dấu các phần văn bản hoặc phần tử nhỏ trên trang web.
* **<table>, <tr>, <td>, <th>:**
  + **<table>**: Được sử dụng để tạo bảng trên trang web.
  + **<tr>**: Được sử dụng để đánh dấu một hàng trong bảng.
  + **<td>**: Được sử dụng để đánh dấu một ô trong bảng.
  + **<th>**: Được sử dụng để đánh dấu một ô tiêu đề trong bảng.
* **<form>, <input>, <textarea>, <button>:**
  + **<form>**: Được sử dụng để tạo biểu mẫu trên trang web.
  + **<input>**: Được sử dụng để tạo các ô nhập liệu, nút radio, checkbox, và nút gửi.
  + **<textarea>**: Được sử dụng để tạo ô văn bản đa dòng trong biểu mẫu.
  + **<button>**: Được sử dụng để tạo các nút trên trang web.
* **<header>, <footer>, <nav>, <section>, <article>:**
  + Các thẻ này được sử dụng để tạo cấu trúc trang web hiện đại và tương thích với HTML5.



## **2.4. Các thẻ tạo biểu mẫu (form)**

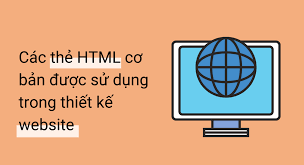
* **<form>:**
  + Thẻ **<form>** được sử dụng để tạo một biểu mẫu trên trang web.
  + Thuộc tính **action** xác định URL mà dữ liệu biểu mẫu sẽ được gửi đến.
  + Thuộc tính **method** xác định phương thức gửi dữ liệu, thường là GET hoặc POST.
* **<input>:**
  + Thẻ **<input>** tạo ra các ô nhập liệu trên biểu mẫu.
  + Thuộc tính **type** xác định loại dữ liệu mà ô nhập liệu chấp nhận, bao gồm text, password, checkbox, radio, number, và nhiều loại khác.
  + Thuộc tính **name** xác định tên của ô nhập liệu, được sử dụng khi gửi dữ liệu biểu mẫu lên máy chủ.
* **<textarea>:**
  + Thẻ **<textarea>** tạo ra một ô nhập liệu văn bản đa dòng trên biểu mẫu.
  + Thuộc tính **rows** và **cols** xác định số hàng và số cột của ô nhập liệu.
* **<select> và <option>:**
  + Thẻ **<select>** tạo ra một danh sách thả xuống cho người dùng chọn.
  + Thẻ **<option>** định nghĩa các mục trong danh sách thả xuống.
* **<button>:**
  + Thẻ **<button>** tạo ra một nút trên biểu mẫu, có thể là nút gửi (submit), nút đặt lại (reset), hoặc nút tùy chỉnh.
* **<label>:**
  + Thẻ **<label>** được sử dụng để tạo ra nhãn cho các ô nhập liệu trên biểu mẫu.
  + Thuộc tính **for** của **<label>** phải khớp với thuộc tính **id** của ô nhập liệu mà nhãn ánh xạ đến.

Ví dụ về một biểu mẫu cơ bản:



Biểu mẫu trên bao gồm ô nhập liệu văn bản, ô nhập liệu email, ô nhập liệu văn bản đa dòng, danh sách thả xuống, ô checkbox và nút gửi.

## **2.5. Một số thẻ HTML đặc biệt**



* **<meta>:**
  + Thẻ **<meta>** được sử dụng để cung cấp các thông tin meta-data về trang web.
  + Thông tin như charset, mô tả trang, từ khóa liên quan, và thông tin tác giả có thể được đặt trong thẻ **<meta>**.
  + Thẻ **<meta>** thường được đặt trong phần **<head>** của trang.
* **<link>:**
  + Thẻ **<link>** được sử dụng để liên kết tới các tài nguyên bên ngoài như các tập tin CSS, các biểu tượng favicon, hoặc các tệp font.
* **<script>:**
  + Thẻ **<script>** được sử dụng để nhúng mã JavaScript vào trang web.
  + Mã JavaScript có thể được đặt trong thẻ **<script>** hoặc được liên kết từ các tệp ngoài.
* **<style>:**
  + Thẻ **<style>** được sử dụng để đặt CSS trực tiếp trong tài liệu HTML.
  + CSS được đặt trong thẻ **<style>** sẽ áp dụng cho toàn bộ tài liệu HTML.
* **<iframe>:**
  + Thẻ **<iframe>** được sử dụng để nhúng một trang web hoặc một phần của một trang web vào trong một tài liệu HTML.
  + Điều này cho phép hiển thị nội dung từ các trang web khác trên trang web hiện tại.
* **<audio> và <video>:**
  + Thẻ **<audio>** và **<video>** được sử dụng để nhúng các phương tiện như âm thanh và video vào trang web.
  + Các thuộc tính như **src**, **controls**, và **autoplay** có thể được sử dụng để tùy chỉnh các phương tiện này.
* **<canvas>:**
  + Thẻ **<canvas>** được sử dụng để vẽ đồ họa và tạo đồ họa vector trên trang web bằng JavaScript.
  + Các hàm JavaScript được sử dụng để vẽ và tương tác với các hình vẽ trên **<canvas>**.
* **<section>, <header>, <footer>, <nav>, <article>:**
  + Các thẻ này là các phần tử semantically đặc biệt được sử dụng trong HTML5 để tạo ra cấu trúc trang web rõ ràng và dễ đọc hơn.

Những thẻ HTML đặc biệt này cung cấp cho bạn các công cụ và khả năng để tạo ra trang web đa dạng, tương tác và hấp dẫn. Bằng cách sử dụng chúng một cách hiệu quả, bạn có thể tạo ra trải nghiệm người dùng tốt và cung cấp các tính năng mạnh mẽ trên trang web của mình.



## **2.6. HTML5**

### HTML5 là gì?

HTML5 là phiên bản mới nhất của ngôn ngữ đánh dấu siêu văn bản (HTML) được sử dụng để tạo và cấu trúc các trang web. HTML5 cung cấp các tính năng và thẻ mới giúp cải thiện trải nghiệm người dùng, hỗ trợ nhiều định dạng phương tiện mới, cải thiện hiệu suất và hỗ trợ tích hợp các ứng dụng web phức tạp.

### Cú pháp HTML5

Cú pháp HTML5 không khác biệt nhiều so với các phiên bản trước của HTML. Tuy nhiên, HTML5 cung cấp một số thẻ mới và các tính năng như audio, video, canvas, và các phần tử semantic như **<header>**, **<footer>**, **<nav>**, **<section>**, và **<article>**.



### Thẻ ngữ nghĩa

HTML5 đưa ra một số thẻ ngữ nghĩa (semantic elements) để cải thiện khả năng đọc và hiểu cấu trúc của trang web bởi cả con người và máy tính. Một số thẻ ngữ nghĩa phổ biến trong HTML5 bao gồm:

* **<header>**: Định nghĩa phần đầu của một trang hoặc một phần tử.
* **<footer>**: Định nghĩa phần cuối của một trang hoặc một phần tử.
* **<nav>**: Định nghĩa một phần chứa các liên kết điều hướng.
* **<section>**: Định nghĩa một phần của một tài liệu hoặc ứng dụng web.
* **<article>**: Định nghĩa một phần tử độc lập, tự đủ trong tài liệu, trang, hoặc ứng dụng web.

### Webform

Webform là một phần quan trọng của HTML5, cho phép người dùng nhập dữ liệu và gửi nó đến máy chủ. Các thẻ HTML5 thường được sử dụng trong webform bao gồm **<input>**, **<textarea>**, **<select>**, **<button>** và nhiều thẻ khác. HTML5 cũng hỗ trợ các kiểu dữ liệu mới cho các ô nhập liệu như email, URL, số điện thoại và các yếu tố ngày tháng.

HTML5 cung cấp các tính năng mới cho webform như validation, placeholder text, autofocus, và các thuộc tính mới cho việc gửi dữ liệu mà không cần tải lại trang (AJAX). Điều này giúp tăng cường trải nghiệm người dùng và hiệu suất của ứng dụng web.



**CHƯƠNG III: CSS và CSS3**

**(Cascading Style Sheets)**

**3.1. CSS là gì?**

CSS là viết tắt của "Cascading Style Sheets" (Bảng kiểu theo dạng dòng xuôi). Đây là một ngôn ngữ lập trình được sử dụng để mô tả cách các phần tử HTML được hiển thị trên màn hình hoặc trình duyệt. CSS giúp định dạng và trang trí trang web, làm cho nó trở nên hấp dẫn và dễ đọc hơn.

CSS được sử dụng để kiểm soát cách mà trang web được hiển thị trên nhiều thiết bị và kích thước màn hình khác nhau. Bằng cách sử dụng CSS, bạn có thể:

Định dạng Text: Điều chỉnh kích thước, kiểu chữ, màu sắc và khoảng cách giữa các dòng văn bản.

Định dạng Màu sắc và Nền: Xác định màu sắc của văn bản, nền và các phần tử khác trên trang.

Bố cục (Layout): Điều chỉnh kích thước và vị trí của các phần tử trên trang, bao gồm cả bố cục đàn hồi (responsive layout) để phản ánh kích thước màn hình.

Định dạng và Kích thước Hình ảnh: Thay đổi kích thước, độ trong suốt và vị trí của hình ảnh.

Tạo Hiệu ứng và Chuyển động: CSS3 (phiên bản mới nhất của CSS) cung cấp nhiều khả năng để tạo hiệu ứng và chuyển động trên trang web.

Kiểm soát Trình duyệt và Thiết bị: Xử lý các sự khác biệt giữa các trình duyệt và thiết bị khác nhau để đảm bảo trang web hiển thị đồng đều.

Với CSS, các phần tử HTML có thể được thiết kế một cách linh hoạt và dễ dàng duy trì, giúp tạo ra trang web có giao diện đẹp mắt và chuyên nghiệp.

**3.2. Cú pháp CSS**

Cú pháp CSS (Cascading Style Sheets) được sử dụng để định dạng và trang trí các phần tử HTML trên trang web. Dưới đây là một số điểm chính về cú pháp CSS:

Chọn phần tử:

Để chọn một phần tử HTML, bạn sử dụng tên của nó: h1, p, div, v.v.

Để chọn một lớp (class), sử dụng dấu chấm: .class-name.

Để chọn một id, sử dụng dấu thăng: #id-name.

Thuộc tính và giá trị:

Mỗi quy tắc CSS bao gồm một hoặc nhiều cặp thuộc tính và giá trị.

Cặp thuộc tính và giá trị được phân tách bởi dấu hai chấm (:), và kết thúc bằng dấu chấm phảy (;).

Khối CSS:

Một khối CSS là một tập hợp các quy tắc CSS được đặt trong dấu ngoặc nhọn {}.

Mỗi quy tắc trong một khối cũng nên kết thúc bằng dấu chấm phảy (;).

Chú thích:

Chú thích trong CSS bắt đầu bằng /\* và kết thúc bằng \*/.

Chú thích được sử dụng để giải thích mã CSS và không ảnh hưởng đến việc hiển thị của trang web.

Đơn vị đo:

Các giá trị trong CSS thường kèm theo đơn vị đo như px (pixels), em, rem, %, v.v.

**3.3. Áp dụng CSS vào trang HTML**

Để áp dụng CSS vào trang HTML, có một số cách bạn có thể thực hiện, như sử dụng tệp CSS bên ngoài, thẻ <style> trong <head>, hoặc thuộc tính style trực tiếp trong các phần tử HTML. Dưới đây là một số cách phổ biến:

1. Tệp CSS Bên Ngoài:

Tạo một tệp CSS riêng biệt (ví dụ: styles.css):



Hình 38:

2. Thẻ <style> trong <head>:

Chèn CSS trực tiếp vào tài liệu HTML bằng thẻ <style> trong <head>:



Hình 39:

3. Thuộc tính style trực tiếp trong HTML:

Áp dụng CSS trực tiếp vào phần tử HTML bằng cách sử dụng thuộc tính style:

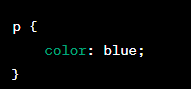


Hình 40:

**3.4. Selectors**

1. Selector Loại (Type Selector):

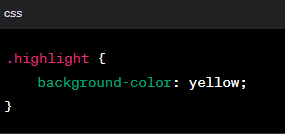
Chọn tất cả các phần tử cùng loại.



Hình 41:

2. Selector Class:

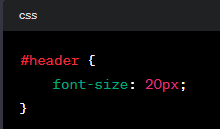
Chọn tất cả các phần tử có một class cụ thể.



Hình 42:

3. Selector ID:

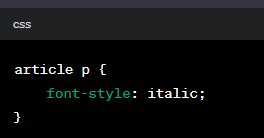
Chọn một phần tử có một ID cụ thể.



Hình 42:

4. Selector Phần Tử (Descendant Selector):

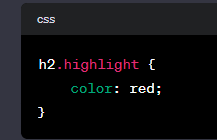
Chọn tất cả các phần tử con của một phần tử cụ thể



Hình 43:

5. Selector Kết hợp (Combination Selector):

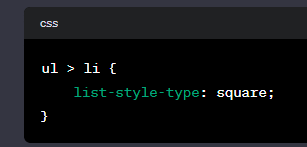
Kết hợp nhiều điều kiện chọn.



Hình 44:

6. Selector Phần Tử Con (Child Selector):

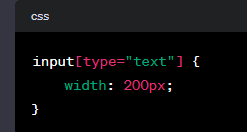
Chọn các phần tử con trực tiếp của một phần tử cụ thể.



Hình 45:

7. Selector Tiền Tố (Attribute Selector):

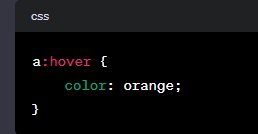
Chọn phần tử có một thuộc tính cụ thể.



Hình 46:

8. Selector Pseudo-class:

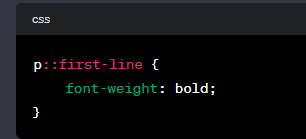
Chọn phần tử trong một trạng thái cụ thể hoặc vị trí trong văn bản.



Hình 47:

1. Selector Pseudo-element:

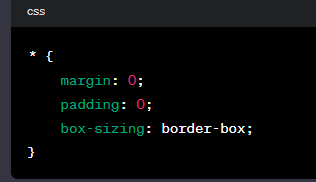
Chọn một phần của một phần tử, chẳng hạn như phần tử đầu tiên hoặc cuối cùng.



Hình 48:

**3.4.1. Universal selector**  
 Universal Selector là một loại selector trong CSS cho phép bạn chọn tất cả các phần tử trên trang web. Nó được biểu diễn bằng dấu \*. Khi bạn áp dụng CSS cho Universal Selector, quy tắc sẽ được áp dụng cho mọi phần tử trên trang, không phụ thuộc vào loại, class, hoặc id.

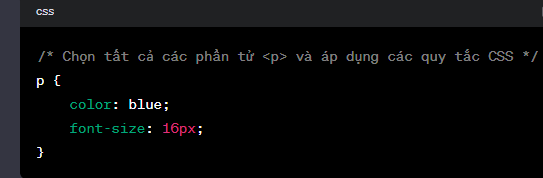
Dưới đây là một ví dụ sử dụng Universal Selector:



Hình 49:

**3.4.2. Type selector**

Type Selector (Selector Loại) trong CSS được sử dụng để chọn tất cả các phần tử có loại (tên) cụ thể. Cú pháp của Type Selector đơn giản là tên loại phần tử. Dưới đây là một ví dụ:

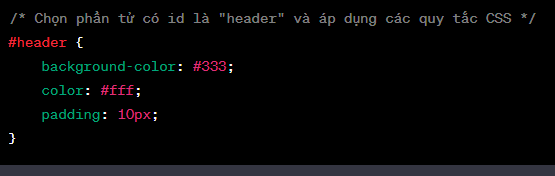


Hình 50:

**3.4.3. Identity selector**

ID Selector:

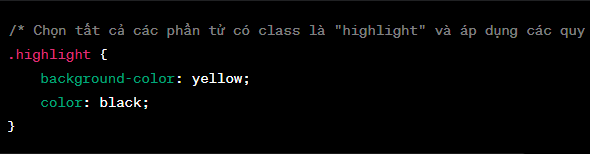
ID Selector được sử dụng để chọn một phần tử cụ thể dựa trên giá trị của thuộc tính id. Cú pháp của ID Selector là " #" sau đó là giá trị id. Ví dụ:



Hình 51:

**3.4.4. Class selector**

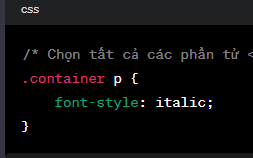
Class Selector trong CSS được sử dụng để chọn tất cả các phần tử có một class cụ thể. Cú pháp của Class Selector là dấu chấm (.) sau đó là tên của class. Ví dụ:



Hình 52:

**3.4.5. Descendant selector**

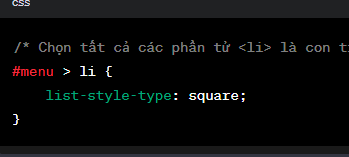
Descendant Selector (Selector Phần tử Con) trong CSS được sử dụng để chọn tất cả các phần tử con của một phần tử cụ thể. Cú pháp của Descendant Selector là sử dụng khoảng trắng ( ) giữa các tên phần tử. Ví dụ:



Hình 53:

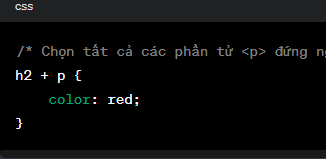
**3.4.6. Child selector**

Child Selector (Selector Phần tử Con) trong CSS được sử dụng để chọn tất cả các phần tử con trực tiếp của một phần tử cha cụ thể. Cú pháp của Child Selector là sử dụng dấu > giữa tên phần tử cha và phần tử con. Ví dụ:



Hình 54:

**3.4.7. Adjacent sibling selector**  
 Adjacent Sibling Selector trong CSS được sử dụng để chọn phần tử anh em (sibling) trực tiếp và kế tiếp của một phần tử cụ thể. Cú pháp của Adjacent Sibling Selector là sử dụng dấu + giữa các tên phần tử. Ví dụ:

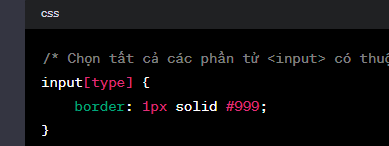


Hình 55:

**3.4.8. Attribute selector**

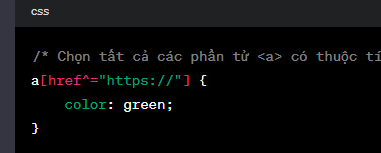
Attribute Selector (Selector Thuộc tính) trong CSS được sử dụng để chọn các phần tử dựa trên giá trị của thuộc tính của chúng. Có một số cách bạn có thể sử dụng Attribute Selector. Dưới đây là một số ví dụ:

Chọn phần tử có một thuộc tính cụ thể:



Hình 56:

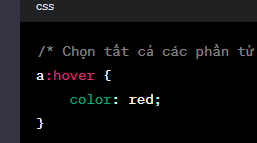
Chọn phần tử có một thuộc tính với giá trị cụ thể:



Hình 57:

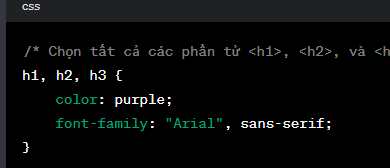
**3.4.9. Pseudo class selector**  
 Pseudo-class Selector (Selector Pseudo-class) trong CSS được sử dụng để chọn các phần tử trong một trạng thái hoặc vị trí cụ thể. Cú pháp của Pseudo-class Selector là sử dụng dấu hai chấm : sau tên phần tử. Dưới đây là một số ví dụ:

Chọn phần tử khi người dùng di chuột qua nó:



**Hình 58:**

**3.4.10. Group selector**  
 Group Selector trong CSS cho phép bạn kết hợp nhiều loại selector để áp dụng cùng một tập hợp quy tắc CSS cho nhiều phần tử. Cú pháp của Group Selector là sử dụng dấu phẩy , để ngăn cách giữa các selector. Dưới đây là một ví dụ:



Hình 59:

**3.5. Đơn vị đo lường CSS**

Trong CSS, có nhiều đơn vị đo lường được sử dụng để xác định kích thước và khoảng cách. Dưới đây là một số đơn vị đo lường phổ biến:

Pixels (px):

Pixels là đơn vị đo lường cụ thể và không phụ thuộc vào bất kỳ đơn vị đo lường nào khác.

Ví dụ: font-size: 16px; sẽ đặt kích thước font là 16 pixels.

Em (em):

1 em bằng kích thước của font hiện tại.

Nếu font-size của phần tử cha là 16px, font-size: 2em; ở phần tử con sẽ là 32px (16px \* 2).

Rem (rem):

1 rem bằng kích thước của font gốc (root element), thường là phần tử <html>.

font-size: 1.5rem; có nghĩa là 1.5 lần kích thước font của phần tử <html>.

Percentage (%):

Phần trăm thường được sử dụng trong việc xác định kích thước, khoảng cách, và các thuộc tính khác dựa trên kích thước của phần tử cha.

width: 50%; sẽ đặt chiều rộng bằng 50% chiều rộng của phần tử cha.

Viewport Width (vw) và Viewport Height (vh):

1vw bằng 1% chiều rộng của viewport (khu vực nhìn thấy trên màn hình).

1vh bằng 1% chiều cao của viewport.

width: 50vw; sẽ đặt chiều rộng bằng 50% chiều rộng của viewport.

Viewport Minimum (vmin) và Viewport Maximum (vmax):

1vmin bằng 1% của giá trị nhỏ nhất giữa chiều rộng và chiều cao của viewport.

1vmax bằng 1% của giá trị lớn nhất giữa chiều rộng và chiều cao của viewport.

Inch (in), Centimeter (cm), Millimeter (mm), Point (pt), Pica (pc):

Các đơn vị đo lường thực tế dựa trên đơn vị đo lường của hệ thống đo lường vật lý.

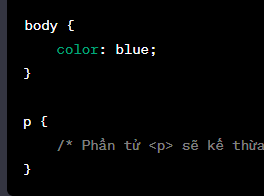
**3.6. Kế thừa thuộc tính**

Trong CSS, kế thừa thuộc tính là khả năng của một phần tử con nhận giá trị của một số thuộc tính từ phần tử cha của nó. Tuy nhiên, không tất cả các thuộc tính đều được kế thừa, và mức độ kế thừa có thể thay đổi tùy thuộc vào thuộc tính cụ thể.

Dưới đây là một số thuộc tính thường được kế thừa:

Màu sắc và Font:

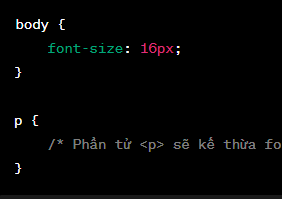
color: Màu chữ thường được kế thừa từ phần tử cha.



Hình 60:

Kích thước:

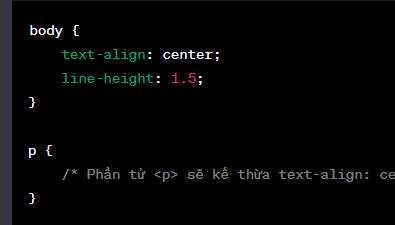
font-size: Kích thước font cũng có thể được kế thừa.



Hình 61:

Thuộc tính vị trí:

text-align, line-height: Các thuộc tính vị trí cũng có thể được kế thừa.



Hình 62:

**3.7. Các nhóm thuộc tính trong CSS**

- Thuộc tính định dạng font chữ, văn bản (font, text)  
 Thuộc tính định dạng font chữ và văn bản trong CSS giúp bạn kiểm soát các khía cạnh như loại font, kích thước, màu sắc, độ dày, và các thuộc tính liên quan đến văn bản. Dưới đây là một số thuộc tính quan trọng:

font-family:

Xác định loại font được sử dụng cho văn bản.

Nếu font mong muốn không tồn tại trên máy tính của người dùng, nó sẽ thử các font backup được liệt kê.

font-size:

Đặt kích thước của font. Có thể sử dụng các đơn vị như px, em, rem, %.

font-weight:

Xác định độ đậm nhẹ của font, thường là giá trị từ 100 đến 900 hoặc từ 'normal' đến 'bold'.

font-style:

Đặt kiểu font, có thể là 'normal', 'italic', hoặc 'oblique'.

text-align:

Xác định vị trí của văn bản bên trong một phần tử, có thể là 'left', 'right', 'center', hoặc 'justify'.

line-height:

Đặt chiều cao của dòng văn bản, có thể sử dụng các đơn vị như px, em, rem, %.

letter-spacing:

Đặt khoảng cách giữa các ký tự.

text-decoration:

Xác định trang trí văn bản như gạch chân, gạch ngang, hoặc hủy gạch.

css

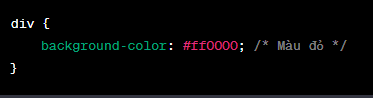
color:

Xác định màu chữ của văn bản.

- Thuộc tính định dạng nền (background)

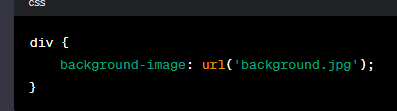
Thuộc tính định dạng nền (background) thường được sử dụng trong lập trình web và CSS để xác định hoặc tùy chỉnh nền của một phần tử HTML. Dưới đây là một số thuộc tính phổ biến liên quan đến định dạng nền:

background-color: Xác định màu sắc của nền. Bạn có thể sử dụng tên màu, mã màu hex, hoặc RGB để đặt giá trị cho thuộc tính này.



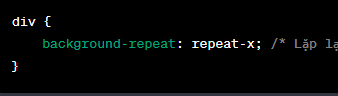
Hình 63:

background-image: Sử dụng để đặt hình ảnh làm nền.



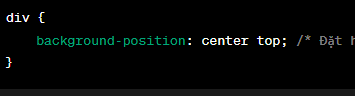
Hình 64:

background-repeat: Xác định cách hình ảnh nền được lặp lại. Các giá trị thường là repeat, repeat-x, repeat-y, hoặc no-repeat.



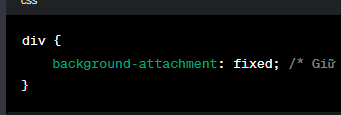
Hình 65:

background-position: Đặt vị trí của hình ảnh nền. Bạn có thể sử dụng giá trị như top, bottom, left, right, hoặc cặp giá trị trong đơn vị đo lường như px, %.



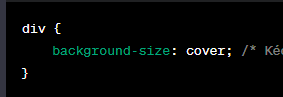
Hình 66:

background-attachment: Xác định liệu hình ảnh nền có di chuyển khi cuộn trang hay không. Các giá trị thường là scroll hoặc fixed.



Hình 67:

background-size: Đặt kích thước của hình ảnh nền. Các giá trị có thể là auto, contain, hoặc cover.



Hình 68:

- Mô hình hộp (box model)

Mô hình hộp (box model) là một khái niệm quan trọng trong CSS, mô tả cách mà mỗi phần tử HTML được hiển thị trong trang web. Nó bao gồm các thành phần chính sau đây:

Content (Nội dung): Là phần chứa nội dung thực sự của phần tử, chẳng hạn như văn bản, hình ảnh, hoặc các phần tử con.

Padding (Độ lề): Là khoảng cách giữa ranh giới của phần nội dung và ranh giới của phần đệm xung quanh nó. Nó giúp tăng khoảng cách giữa nội dung và ranh giới của phần tử.

Border (Viền): Là ranh giới xung quanh phần đệm. Bạn có thể đặt kiểu, màu sắc và độ dày cho viền.

Margin (Lề): Là khoảng cách giữa ranh giới của phần viền và ranh giới của các phần tử khác xung quanh nó. Nó giúp kiểm soát khoảng cách giữa phần tử hiện tại và các phần tử xung quanh.

Trong CSS, bạn có thể kiểm soát kích thước và khoảng cách của các thành phần này thông qua các thuộc tính như:

width và height: Đặt kích thước của phần tử.

padding: Đặt độ lề xung quanh nội dung.

border: Đặt độ dày, kiểu, và màu sắc của viền.

margin: Đặt lề giữa phần tử và các phần tử khác.

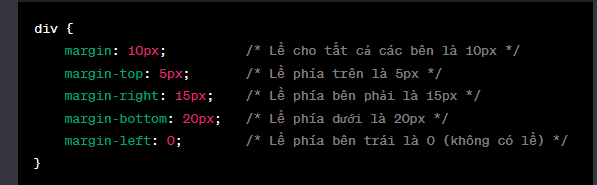
- Các thuộc tính định margin, padding, border

Dưới đây là các thuộc tính CSS liên quan đến các thành phần của mô hình hộp (box model): margin, padding, và border.

1. Margin (Lề)

margin: Đặt lề cho tất cả các bên (top, right, bottom, left).

margin-top, margin-right, margin-bottom, margin-left: Đặt lề cho từng bên cụ thể.

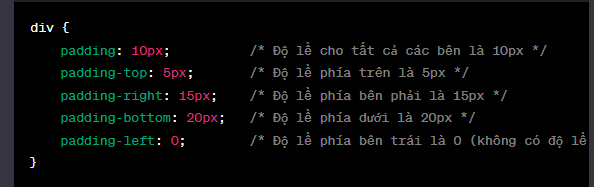


Hình 69

Padding (Độ lề)

padding: Đặt độ lề cho tất cả các bên (top, right, bottom, left).

padding-top, padding-right, padding-bottom, padding-left: Đặt độ lề cho từng bên cụ thể.



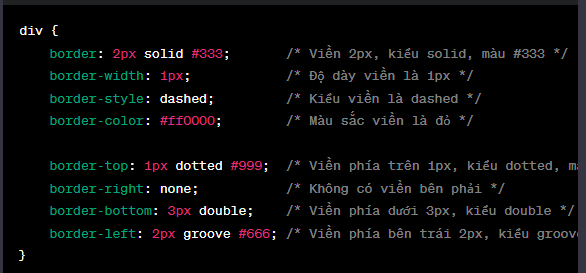
Hình 70:

Border (Viền)

border: Đặt thông số cho viền. Bao gồm độ dày, kiểu, và màu sắc.

border-width, border-style, border-color: Đặt độ dày, kiểu, và màu sắc của viền.

border-top, border-right, border-bottom, border-left: Đặt thông số của viền cho từng bên cụ thể.



Hình 71:

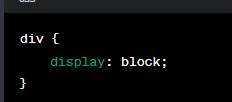
Những thuộc tính trên giúp bạn kiểm soát khoảng cách (margin), độ lề (padding), và trang trí viền (border) của các phần tử trên trang web của bạn.

- Thuộc tính định dạng cách hiển thị (display)

Thuộc tính display trong CSS được sử dụng để xác định cách một phần tử HTML được hiển thị trên trang web. Các giá trị của thuộc tính này quyết định kiểu hiển thị của phần tử, và có thể ảnh hưởng đến cách mà nó tương tác với các phần tử khác xung quanh.

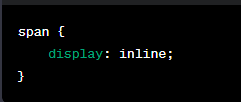
Dưới đây là một số giá trị phổ biến của thuộc tính display:

block: Phần tử sẽ hiển thị như một khối và chiếm toàn bộ chiều rộng của phần tử cha. Các phần tử block sẽ xuống dòng mới sau mỗi phần tử khác.



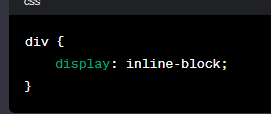
Hình 72:

inline: Phần tử sẽ hiển thị như một phần tử nội dung trên cùng một dòng với phần tử trước và sau nó, nhưng chiều rộng và chiều cao chỉ chiếm phần cần thiết.



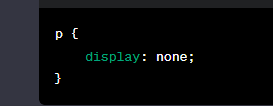
Hình 73:

inline-block: Kết hợp tính chất của inline và block. Phần tử sẽ hiển thị như một phần tử nội dung trên cùng một dòng và có thể đặt chiều rộng và chiều cao.



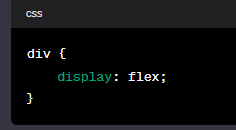
Hình 74:

none: Phần tử sẽ bị ẩn và không chiếm không gian trên trang. Điều này giống như việc phần tử không tồn tại.



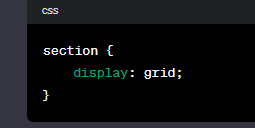
Hình 75:

flex: Sử dụng với mô hình linh động (flexbox), phần tử trở thành một container linh động cho các phần tử con.



Hình 76:

grid: Sử dụng với mô hình lưới (grid), phần tử trở thành một container lưới cho các phần tử con.



Hình 77:

Những giá trị này cung cấp linh động trong việc quản lý bố cục và hiển thị của các phần tử trong trang web của bạn. Sự lựa chọn giữa các giá trị này phụ thuộc vào yêu cầu cụ thể của bạn về bố cục và thiết kế trang web.

- Thuộc tính xác định vị trí (position)

Thuộc tính position trong CSS được sử dụng để xác định cách một phần tử HTML được định vị trong trang web. Các giá trị của thuộc tính này quyết định cách mà phần tử tương tác với các phần tử xung quanh và cách nó được đặt vị trí trên trang.

Dưới đây là một số giá trị phổ biến của thuộc tính position:

static: Là giá trị mặc định. Phần tử sẽ nằm trong luồng bình thường của trang, và bố cục sẽ không được ảnh hưởng bởi các thuộc tính top, right, bottom, và left.

relative: Phần tử sẽ được đặt vị trí dựa trên vị trí của nó trong luồng bình thường, sau đó có thể được di chuyển bằng cách sử dụng các thuộc tính top, right, bottom, và left.

absolute: Phần tử sẽ được đặt vị trí dựa trên phần tử cha gần nhất có thuộc tính position khác static. Nếu không có phần tử cha nào thỏa điều kiện, phần tử sẽ được đặt vị trí dựa trên thẻ <html>.

fixed: Phần tử sẽ được đặt vị trí dựa trên cửa sổ trình duyệt, và sẽ giữ vị trí ngay cả khi trang web được cuộn xuống hoặc lên.

sticky: Phần tử sẽ hoạt động như relative cho đến khi nó chạm vào biên trên hoặc dưới của trình duyệt, sau đó nó sẽ được đặt vị trí như fixed.

Các giá trị của thuộc tính position cung cấp linh động trong việc định vị và điều chỉnh vị trí của các phần tử trong trang web. Sự lựa chọn giữa chúng phụ thuộc vào yêu cầu cụ thể của bạn về bố cục và tương tác của phần tử.

**3.8. Float &amp; Clear**

Thuộc tính float trong CSS được sử dụng để di chuyển một phần tử về bên trái hoặc bên phải của phần tử cha của nó, và nó tạo ra một phần tử chảy trên cạnh đó. Trong khi đó, thuộc tính clear được sử dụng để xác định xem phải ngăn chặn các phần tử chảy theo bên trái, bên phải, cả hai hoặc không có phần tử nào.

Dưới đây là một mô tả về cách sử dụng float và clear:

Float (float):

float: left;: Phần tử sẽ được đặt về phía bên trái và các phần tử khác sẽ chảy xung quanh nó về phía bên phải.

float: right;: Phần tử sẽ được đặt về phía bên phải và các phần tử khác sẽ chảy xung quanh nó về phía bên trái.

Clear (clear):

clear: left;: Ngăn chặn phần tử chảy theo bên trái của các phần tử đã được đặt float: left;.

clear: right;: Ngăn chặn phần tử chảy theo bên phải của các phần tử đã được đặt float: right;.

clear: both;: Ngăn chặn phần tử chảy theo cả hai bên, không theo bên trái hoặc bên phải của các phần tử đã được đặt float.

Thuộc tính clear thường được sử dụng để giải quyết vấn đề về bố cục khi có các phần tử đã được đặt float và bạn muốn đảm bảo rằng các phần tử sau đó không chảy theo bên trái hoặc bên phải của chúng.

**3.9. Flex**

CSS Flexbox (Flexible Box) là một mô hình bố cục được thiết kế để tạo ra bố cục linh hoạt và hiệu quả cho các phần tử trong trang web. Thuộc tính chính liên quan đến Flexbox là display: flex; và display: inline-flex;. Dưới đây là một mô tả về cách sử dụng Flexbox:

Bắt đầu với Flexbox:

Hiển thị Flex Container:

display: flex;: Biến một phần tử thành một Flex Container.

display: inline-flex;: Biến một phần tử thành một Flex Container nhưng nói chung sẽ hiển thị như một phần tử inline.

Các thuộc tính trên Flex Container:

Hướng Chính và Ngược của Flex Container:

flex-direction: Xác định hướng chính của các phần tử con bên trong Flex Container.

Xếp chồng và Canh lề Các Phần Tử Con:

flex-wrap: Quyết định liệu các phần tử con có nên xếp chồng khi không đủ không gian hay không.

justify-content: Canh lề theo chiều chính của Flex Container.

align-items: Canh lề theo chiều phụ của Flex Container.

Các thuộc tính trên Flex Items (Phần Tử Con):

Kích Thước và Xếp Cột:

flex: Xác định kích thước tỷ lệ của phần tử con trong Flex Container.

**3.10. Grid**

CSS Grid Layout là một mô hình bố cục mạnh mẽ, cho phép bạn tạo ra bố cục linh hoạt với các hàng và cột. Dưới đây là một mô tả về cách sử dụng CSS Grid:

Bắt đầu với Grid:

Hiển thị Grid Container:

display: grid;: Biến một phần tử thành một Grid Container.

display: inline-grid;: Biến một phần tử thành một Grid Container nhưng nói chung sẽ hiển thị như một phần tử inline.

Các thuộc tính trên Grid Container:

Định Dạng Các Hàng và Cột:

grid-template-rows và grid-template-columns: Định nghĩa kích thước của các hàng và cột trong Grid Container.

Đặt Vị Trí Các Hàng và Cột:

grid-row và grid-column: Xác định vị trí của một phần tử trong Grid Container.

Các Thuộc Tính Bổ Sung:

Justify và Align Các Phần Tử:

justify-content: Căn lề theo chiều chính của Grid Container.

align-content: Căn lề theo chiều phụ của Grid Container.

**3.11. CSS3**  
 - CSS3 là gì?  
 CSS3 là phiên bản thứ ba của Cascading Style Sheets (CSS), một ngôn ngữ mô tả kiểu dáng của trang web. CSS3 đưa ra nhiều cải tiến so với phiên bản trước đó, CSS2, bằng cách thêm vào nhiều tính năng mới và mở rộng khả năng thiết kế của lập trình viên và nhà phát triển web.

Dưới đây là một số điểm nổi bật của CSS3:

Hiệu Ứng và Chuyển Động: CSS3 cung cấp khả năng tạo hiệu ứng và chuyển động mà không cần sử dụng JavaScript. Điều này bao gồm transitions (chuyển động mềm mại khi thay đổi trạng thái), transforms (biến đổi hình dạng, kích thước, vị trí), và animations (hoạt hình phức tạp).

Văn Bản và Phông Chữ: CSS3 cung cấp khả năng sử dụng các phông chữ tùy chỉnh thông qua @font-face, tạo shadow cho văn bản (text-shadow), và các tính năng khác để làm cho văn bản trở nên đa dạng và thú vị.

Bố Cục và Định Dạng: CSS3 cung cấp các tính năng như box-sizing để quản lý kích thước của các phần tử, media queries để phản ánh phản ứng với kích thước màn hình, và mô hình grid layout để tạo ra bố cục đa cột.

Hình Ảnh và Nền: CSS3 cho phép sử dụng hình nền gradient, bo góc của các phần tử (border-radius), và các hiệu ứng như box shadow để tạo chiều sâu và kết cấu.

Truy Cập và Chọn Lọc: Các tiện ích chọn lọc trong CSS3 như :nth-child và :nth-of-type giúp lựa chọn các phần tử dựa trên vị trí của chúng trong cấu trúc DOM, còn các attribute selectors giúp lựa chọn phần tử dựa trên thuộc tính của chúng.

CSS3 không phải là một phiên bản chính thức, mà là một tập hợp các mô-đun và tính năng được phát triển độc lập, cho phép người phát triển web lựa chọn và sử dụng theo nhu cầu của dự án cụ thể. Việc sử dụng CSS3 giúp tăng cường khả năng thiết kế và trải nghiệm người dùng trên trang web.

- CSS3 Selectors

CSS3 Selectors là một phần quan trọng của CSS3, cung cấp cách chọn lọc và áp dụng các kiểu dáng cho các phần tử HTML. Dưới đây là một số CSS3 Selectors phổ biến:

1. Selector cơ bản:

Element Selector: Chọn tất cả các phần tử của một loại nào đó.

2. Selectors Kết Hợp:

Descendant Selector: Chọn tất cả các phần tử con của một phần tử.

3. Pseudo-classes và Pseudo-elements:

Pseudo-classes: Làm thay đổi kiểu dáng dựa trên trạng thái hoặc hành động của phần tử.

4. Attribute Selectors:

Attribute Equals Selector: Chọn các phần tử có thuộc tính có giá trị cụ thể.

5. Structural Selectors:

:first-child, :last-child, :only-child: Chọn phần tử là con đầu tiên, con cuối cùng, hoặc là con duy nhất của phần tử cha.

CSS3 Selectors cung cấp rất nhiều cách để chọn lọc phần tử và áp dụng kiểu dáng một cách linh hoạt và mạnh mẽ. Sử dụng ch

- CSS3 Rounded Corners

Trong CSS3, bạn có thể sử dụng thuộc tính border-radius để tạo các góc cong cho các phần tử. Thuộc tính này cho phép bạn làm cho các hình dạng khối trở nên mềm mại và thân thiện hơn. Dưới đây là cách sử dụng border-radius để tạo góc cong:

Cú Pháp Cơ Bản:

Bạn có thể sử dụng border-radius với giá trị duy nhất để tạo các góc cong giống nhau cho tất cả bốn góc:

Tùy Chỉnh Góc Cho Mỗi Góc:

Nếu bạn muốn tùy chỉnh độ cong của từng góc, bạn có thể sử dụng bốn giá trị (theo thứ tự: góc trên bên trái, góc trên bên phải, góc dưới bên phải, góc dưới bên trái):

Tùy Chỉnh Góc Cho Góc Cụ Thể:

Nếu bạn chỉ muốn tùy chỉnh một số góc cụ thể, bạn có thể sử dụng các giá trị riêng lẻ:

Góc Cong Tính Theo Phần Trăm:

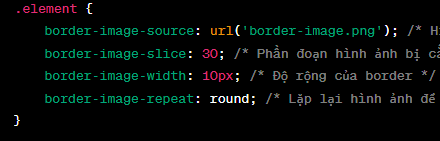
Bạn cũng có thể sử dụng phần trăm để đặt giá trị của border-radius, giúp tạo ra các góc cong linh hoạt hơn:

Các giá trị của border-radius có thể được kết hợp và điều chỉnh để tạo ra các hình dạng góc cong độc đáo cho các phần tử trên trang web của bạn. Sử dụng border-radius có thể làm cho giao diện trang web trở nên thú vị và thân thiện hơn.

- CSS3 Border Images

CSS3 Border Images cho phép bạn sử dụng hình ảnh làm nền cho các đường viền của phần tử. Điều này mang lại khả năng tạo ra đường viền có hình ảnh, đặc biệt là những đường viền phức tạp hơn so với việc chỉ sử dụng màu và kiểu đơn giản. Dưới đây là cách sử dụng CSS3 Border Images:

Cú Pháp Cơ Bản:



Hình 78:

Các Thuộc Tính:

border-image-source: Xác định hình ảnh sử dụng cho border.

border-image-slice: Xác định phần của hình ảnh sẽ được sử dụng làm border, được đo bằng pixel hoặc phần trăm.border-image-width: Đặt độ rộng của border. Nó có thể là giá trị pixel, phần trăm hoặc một giá trị tỷ lệ.

border-image-outset: Đặt độ rộng của vùng biên giữa border và nội dung. Cũng có thể là giá trị pixel, phần trăm hoặc tỷ lệ.

border-image-repeat: Đặt cách thức lặp lại hình ảnh trên các border.

stretch: Mở rộng hình ảnh để fill toàn bộ border.

repeat: Lặp lại hình ảnh để fill toàn bộ border.

round: Làm đầy đủ border bằng cách lặp lại hình ảnh và điều chỉnh kích thước.

- CSS3 Backgrounds  
 CSS3 cung cấp nhiều tính năng mới cho việc định dạng nền của phần tử HTML. Dưới đây là một số tính năng quan trọng của CSS3 Backgrounds:

1. Background Color:

2. Background Image:

3. Background Repeat:

4. Background Position:

5. Background Size:

6. Background Attachment:

7. Multiple Backgrounds:

8. Linear Gradient:

9. Radial Gradient:

10. Multiple Background Layers:

11. Background Clip:

12. Background Origin:

13. Background Attachment:

14. Background Shorthand:

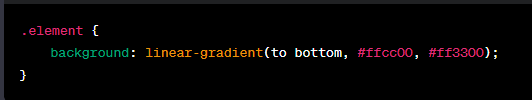
15. Background Blend Mode:

- CSS Gradients

CSS Gradients cho phép bạn tạo ra hiệu ứng màu sắc chuyển động và mềm mại trên nền của phần tử, thay vì chỉ sử dụng một màu đơn lẻ. Dưới đây là một số cách sử dụng CSS Gradients:

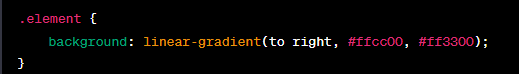
Linear Gradients:

1. Dọc (Top to Bottom):



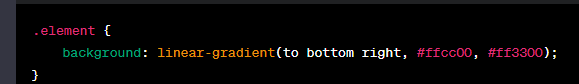
Hình 79:

1. Ngang (Left to Right):



Hình 80:

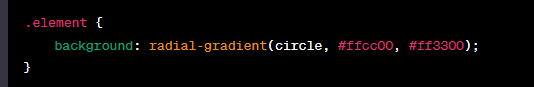
1. Đường chéo:



Hình 81:

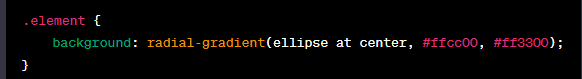
Radial Gradients:

1. Radial:



Hình 82:

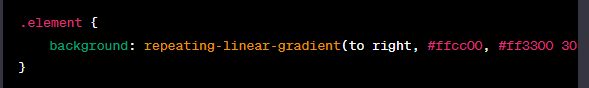
1. Elongated Radial:



Hình 83:

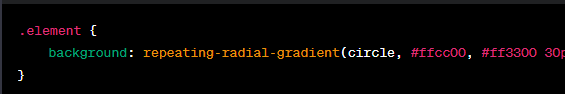
Repeating Gradients:

1. Linear Repeating:



Hình 84:

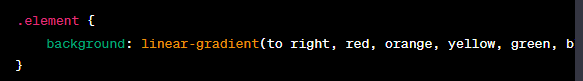
1. Radial Repeating:



Hình 85:

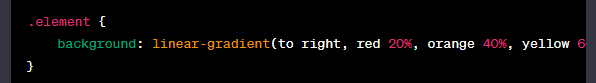
Color Stops:

1. Sử dụng Nhiều Màu:



Hình 86:

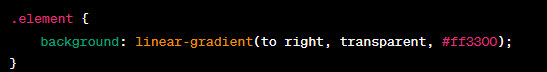
1. Sử dụng Màu và Điểm Dừng:



Hình 87:

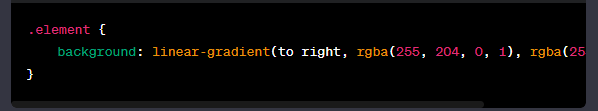
Transparent Gradients:

1. Màu Trong Suốt:



Hình 88:

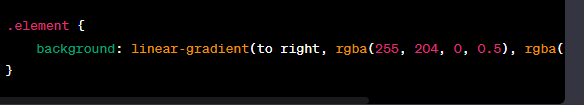
1. Trộn Giữa Màu và Trong Suốt:



Hình 89:

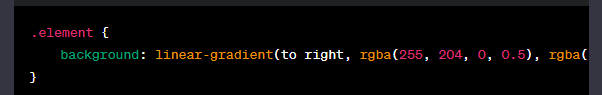
Kết hợp với Background:

1. Kết hợp với Hình Ảnh:



Hình 90:

13. Kết hợp với Hình Ảnh và Color Stops:



Hình 91:

- CSS3 Shadows

CSS3 Shadows cho phép bạn thêm đổ bóng cho các phần tử HTML, tạo chiều sâu và kết cấu. Dưới đây là cách sử dụng các thuộc tính đổ bóng trong CSS3:

1. Box Shadow:

Offset X, Offset Y: Điều chỉnh vị trí đổ bóng theo trục x và y.

Blur Radius: Xác định độ mờ của bóng.

Color: Màu của đổ bóng.

2. Text Shadow:

Offset X, Offset Y: Điều chỉnh vị trí đổ bóng văn bản theo trục x và y.

Blur Radius: Xác định độ mờ của bóng.

Color: Màu của đổ bóng.

3. Multiple Shadows:

Bạn có thể thêm nhiều bóng khác nhau bằng cách liệt kê chúng, mỗi bóng được ngăn cách bằng dấu phẩy.

4. Inset Shadows:

Inset: Chỉ định đổ bóng ở bên trong phần tử thay vì ở bên ngoài.

5. Ambient Shadows (Ambient Light):

Sử dụng màu có độ trong suốt (alpha) để tạo hiệu ứng đổ bóng giống như ánh sáng xung quanh.

6. Neumorphism Shadows:

Sử dụng đồng thời bóng âm và bóng sáng để tạo hiệu ứng neumorphism.

7. Responsive Shadows:

Có thể sử dụng media queries để thay đổi đổ bóng dựa trên kích thước màn hình.

- CSS3 Fonts

CSS3 Fonts cung cấp nhiều tính năng mới và cải tiến cho việc quản lý và thiết lập kiểu chữ trên trang web của bạn. Dưới đây là một số thuộc tính và tính năng liên quan đến CSS3 Fonts:

1. Font Family:

Font Family: Xác định các kiểu chữ sẽ được sử dụng. Nếu kiểu chữ đầu tiên không khả dụng, trình duyệt sẽ chuyển sang kiểu chữ thứ hai và tiếp tục như vậy.

2. Font Size:

Font Size: Xác định kích thước của chữ.

3. Font Weight:

Font Weight: Xác định độ đậm nhạt của chữ. Các giá trị phổ biến bao gồm: normal, bold, bolder, lighter, và các giá trị từ 100 đến 900.

4. Font Style:

Font Style: Xác định kiểu chữ là chữ bình thường, nghiêng (italic) hoặc in đậm (oblique).

5. Font Variant:

Font Variant: Xác định kiểu chữ là chữ thường, chữ in hoa (uppercase), hoặc chữ in hoa với kích thước nhỏ (small-caps).

6. Line Height:

Line Height: Xác định khoảng cách giữa các dòng văn bản, thường được đo bằng số không có đơn vị hoặc bằng pixel hoặc em.

7. Text Transform:

Text Transform: Chuyển đổi văn bản thành chữ in hoa (uppercase), chữ thường (lowercase), hoặc chữ cái đầu tiên của mỗi từ thành in hoa (capitalize).

8. Letter Spacing:

Letter Spacing: Xác định khoảng cách giữa các chữ cái.

9. Word Spacing:

Word Spacing: Xác định khoảng cách giữa các từ.

10. Font Shorthand:

Font Shorthand: Cung cấp cách ngắn gọn để thiết lập nhiều thuộc tính font cùng một lúc.

11. Web Fonts:

@font-face: Cho phép bạn sử dụng font tải về từ máy chủ.

12. Font Display:

Font Display: Xác định cách trình duyệt hiển thị font trong quá trình tải về.

13. Variable Fonts:

Variable Fonts: Cho phép điều chỉnh độ dày, nghiêng, và nhiều thuộc tính khác của font.

14. Font Feature Settings:

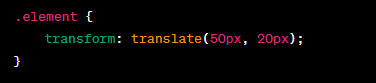
Font Feature Settings: Cho phép bật tắt các tính năng của font, như ligatures.

Các thuộc tính và tính năng trên giúp bạn quản lý kiểu chữ một cách linh hoạt và tùy chỉnh trên trang web của bạn.

- CSS3 2D Transforms

CSS3 2D Transforms cung cấp các tính năng để biến đổi phần tử HTML trên mặt phẳng 2 chiều. Dưới đây là một số thuộc tính chính của CSS3 2D Transforms:

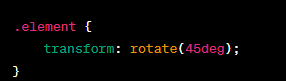
1. Translate (Dịch chuyển):



Hình 92:

Translate X, Translate Y: Di chuyển phần tử theo trục x và y.

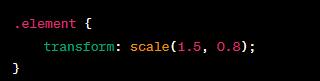
2. Rotate (Xoay):



Hình 93:

Rotate: Xoay phần tử một góc cụ thể. Đơn vị mặc định là độ (deg).

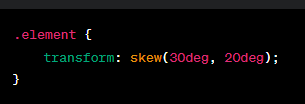
3. Scale (Thu phóng):



Hình 94:

Scale X, Scale Y: Thu phóng phần tử theo trục x và y.

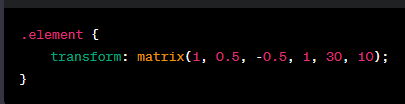
4. Skew (Nghiêng):



Hình 95:

Skew X, Skew Y: Nghiêng phần tử theo trục x và y.

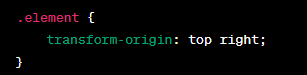
5. Matrix Transform:



Hình 96:

Matrix: Biến đổi phần tử sử dụng ma trận 2x3.

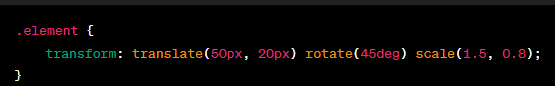
6. Transform Origin:



Hình 97:

Transform Origin: Xác định điểm trục xoay (origin) cho các biến đổi.

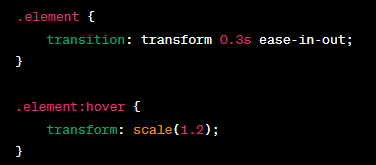
7. Multiple Transforms:



Hình 98:

Bạn có thể kết hợp nhiều phép biến đổi trong một luồng duy nhất.

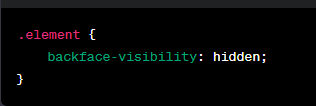
8. Transition (Chuyển động):



Hình 99:

Sử dụng cùng với CSS Transitions để tạo hiệu ứng chuyển động mượt mà.

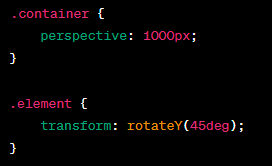
9. Backface Visibility:



Hình 100:

Backface Visibility: Xác định sự hiển thị của mặt sau của phần tử khi nó được quay.

10. Perspective (Chưa hoàn chỉnh):



Hình 101:

Perspective: Xác định khoảng cách giữa mắt người nhìn và mặt phẳng trước của phần tử. Sử dụng cùng với 3D Transforms.

Các tính năng trên giúp bạn thực hiện các biến đổi đơn giản hoặc phức tạp trên các phần tử trong mặt phẳng 2 chiều, tạo ra hiệu ứng và giao diện động trên trang web của bạn.

- CSS3 3D Transforms

CSS3 3D Transforms mở rộng khả năng biến đổi của CSS3 bằng cách thêm khả năng làm việc với chiều sâu, cho phép bạn thực hiện biến đổi trong không gian 3 chiều. Dưới đây là một số thuộc tính chính của CSS3 3D Transforms:

1. RotateX, RotateY, RotateZ:

RotateX, RotateY, RotateZ: Xoay phần tử quanh trục X, Y, hoặc Z.

2. Translate3D:

Translate3D: Di chuyển phần tử theo các giá trị x, y, z.

3. Scale3D:

Scale3D: Thu phóng phần tử theo các giá trị x, y, z.

4. Matrix3D:

Matrix3D: Biến đổi phần tử sử dụng ma trận 4x4.

5. Perspective (Chưa hoàn chỉnh):

Perspective: Xác định khoảng cách giữa mắt người nhìn và mặt phẳng trước của phần tử.

6. Backface Visibility:

Backface Visibility: Xác định sự hiển thị của mặt sau của phần tử khi nó được quay.

7. Transform Style:

Transform Style: Xác định cách các phần tử con của phần tử cha sẽ thực hiện biến đổi.

8. Transition (Chuyển động):

Sử dụng cùng với CSS Transitions để tạo hiệu ứng chuyển động mượt mà.

9. Perspective Origin:

Perspective Origin: Xác định điểm trung tâm từ đó perspective sẽ được áp dụng.

Các thuộc tính trên giúp bạn tạo ra hiệu ứng 3D trực quan và động, mở ra nhiều khả năng sáng tạo trong thiết kế web. Tuy nhiên, cần lưu ý rằng hỗ trợ cho một số thuộc tính này có thể khác nhau trên các trình duyệt, đặc biệt là trong các phiên bản cũ.

- CSS3 Transitions

CSS3 Transitions là một tính năng cho phép bạn tạo ra hiệu ứng chuyển động mượt mà khi thuộc tính của phần tử thay đổi. Dưới đây là một số điểm quan trọng và cú pháp cơ bản khi sử dụng CSS3 Transitions:

Cú Pháp Cơ Bản:

property: Thuộc tính cần thay đổi (ví dụ: opacity, background-color, transform,...).

duration: Thời gian diễn ra chuyển động (ví dụ: 0.5s, 2s,...).

timing-function: Hàm thời gian xác định cách chuyển động diễn ra (ví dụ: ease, linear, ease-in-out,...).

delay: Thời gian trễ trước khi chuyển động bắt đầu (ví dụ: 0s, 1s,...).

- CSS3 Multiple Columns

CSS3 Multiple Columns cho phép bạn chia nội dung của một phần tử thành nhiều cột, tạo ra bố cục đa cột mà không cần sử dụng các phần tử con hoặc float. Dưới đây là một số thuộc tính và cú pháp khi sử dụng CSS3 Multiple Columns:

1. Cú Pháp Cơ Bản:

column-count: Xác định số lượng cột.

column-gap: Xác định khoảng cách giữa các cột.

column-rule: Đặt rèn giữa các cột (kích thước, kiểu, màu).

2. Column Width:

column-width: Xác định chiều rộng của mỗi cột.

3. Column Span:

column-span: Cho phép một phần tử chấp nhận nhiều cột.

4. Column Fill:

column-fill: Xác định cách phân phối nội dung khi có sự chênh lệch chiều cao giữa các cột (auto hoặc balance).

5. Column Rule Style, Width, Color:

column-rule-style: Xác định kiểu của rèn.

column-rule-width: Xác định kích thước của rèn.

column-rule-color: Xác định màu sắc của rèn.

6. Column Break Before/After:

break-before: Xác định cách thức ngắt trước phần tử.

break-after: Xác định cách thức ngắt sau phần tử.

7. Column Rule Shorthand:

column-rule: Sử dụng shorthand để thiết lập style, width, và color của rèn.

8. Column Count Auto:

Nếu sử dụng auto cho column-count, trình duyệt sẽ cố gắng tối ưu số lượng cột dựa trên kích thước của phần tử và các thuộc tính khác.

9. Responsive Columns:

Bạn có thể sử dụng media queries để thay đổi số lượng cột theo kích thước màn hình.

CSS3 Multiple Columns là một cách tiện lợi để tạo ra bố cục đa cột mà không cần sử dụng float hay phần tử con. Điều này giúp tối ưu hóa không gian và làm cho nội dung trở nên linh hoạt và dễ đọc.

- CSS3 Animations

CSS3 Animations là một tính năng mạnh mẽ cho phép bạn tạo ra các hiệu ứng chuyển động và hoạt hình trên trang web mà không cần sử dụng JavaScript hoặc các thư viện bên ngoài. Dưới đây là một số điểm chính và cú pháp khi sử dụng CSS3 Animations:

**3.12. SCSS**

SCSS (Sassy CSS) là một mô phỏng của ngôn ngữ CSS3, giúp tăng cường tính linh hoạt và tiện lợi trong quá trình viết mã CSS. SCSS là một phần của Sass (Syntactically Awesome Stylesheets), và nó là một biến thể của ngôn ngữ Sass cũ hơn.

Dưới đây là một số điểm quan trọng về SCSS:

1. Cú Pháp:

SCSS sử dụng cú pháp giống CSS, với việc sử dụng dấu ngoặc nhọn {} để nhóm các quy tắc CSS, và dấu chấm phẩy ; để kết thúc mỗi quy tắc.

2. Biến:

SCSS hỗ trợ sử dụng biến để lưu giữ giá trị và tái sử dụng chúng trong mã nguồn.

3. Nesting:

SCSS cho phép bạn lồng các quy tắc CSS bên trong nhau, tạo mã nguồn có cấu trúc hơn.

4. Mixin:

Mixin là một khối mã có thể được tái sử dụng trong nhiều định nghĩa CSS khác nhau.

5. Import:

SCSS cho phép bạn import nội dung của một file SCSS khác vào một file SCSS khác.

6. Extend:

Extend cho phép bạn kế thừa các quy tắc từ một selector khác.

7. Operators:

SCSS hỗ trợ các toán tử, giúp thực hiện các phép toán trong mã nguồn CSS.

8. Control Directives:

SCSS hỗ trợ các chỉ thị kiểm soát như @if, @for, @each, @while để quản lý luồng kiểm soát trong mã nguồn.

9. SCSS và Sass:

SCSS và Sass đều là hai ngôn ngữ biên dịch thành CSS, nhưng SCSS sử dụng cú pháp giống CSS, trong khi Sass sử dụng cú pháp không giống.

SCSS được chọn nhiều hơn bởi vì nó giữ nguyên cú pháp của CSS, giúp người mới sử dụng dễ dàng chuyển từ CSS sang SCSS mà không gặp nhiều khó khăn.

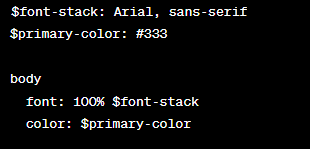
**3.13. SASS**

Sass (Syntactically Awesome Stylesheets) là một ngôn ngữ mở rộng của CSS, mang lại nhiều tính năng và tiện ích giúp làm cho mã nguồn CSS trở nên linh hoạt và dễ quản lý hơn. Sass có hai cú pháp chính: Sass (cú pháp cũ không giống với CSS) và SCSS (Sassy CSS, cú pháp giống với CSS).

Dưới đây là một số điểm quan trọng về Sass:

1. Cú Pháp Cơ Bản:

Sass:



Hình 102:

2. Biến:

Sass hỗ trợ việc sử dụng biến để lưu giữ giá trị và tái sử dụng chúng trong mã nguồn.

3. Nesting:

Sass cho phép lồng các quy tắc CSS bên trong nhau để tạo cấu trúc mã nguồn rõ ràng hơn.

4. Mixin:

Mixin là một khối mã có thể tái sử dụng trong nhiều định nghĩa CSS khác nhau.

5. Extend:

Extend cho phép bạn kế thừa các quy tắc từ một selector khác.

6. Import:

Sass cho phép bạn import nội dung của một file Sass khác vào một file Sass khác.

7. Operators:

Sass hỗ trợ các toán tử, giúp thực hiện các phép toán trong mã nguồn CSS.

8. Control Directives:

Sass hỗ trợ các chỉ thị kiểm soát như @if, @for, @each, @while để quản lý luồng kiểm soát trong mã nguồn.

9. Sass và SCSS:

Sass và SCSS đều được biên dịch thành CSS, nhưng SCSS sử dụng cú pháp giống CSS, trong khi Sass sử dụng cú pháp không giống.

Sass có nhiều tính năng hữu ích để giúp tăng cường quản lý mã nguồn CSS và làm cho quá trình phát triển trang web trở nên hiệu quả và dễ dàng bảo trì.

**KẾT LUẬN CHƯƠNG 3**

CSS (Cascading Style Sheets) và CSS3 đều là ngôn ngữ mô tả kiểu cho trang web, được sử dụng để thiết kế và định dạng giao diện người dùng. Dưới đây là một kết luận về cả hai:

CSS:

Cú Pháp:

CSS sử dụng cú pháp cơ bản với các quy tắc và thuộc tính được đặt trong cặp dấu ngoặc nhọn {}.

Mỗi quy tắc gồm một selector và một tập hợp các thuộc tính giá trị.

Chức Năng Cơ Bản:

CSS được sử dụng để định dạng và trang trí các phần tử HTML trên trang web.

Nó cung cấp kiểu, màu sắc, kích thước và căn chỉnh vị trí của các phần tử.

Float và Position:

CSS sử dụng float và position để kiểm soát vị trí và luồng của các phần tử trong trang.

Grid Layout:

CSS không hỗ trợ lưới định vị phức tạp và phải dựa vào các kỹ thuật như float và flexbox để tạo bố cục.

CSS3:

Cải Tiến Cú Pháp và Chức Năng:

CSS3 là phiên bản cải tiến của CSS, đưa vào nhiều tính năng mới và cú pháp tiện ích hơn.

Nó bao gồm các module như Flexbox, Grid, Transitions, Animations, và nhiều tính năng khác.

CSS3 cung cấp các tính năng như media queries để hỗ trợ thiết kế đáp ứng, tối ưu hóa trải nghiệm người dùng trên nhiều thiết bị.

Hiệu Ứng và Hoạt Hình:

CSS3 mang lại khả năng tạo hiệu ứng và hoạt hình mà không cần sử dụng JavaScript, như transitions và animations.

Border Radius và Box Shadow:

CSS3 cung cấp các thuộc tính như border-radius và box-shadow để tạo các hiệu ứng trang trí cho phần tử.

Transforms và Transitions:

CSS3 hỗ trợ transforms để thực hiện các biến đổi 2D và 3D trên phần tử, cũng như transitions để tạo hiệu ứng chuyển động mượt mà.

Columns và Multiple Backgrounds:

CSS3 hỗ trợ tạo nhiều cột với thuộc tính column và multiple backgrounds cho phép sử dụng nhiều hình ảnh nền.

Tóm lại, CSS3 là một phiên bản mạnh mẽ và tiên tiến hơn của CSS, mang lại nhiều tính năng và khả năng thiết kế mới, giúp tạo ra trải nghiệm người dùng đa dạng và thú vị trên web.

# **CHƯƠNG 4: THIẾT KẾ WEBISTE THEO ĐỀ TÀI BẠN CHỌN**

## **4.1. Ý tưởng của Website**

Giao diện dễ sử dụng: Website cần có một giao diện thân thiện với người dùng, dễ dàng để họ tìm kiếm sản phẩm, xem thông tin chi tiết và thực hiện thanh toán một cách thuận tiện.

Danh sách sản phẩm đa dạng: Website nên cung cấp một loạt các sản phẩm điện thoại từ các nhà sản xuất phổ biến như Apple, Samsung, Xiaomi, OnePlus, và Huawei. Đa dạng các mẫu mã, tính năng, và mức giá để phù hợp với nhu cầu của đa dạng khách hàng.

Thông tin sản phẩm chi tiết: Mỗi sản phẩm cần được mô tả chi tiết, bao gồm hình ảnh, thông số kỹ thuật, tính năng nổi bật, đánh giá từ người dùng (nếu có), và các thông tin liên quan khác.

Tính năng tìm kiếm và sắp xếp: Người dùng cần có thể tìm kiếm sản phẩm dựa trên các tiêu chí như thương hiệu, mức giá, kích thước màn hình, dung lượng pin, và các tính năng khác. Sắp xếp kết quả tìm kiếm theo độ phổ biến, giá cả, hoặc đánh giá từ người dùng.

Giỏ hàng và thanh toán thuận tiện: Người dùng cần có thể thêm sản phẩm vào giỏ hàng, xem lại đơn hàng và thanh toán một cách dễ dàng và an toàn thông qua các phương thức thanh toán trực tuyến như thẻ tín dụng, chuyển khoản ngân hàng, hoặc ví điện tử.

Đánh giá và đánh giá từ người dùng: Tính năng này giúp người dùng đưa ra ý kiến về sản phẩm sau khi mua hàng, từ đó cung cấp thông tin hữu ích cho người khác khi quyết định mua sản phẩm.

Chăm sóc khách hàng: Website cần có một hệ thống hỗ trợ khách hàng thông qua email, tin nhắn trực tuyến, hoặc số điện thoại để giải đáp thắc mắc và xử lý các vấn đề phát sinh từ quá trình mua sắm.

Chính sách đổi trả và bảo hành rõ ràng: Người dùng cần biết được điều kiện và quy trình đổi trả, bảo hành của sản phẩm để cảm thấy yên tâm khi mua hàng trực tuyến.

Tối ưu hóa cho thiết bị di động: Với sự phổ biến của thiết bị di động, website cần được tối ưu hóa để hiển thị một cách tốt nhất trên các thiết bị nhỏ như điện thoại thông minh và máy tính bảng.

## **4.2. Xây dựng bố cục của trang Web**

Header (Đầu trang):

Logo của trang web và tên thương hiệu.

Menu điều hướng cho các danh mục sản phẩm như Điện thoại Di động, Phụ kiện, Tin tức, Liên hệ.

Ô tìm kiếm để người dùng có thể tìm kiếm sản phẩm một cách thuận tiện.

Banner Quảng cáo:

Banner lớn hoặc slide ảnh hiển thị các chương trình khuyến mãi, sản phẩm nổi bật hoặc tin tức mới nhất về sản phẩm.

Danh mục sản phẩm:

Hiển thị các danh mục sản phẩm chính, ví dụ: Điện thoại mới nhất, Điện thoại giảm giá, Điện thoại theo hãng sản xuất, Phụ kiện đi kèm.

Sản phẩm Nổi bật:

Hiển thị một số sản phẩm nổi bật, được đánh dấu hoặc giảm giá đặc biệt.

Khuyến mãi và Ưu đãi:

Khu vực dành riêng cho thông tin về các chương trình khuyến mãi, ưu đãi đặc biệt.

Phần Danh sách Sản phẩm:

Hiển thị danh sách các sản phẩm, mỗi sản phẩm đi kèm với hình ảnh, tên, giá cả và nút "Thêm vào giỏ hàng" hoặc "Mua ngay".

Bộ lọc và Sắp xếp:

Cho phép người dùng lọc sản phẩm theo các tiêu chí như thương hiệu, giá cả, mức giảm giá, đánh giá của người dùng.

Chi tiết sản phẩm:

Trang chi tiết sản phẩm nên hiển thị thông tin chi tiết, mô tả, hình ảnh lớn của sản phẩm, tính năng, đánh giá và phản hồi từ người dùng.

Giỏ hàng:

Hiển thị tổng số sản phẩm trong giỏ hàng và tổng số tiền.

Nút chuyển hướng đến trang thanh toán.

Footer (Chân trang):

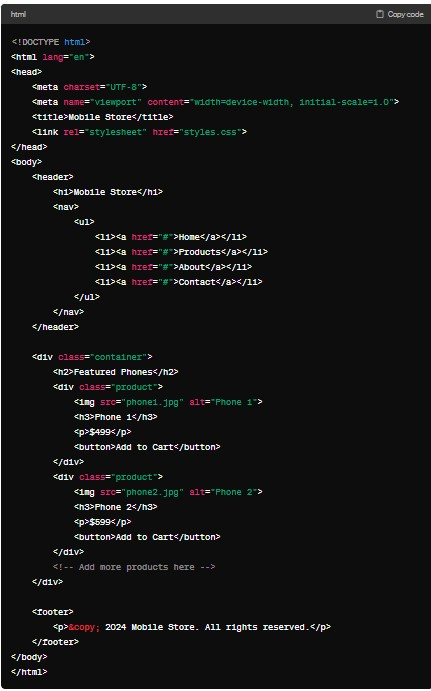
Liên kết đến các trang quan trọng như Trang chủ, Danh mục sản phẩm, Về chúng tôi, Chính sách bảo mật, Chính sách đổi trả.

Thông tin liên hệ như địa chỉ, số điện thoại, email.

Biểu tượng liên kết mạng xã hội.

## **4.3. Thiết kế trang Web bằng HTML và CSS**

html



css



## **4.4. Kết quả đạt được**

Kết quả đạt được là biết tạo một trang wed riêng cho mình và xử dụng thành thạo html và css . Biết cách chỉnh sửa các wed Biết các xử dụng các thẻ trong html như h1,h2,… và xử dụng thành thạo các thẻ html.

**Kết luận chương 4**

Qua chương 4 cho em hiểu biết thêm về lập trình wed html va css.Giúp em biết dùng các thẻ html và tạo ra được một trang wed hoàn chỉnh .

# **KẾT LUẬN**

Giao diện Thân Thiện và Hấp Dẫn: Thiết kế website có giao diện sáng sủa, dễ nhìn và dễ sử dụng, thu hút người dùng ngay từ cái nhìn đầu tiên. Tính Năng Tìm Kiếm Tốt: Hệ thống tìm kiếm được tích hợp một cách hiệu quả, giúp người dùng dễ dàng tìm kiếm sản phẩm mong muốn trong danh mục đa dạng của cửa hàng. Trải Nghiệm Người Dùng Thuận Tiện: Quy trình mua hàng được tối ưu hóa, giúp người dùng dễ dàng thêm sản phẩm vào giỏ hàng, xem lại đơn hàng và tiến hành thanh toán một cách nhanh chóng và thuận tiện. Responsive Design: Website được thiết kế để phản ánh tốt trên các thiết bị di động, máy tính bảng và desktop, đảm bảo rằng người dùng có trải nghiệm mua sắm mượt mà trên mọi nền tảng. Hiển Thị Sản Phẩm Rõ Ràng: Các sản phẩm được hiển thị một cách rõ ràng và hấp dẫn, với hình ảnh chất lượng cao và thông tin sản phẩm chi tiết. Tính Năng Bình Luận và Đánh Giá: Website có tính năng cho phép người dùng đăng bình luận và đánh giá về sản phẩm, giúp người mua có cái nhìn toàn diện về chất lượng và độ tin cậy của sản phẩm. Tối Ưu Hóa Tốc Độ: Trang web được tối ưu hóa để tải nhanh, giảm thời gian chờ đợi và tăng khả năng chuyển đổi. Bảo Mật Thông Tin: Các biện pháp bảo mật được triển khai để bảo vệ thông tin cá nhân và thanh toán của người dùng, tạo ra một môi trường mua sắm an toàn và tin cậy.

Giao diện đẹp,thân thiện vớ ingười dùng,có nhiều chức năng và tương tác tốt hơn với người dùng

Thiết kế,bổ sung các tính năng còn thiếu hay chưa thật sự hoàn chỉnh trong quá trình sử dụng.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.
2. Trang wed thegioididong.
3. Trang wed FPTshop.