**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 TRANG WEB BÁN HÀNG THỜI TRANG**

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

**Sinh viên thực hiện:**  **Hà Thị Phương Thanh**

**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 TRANG WEB BÁN HÀNG THỜI TRANG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020626 | Hà Thị Phương Thanh | 3/6/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

Lời nói đầu về Thiết kế Web và Triển khai Hệ thống Phần mềm có thể bắt đầu bằng việc nhấn mạnh tầm quan trọng của hai lĩnh vực này trong ngữ cảnh ngày nay, khi mà công nghệ ngày càng đóng vai trò quan trọng trong mọi lĩnh vực đời sống và kinh doanh. Thiết kế Web và Triển khai Hệ thống Phần mềm đóng vai trò quyết định trong việc tạo ra trải nghiệm người dùng hiệu quả và đảm bảo hoạt động ổn định của các ứng dụng và hệ thống.

Thiết Kế Web:

Thiết kế Web không chỉ là về việc tạo ra giao diện đẹp mắt mà còn liên quan mật thiết đến trải nghiệm người dùng. Một giao diện người dùng thân thiện, dễ sử dụng và tương thích với nhiều thiết bị không chỉ là yếu tố quyết định thành công mà còn ảnh hưởng trực tiếp đến sự hài lòng của người dùng. Thiết kế Web cũng đòi hỏi sự tích hợp đúng đắn của các yếu tố như tốc độ tải trang, khả năng tương tác, và tích hợp an toàn. Qua đó, chúng ta có thể tạo ra những trang web không chỉ đẹp mắt mà còn mang lại giá trị thực sự cho người sử dụng.

Triển Khai Hệ Thống Phần Mềm:

Triển khai Hệ thống Phần mềm là quá trình chuyển đổi ý tưởng từ một khả năng trên giấy thành một thực tế hoạt động. Việc triển khai đòi hỏi sự chính xác, kiểm soát và quản lý chặt chẽ để đảm bảo rằng hệ thống sẽ hoạt động đúng cách và đáp ứng đúng nhu cầu. Tính ổn định, bảo mật và khả năng mở rộng là những yếu tố không thể thiếu trong quá trình triển khai. Đồng thời, việc đào tạo người sử dụng cũng là một phần quan trọng để họ có thể tận dụng tối đa các tính năng của hệ thống.

Tầm Quan Trọng:

Tầm quan trọng của việc kết hợp chặt chẽ giữa Thiết kế Web và Triển khai Hệ thống Phần Mềm không chỉ là về việc tạo ra sản phẩm hoạt động hiệu quả mà còn đảm bảo sự hài lòng của người sử dụng và sự thành công của dự án. Sự hiểu biết sâu sắc về nhu cầu của người dùng, khả năng thích ứng với xu hướng công nghệ mới và kỹ năng quản lý dự án là những yếu tố quyết định sự thành công trong lĩnh vực này.

Trong lời nói đầu này, chúng ta hy vọng sẽ khám phá và thấu hiểu sâu hơn về những thách thức và cơ hội trong việc Thiết kế Web và Triển khai Hệ thống Phần Mềm, đồng thời tìm kiếm sự sáng tạo và hiệu suất để đáp ứng những yêu cầu ngày càng cao trong thế giới kỹ thuật số ngày nay.

**Mục lục**

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ**](#_Toc162170260)[**THỐNG PHẦN MỀM 3**](#_Toc162170261)

[1.1. Các khái niệm cơ bản 3](#_Toc162170262)

[1.2. Phân biệt Web tĩnh, Web động 3](#_Toc162170263)

[1.3. Một số thuật ngữ. 5](#_Toc162170265)

[1.4. Một số công cụ dùng thiết kế Web 8](#_Toc162170266)

[**KẾT LUẬN CHƯƠNG 1 9**](#_Toc162170267)

[**CHƯƠNG II: HTML & HTML5 11**](#_Toc162170268)

[2.1. Tổng quan về HTML 11](#_Toc162170270)

[2.2. Cấu trúc tổng quát trang HTML 12](#_Toc162170271)

[2.4. Các thẻ tạo biểu mẫu (form) 16](#_Toc162170272)

[2.5. Một số thẻ HTML đặc biệt 18](#_Toc162170273)

[2.6. HTML5 20](#_Toc162170274)

[**KẾT LUẬN CHƯƠNG 2 27**](#_Toc162170275)

[**CHƯƠNG III: CSS và CSS3 29**](#_Toc162170276)

[3.1. CSS là gì? 29](#_Toc162170278)

[3.2. Cú pháp CSS 30](#_Toc162170279)

[3.3. Áp dụng CSS vào trang HTML 31](#_Toc162170280)

[3.4. Selectors 32](#_Toc162170281)

[3.4.1.Universal selector . 35](#_Toc162170282)

[3.4.2. Type selector 35](#_Toc162170283)

[3.4.3. Identity selector 35](#_Toc162170284)

[3.4.4. Class selector 36](#_Toc162170285)

[3.4.5. Descendant selector 36](#_Toc162170286)

[3.4.6. Child selector 36](#_Toc162170287)

[3.4.7.Adjacent sibling selector 37](#_Toc162170288)

[3.4.8. Attribute selector 37](#_Toc162170289)

[3.4.9.Pseudo class selector 38](#_Toc162170290)

[3.4.10.Group selector 38](#_Toc162170291)

[3.5. Đơn vị đo lường CSS 38](#_Toc162170292)

[3.6. Kế thừa thuộc tính 39](#_Toc162170293)

[3.7. Các nhóm thuộc tính trong CSS 40](#_Toc162170294)

[3.8. Float &amp; Clear 47](#_Toc162170295)

[3.9. Flex 47](#_Toc162170296)

[3.10. Grid 48](#_Toc162170297)

[3.11.CSS3 . 49](#_Toc162170298)

[3.12. SCSS 61](#_Toc162170299)

[3.13. SASS 62](#_Toc162170300)

[**KẾT LUẬN CHƯƠNG 3 64**](#_Toc162170301)

[**CHƯƠNG 4: THIẾT KẾ TRANG WEB BÁN HÀNG THỜI TRANG 66**](#_Toc162170302)

[4.1. Ý tưởng của Website 66](#_Toc162170303)

[4.2. Xây dựng bố cục của trang Web 66](#_Toc162170304)

[4.3. Thiết kế trang Web bằng HTML và CSS 81](#_Toc162170305)

[4.4. Kết quả đạt được 84](#_Toc162170306)

[**Kết luận 87**](#_Toc162170307)

[**DANH MỤC TÀI LIỆU THAM KHẢO 88**](#_Toc162170308)

# **CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ**

# **THỐNG PHẦN MỀM**

### 1.1. Các khái niệm cơ bản

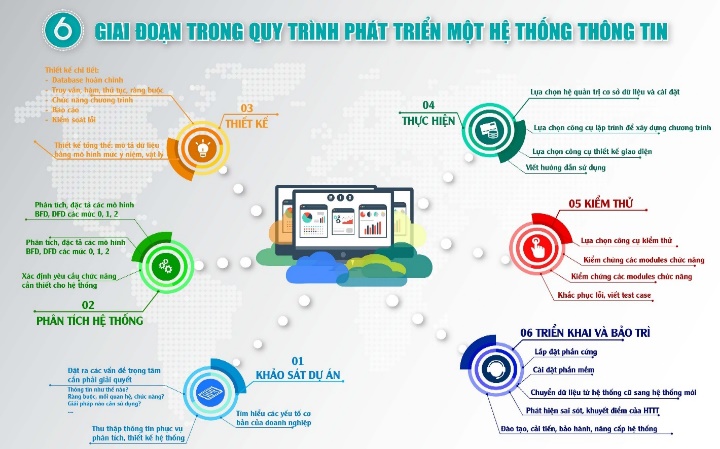
Thiết kế web là quá trình bao gồm tạo ra giao diện và trải nhiệm người dùng cho một trang web. Nó bao gồm việc lập kế hoạch, tạo bố cục, chọn màu sắc, font chữ, hình ảnh và xác

định cách người dùng tương tác với trang web. Mục tiêu của cảu thiết kế web là tạo ra một giao diện hấp dẫn, tương thích với nhiều thiết bị và dễ sử dụng.

Triển khai hệ thống phần mềm là quá trình đưa hệ thống phần mềm từ môi trường phát triển sang môi trường hoạt động. Nó bao gồm cấu hình phần mềm, cài đặt, kiểm tra và triển khai

đặc điểm riêng của hệ thống phần mềm trong môi trường đích. Mục tiêu của triển khai hệ thống phần mềm là :

đảm bảo hệ thống hoạt động đúng như mong đợi, tối ưu hóa hiệu suất và đảm bảo tính bảo mật.

. 

Việc thiết kế web và triển khai hệ thống phần mềm là hai khía cạnh chủ yếu của quá trình phát triển trang web hoặc ứng dụng web. Các kỹ năng liên quan bao gồm kiến thức về các ngôn ngữ lập trình web, giao diện người dùng, quy trình phát triển phần mềm và triển khai hệ thống.

### 1.2. Phân biệt Web tĩnh, Web động

### 

*a. Web tĩnh*

Đặc điểm chính:

Trang web tĩnh là những trang web mà nội dung của chúng được tạo ra trước và không thay đổi dựa trên hành động của người dùng.

Nội dung của trang web tĩnh thường được lưu trữ trong các tệp HTML, CSS và JavaScript cố định.

Ưu điểm:

Hiệu suất cao vì nội dung được tải nhanh và không đòi hỏi xử lý phức tạp từ phía máy chủ.

Dễ cache, giảm tải cho máy chủ và tăng tốc độ trang web.

Thiết kế đồ họa đẹp web tĩnh thường được trình bày ấn tượng và cuốn hút hơn web động về phần mỹ thuật đồ họa vì chúng ta có thể hòa toàn tự do trình bày các ý tưởng về đồ họa và mỹ thuật trên toàn diện tích từng trang web tĩnh.

Tốc độ truy cập của người dùng vào các trang web tĩnh nhanh hơn cá trang web động vì không mất thời gian trong việc truy vấn cơ sở dữ liệu như web động.

Nhược điểm:

Yêu cầu xử lý máy chủ để tạo và đưa ra nội dung, có thể làm tăng tải cho máy chủ.

Thường yêu cầu kỹ năng lập trình và quản lý cơ sở dữ liệu.

Khó khăn trong việc thay đổi và cập nhật thông tin muốn thay đổi và cập nhật nội dung thông tin của trang website tĩnh Bạn cần phải biết về ngôn ngữ html, sử dụng được các chương trình thiết kế đồ hoạ và thiết kế web cũng như các chương trình cập nhật file lên server.Thông tin không có tính linh hoạt, không thân thiện với người dùng: Do nội dung trên trang web tĩnh được thiết kế cố định nên khi nhu cầu về thông tin của người truy cập tăng cao thì thông tin trên website tĩnh sẽ không đáp ứng được.Khó tích hợp, nâng cấp, mở rộng: Khi muốn mở rộng, nâng cấp một website tĩnh hầu như là phải làm mới lại website.

*b. Web động*

. Đặc điểm:

Có khả năng thay đổi nội dung và chức năng dựa trên hành động của người dùng.Thường được sử dụng cho các ứng dụng trang web yêu cầu tương tác cao như các ứng dụng thương mại điện tử, trò chơi trực tuyến.Có tốc độ tải trang web chậm hơn web tĩnh do cần tải lại dữ liệu từ máy chủ khi có sự tương tác từ người dùng.

. Ưu điểm:

Tương tác người dùng cao.

Có khả năng tích hợp dữ liệu động.

Cung cấp trải nhiệm người dùng linh hoạt và đa dạng.

Nhược điểm:

Tốc độ tải trang có thể chậm hơn so với web tĩnh.

Tính tương thích với trình duyệt và SEO có thể sẽ gặp khó khăn.

Phát triển và duy trì phức tạp hơn.

### 1.3. Một số thuật ngữ.

Hosting: là một dịch vụ online giúp bạn xuất bản website hoặc ứng dụng web lên internet. Khi đăng kí dịch vụ Hosting tức là ta đang thuê một chỗ đặt trên sever chứa tất cả các files và dữ liệu cần thiết để website chạy được. Một server là một máy tính vật lý chạy không gián đoạn để website có thể luôn hoạt động mọi lúc cho tất cả mọi người truy cập vào. Nhà cung cấp Web Hosting chịu trách nhiệm cho việc giữ server hoạt động, chống tấn công bởi mã độc, và chuyển nội dung văn bản, hình ảnh, files từ server xuống trình duyệt người dùng.

Tên miền (Domain Name): là tên của một trang web hoạt động trên internet, là địa chỉ IP cho mỗi một website. Tên miền giúp cho người dùng dễ dàng nhận biết và nhớ đến trang web. Có thể hình dung tên miền giống như địa chỉ nhà hay mã zipcode để định vị: nhờ vào đó mà người dùng có thể tìm đến được website. Địa chỉ IP đang được sử dụng hiện tại là IPv4 gồm có 32 bit, chia thành 4 Octet. Các OCtet này được ngăn cách nhau bởi dấu chấm và được biểu hiện ở dưới dạng thập phân có đầy đủ 12 chữ số. Chính vì là dãy số dài như vậy thì người dùng không thể nào nhớ rõ được. Cho nên, tên miền được dùng để thay thế cho những địa chỉ IP, để người dùng cảm thấy dễ nhớ, dễ hiểu hơn.

HomePage: là trang đầu tiên mà người dùng nhìn thấy khi truy cập vào một website chỉ chứa duy nhất tên miền của nó. Không chỉ là một thành phần cơ bản của website, trang chủ còn có vai trò vô cùng quan trọng trong việc tạo nên ấn tượng ban đầu cho người

dùng, thu hút họ khám phá thêm các nội dung khác trên website. Chức năng chính của HomePage là cung cấp cho người dùng thông tin tổng quan về doanh nghiệp, sản phẩm,

dịch vụ của web đó, và điều hướng người dùng đến xem các trang web khác được lồng trong website đó, để có thể gia tăng tỉ lệ chuyển đổi trang chủ thường được được tích

hợp thêm những lời kêu gọi hành động rõ ràng và hấp dẫn nhằm thu hút người dùng thực hiện hành động.

Web Site: là tập hợp các trang chứa thông tin bảo gồm văn bản, hình ảnh, video, dữ liệu, nằm trên một domain, được lưu trữ trên máy chủ web. Website có thể được người dùng

truy cập từ xa thông qua mạng internet. Một trang web tồn tại dưới dạng tập tin HTML hoặc XHTML có thể truy cập bằng giao thức HTTP hoặc HTTPS. Website có thể tự xây

dựng các tập tin HTML ( website tĩnh ) hoặc vận hành với các CMS chạy trên máy chủ ( website động ). Bên cạnh đó, website được xây dựng trên nhiều ngôn ngữ lập trình

khác nhau như PHP, JavaScript, Java. Hiểu đơn giản hơn về khái niệm website là chỉ một site nằm trên một web. Website cho phép người dùng đưa thông tin cá nhân, thông

tin doanh nghiệp hay đăng tải bất kì chủ đề nào để người khác có thể truy cập thông qua internet.

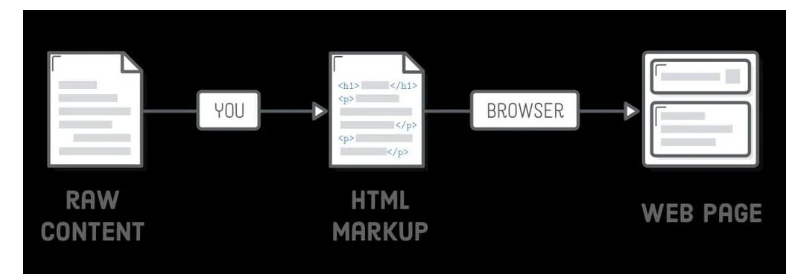
WebServer: là máy chủ cài đặt các chương trình phục vụ cá ứng dụng web. Webserver có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hồi đến client thông qua giao

thức HTTP hoặc các giao thức khác. Có nhiều web server khác nhau như: Apache, Nginx, IIS, …, web server thông dụng nhất hiện nay gồm

|  |
| --- |
| **C:\Users\tmhung\Pictures\Screenshots\Screenshot (105).png** |

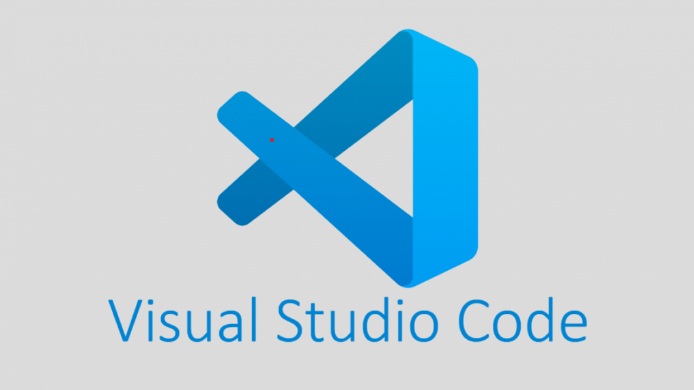
URL-Uniform Resource Location: nghĩa tiếng việt là định vị tài nguyên thống nhất. Cụ thể URL là địa chỉ của một tài nguyên duy nhất trên web. Mỗi URL hợp lệ sẽ trỏ đến một tài nguyên duy nhất tài nguyên đó có thể là trang HTML, tài liệu CSS, hình ảnh, video, file PDF, trong một số trường hợp ngoại lệ, URL có thể trỏ đến những tài nguyên không còn tồn tại hoặc đã bị di chuyển sang địa chỉ khác. URL có ther chứa nhiều thành phần khác nhau. Nó bao gồm hostname ánh tới địa chỉ IP của một tài nguyên cụ thể trên internet và một loạt các thông tin bổ sung thông báo cho quá trình duyệt và máy chủ biết cách sử lý mọi thứ. Có thể nghĩ địa chỉ IP như số điện thoại, hostname như tên của người sở hữu số điện thoại đó mà chúng ta muốn tra cứu. Và một tiêu chuẩn được gọi là hệ thống tên miền hoạt động trong nền giống như một cuốn danh bạ điện thoại, dịch các hostname thành địa chỉ IP để mạng sử dụng để định tuyến lưu lượng truy cập.

Browser: web browser hay trình duyệt web là công cụ dùng để đọc và truy cập vào các trang web bằng đường dẫn URL riêng biệt. Các trang đấy được thiết kế và tạo nên từ các ngôn ngữ HTML với các dòng code, web browser sẽ mã hóa các dòng lệnh ấy cho phép hiển thị giao diện hoàn chỉnh của một trang web.



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

Visual Studio code: là một trình soạn thảo mã nguồn mở gọn nhẹ nhưng có khả năng vận hành mạnh mẽ trên 3 nền tảng là Windows, Linux và macOS được phát triển bởi microsoft. Nó hỗ trợ cho JaveScript, Node.js và TypeScript, cũng như cung cấp một hệ sinh thái mở rộng vô cùng phong phú cho nhiều ngôn ngữ lập trình khác. Có thể nói, VS Code là sự kết hợp giữa một trình soạn thảo mã nguồn với những công cụ phát triển mạnh mẽ như Git, Debug hay Syntax Highlighter.



Adobe Dreamweaver CC là một thiết kế web và ứng dụng Môi trường phát triển tích hợp (IDE) được sử dụng để phát triển và thiết kế trang web. Dreamweaver bao gồm một trình hỗ trợ mã hóa hỗ trợ soạn thảo cú pháp, hoàn thành mã hóa, kiểm tra thực tế thời gian cú pháp và nội dung hướng dẫn để tạo mẹo mã để hỗ trợ người dùng viết mã.



# **KẾT LUẬN CHƯƠNG 1**

Thiết kế Web và triển khai hệ thống phần mềm đều là các khía cạnh quan trọng trong quá trình phát triển ứng dụng web và dịch vụ trực tuyến. Dưới đây là một số điểm kết luận:

a)Thiết Kế Web

Là công việc của một Web Designer (Chuyên viên Thiết kế web) có nhiệm vụ tạo ra bộ mặt hay còn gọi là Giao diện (Template) website một cách hoàn chỉnh. Giao diện này có thể ở dạng Ảnh hoặc dạng Web Tĩnh HTML.

Sự chọn lựa của ngôn ngữ lập trình và công cụ thiết kế web quyết định độ linh hoạt và hiệu suất của trang web. Các công cụ như Visual Studio Code, Adobe Dreamweaver, và Sublime Text giúp tăng cường quá trình phát triển.

Responsive design là yếu tố quan trọng để đảm bảo trang web hiển thị đẹp trên nhiều thiết bị. UX/UI được tối ưu hóa để cung cấp trải nghiệm người dùng tốt nhất.

Sự tương tác mượt mà và hiệu suất của trang web phụ thuộc vào sự hài hòa giữa front-end (giao diện người dùng) và back-end (xử lý dữ liệu và logic) development.

Bảo mật là ưu tiên hàng đầu. Tương tác người dùng thông qua các yếu tố như biểu mẫu và hộp thoại cũng cần được xem xét để cải thiện trải nghiệm người dùng.

b)Triển Khai Hệ Thống Phần Mềm:

Lựa chọn hosting và server ảnh hưởng đến hiệu suất và sẵn sàng của ứng dụng. Các dịch vụ như Amazon Web Services (AWS) và Microsoft Azure cung cấp các giải pháp mạnh mẽ.

Lựa chọn hệ quản trị cơ sở dữ liệu (DBMS) như MySQL, PostgreSQL hay MongoDB cần được cân nhắc để đảm bảo khả năng mở rộng và hiệu suất tốt.

Sử dụng hệ thống quản lý phiên bản như Git và triển khai liên tục (CI/CD) giúp quản lý mã nguồn và đảm bảo quá trình triển khai được thực hiện một cách an toàn và hiệu quả.

Hệ thống cần có khả năng mở rộng để đối mặt với tăng trưởng người dùng. Các biện pháp bảo mật, bao gồm mã hóa và kiểm soát truy cập, là quan trọng để bảo vệ dữ liệu.

Tổng Quan:

Thiết kế web và triển khai hệ thống phần mềm không chỉ là quá trình kỹ thuật mà còn liên quan đến trải nghiệm người dùng và hiệu suất hệ thống. Đảm bảo sự hài hòa giữa chúng là quan trọng để xây dựng và duy trì các ứng dụng web hiệu quả và bền vững trong thời gian. Sự cân nhắc kỹ lưỡng và kiến thức đa ngành là chìa khóa để đạt được mục tiêu này

# **CHƯƠNG II: HTML & HTML5**

# **(Hyper Text Markup Language)**

### 2.1. Tổng quan về HTML

HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu chủ yếu được sử dụng để xây dựng cấu trúc và hiển thị nội dung trên trang web. Được phát triển bởi Tim Berners-Lee vào những năm 1990, HTML là một thành phần chính của công nghệ World Wide Web và chủ yếu được duy trì và phát triển bởi Tổ chức World Wide Web (W3C).

Cơ Bản về HTML:

Thẻ (Tag):

HTML sử dụng các thẻ để đánh dấu các phần khác nhau của trang web. Mỗi thẻ bắt đầu bằng < và kết thúc bằng >. Thẻ thường đi cặp, bao gồm thẻ mở <tag> và thẻ đóng </tag>. Ví dụ: <p>Đây là một đoạn văn.</p>.

Phần Tử (Element):

Một phần tử HTML bao gồm cặp thẻ mở và thẻ đóng cùng với nội dung bên trong. Ví dụ: <p>Đây là một đoạn văn bản.</p>.

Thuộc Tính (Attribute):

Thuộc tính cung cấp thông tin bổ sung về một phần tử và thường xuất hiện trong thẻ mở. Ví dụ: <a href="https://www.example.com">Đây là một liên kết</a>.

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

Cấu trúc tổng quát của một trang HTML bao gồm nhiều phần quan trọng, mỗi phần đóng vai trò khác nhau trong việc mô tả và hiển thị nội dung trên trình duyệt. Dưới đây là một mô tả chi tiết về cấu trúc này:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

</html>

**2.3. Các thẻ HTML thông dụng**

- Các thẻ xử lý văn bản

Các thẻ xử lý văn bản trong HTML là các thẻ được sử dụng để định dạng và hiển thị văn bản trên trang web. Dưới đây là một số thẻ phổ biến:

<h1> to <h6> - Tiêu đề:

Được sử dụng để định nghĩa các tiêu đề với kích thước giảm dần từ lớn nhất <h1> đến nhỏ nhất <h6>.

<p> - Đoạn văn bản:

Sử dụng để định nghĩa đoạn văn bản

<strong> và <em> - In đậm và nghiêng:

strong> được sử dụng để làm đậm văn bản, còn <em> để làm nghiêng

<u> - Gạch chân:

Sử dụng để gạch chân văn bản. Tuy nhiên, nên tránh sử dụng nó để chỉ định liên kết vì đối với người dùng, gạch chân thường được kết hợp với liên kết

<br> - Xuống dòng:

Sử dụng để tạo một dòng mới trong văn bản

<blockquote> - Trích dẫn đoạn văn:

Được sử dụng để bao bọc đoạn văn được trích dẫn từ nguồn ngoạ

<sup> và <sub> - Chỉ số trên và dưới:

<sup> được sử dụng để hiển thị chỉ số trên, còn <sub> để chỉ số dưới

<hr> - Đường kẻ ngang:

Sử dụng để chèn một đường kẻ ngang để phân tách nội dung.

-Các thẻ tạo bảng  
 HTML cung cấp một số thẻ để tạo và định dạng bảng trên trang web. Dưới đây là một số thẻ phổ biến được sử dụng để tạo bảng:

<table> - Bảng:

Thẻ <table> được sử dụng để bắt đầu và kết thúc một bảng.

<tr> - Hàng (row):

Thẻ <tr> được sử dụng để xác định một hàng trong bảng.

<th> - Ô tiêu đề (header cell):

Thẻ <th> được sử dụng để xác định một ô tiêu đề trong bảng.

<td> - Ô dữ liệu (data cell):

Thẻ <td> được sử dụng để xác định một ô chứa dữ liệu trong bảng.

<thead>, <tbody>, <tfoot> - Phần đầu, phần thân và phần chân của bảng:

<thead> được sử dụng để nhóm các ô tiêu đề trong bảng.

<tbody> được sử dụng để nhóm các ô dữ liệu trong bảng.

<tfoot> được sử dụng để nhóm các ô chân bảng (footer).

<caption> - Tiêu đề bảng:

Thẻ <caption> được sử dụng để thêm một tiêu đề cho bảng.

Với cú pháp trên, bạn có thể tùy chỉnh và xây dựng các bảng phức tạp hơn tùy thuộc vào yêu cầu của trang web.

- Thẻ liên kết

Thẻ liên kết (<a>) trong HTML được sử dụng để tạo hyperlinks, cho phép người dùng chuyển đến một trang web khác, tải xuống một tệp, hoặc thực hiện các hành động khác trên trang web:

 <a href="áo phông.html >

<img  src="logo.jpg" alt="Logo">

</a>

Thẻ mục tiêu (target) trong thẻ liên kết:

blank: Mở liên kết trong cửa sổ hoặc tab mới.

self: Mở liên kết trong cùng cửa sổ hoặc tab.

parent: Mở liên kết trong cửa sổ cha của nó (đối với các cửa sổ iframe).

top: Mở liên kết trong cửa sổ cha nhất, bỏ qua các cửa sổ iframe.

- Các thẻ đa phương tiện

Các thẻ đa phương tiện trong HTML được sử dụng để tích hợp và hiển thị nhiều loại nội dung đa phương tiện như hình ảnh, video, âm thanh và đối tượng nhúng trực tiếp vào trang web. Dưới đây là một số thẻ và thuộc tính phổ biến liên quan đến đa phương tiện:

Thẻ <img> - Hình Ảnh:

<img  src="logo.jpg" alt="Logo">

src: Đường dẫn đến tệp hình ảnh.

alt: Mô tả hình ảnh, hiển thị khi hình ảnh không thể tải.

Thẻ <audio> - Âm Thanh:

 <audio src="audio/happy.mp3" controls ></audio>

controls: Hiển thị các điều khiển phát như nút play, stop.

<source>: Chứa nguồn dữ liệu âm thanh và kiểu MIME.

Thẻ <video> - Video:



width, height: Kích thước của video.

controls: Hiển thị các điều khiển phát như nút play, stop.

<source>: Chứa nguồn dữ liệu video và kiểu MIME.

Thẻ <iframe> - Nhúng Trang Web hoặc Nội Dung:

<iframe src="https://www.youtube.com//embed/FSeGrBw5eFA" width="900px" height="500px" frameborder="0"></iframe>

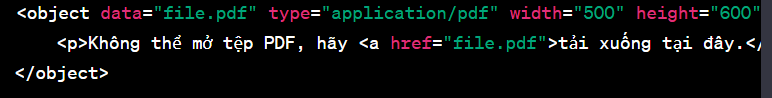
src: Đường dẫn đến trang web hoặc nội dung được nhúng.

width, height: Kích thước của iframe.

frameborder: Thiết lập viền xung quanh iframe.

allowfullscreen: Cho phép chế độ toàn màn hình.

Thẻ <object> - Đối Tượng Nhúng:



Hình 7:

data: Đường dẫn đến tệp đối tượng.

type: Kiểu MIME của đối tượng.

width, height: Kích thước của đối tượng.

Thẻ <embed> - Nhúng Nội Dung:



Hình 8:

src: Đường dẫn đến tệp nhúng.

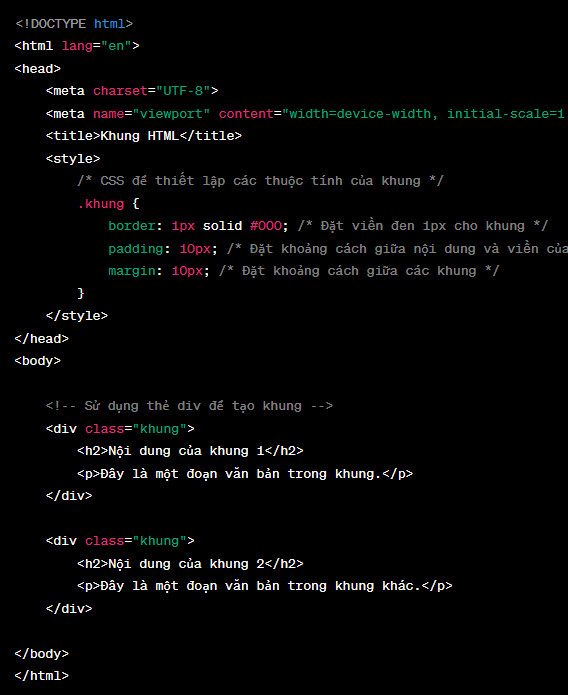
type: Kiểu MIME của nội dung nhúng.

width, height: Kích thước của nội dung.

- Thẻ tạo khung

Trong HTML, để tạo khung cho nội dung của bạn, bạn có thể sử dụng thẻ <div> (tức là division hoặc phân khúc). Thẻ <div> không có ý nghĩa đặc biệt về mặt nội dung, nhưng nó thường được sử dụng để nhóm các phần tử HTML lại với nhau và tạo cấu trúc cho trang web.

Dưới đây là một ví dụ cơ bản về cách sử dụng thẻ <div> để tạo khung:

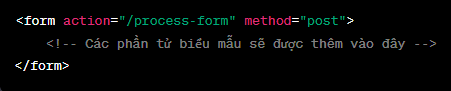


Hình 9:

**2.4. Các thẻ tạo biểu mẫu (form)**

Để tạo biểu mẫu (form) trong HTML, bạn sử dụng một loạt các thẻ và thuộc tính. Dưới đây là một số thẻ quan trọng liên quan đến việc tạo biểu mẫu:

<form>: Thẻ này định nghĩa một biểu mẫu HTML và chứa các phần tử biểu mẫu khác bên trong nó.

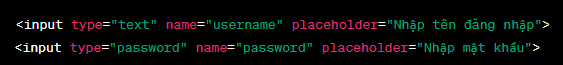


Hình 10:

action: Đường dẫn mà biểu mẫu sẽ được gửi đến khi người dùng nhấn nút gửi.

method: Phương thức gửi dữ liệu, thường là "get" hoặc "post".

<input>: Thẻ này tạo một ô nhập dữ liệu trong biểu mẫu. Có nhiều loại input khác nhau như text, password, checkbox, radio, và nhiều loại khác.



Hình 11:

<label>: Thẻ này định nghĩa một nhãn cho các phần tử biểu mẫu. Nó giúp cải thiện trải nghiệm người dùng và làm cho biểu mẫu có thể được nhập liệu một cách dễ dàng hơn.



Hình 12:

<select> và <option>: Sử dụng chúng để tạo danh sách thả xuống (dropdown).



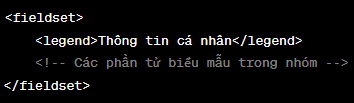
Hình 13:

<textarea>: Được sử dụng để tạo một ô văn bản lớn hơn cho người dùng nhập dữ liệu.



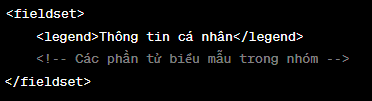
Hình 14:

<button>: Tạo nút trong biểu mẫu, thường được sử dụng để gửi dữ liệu.



Hình 15:

<fieldset> và <legend>: Sử dụng để nhóm các phần tử biểu mẫu và tạo một đầu đề cho nhóm.

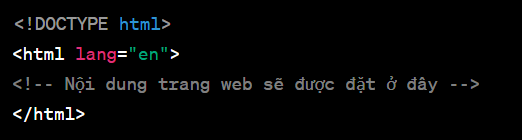


Hình 16:

### 2.5. Một số thẻ HTML đặc biệt

Dưới đây là một số thẻ HTML đặc biệt mà bạn có thể gặp khi phát triển trang web:

<html>: Định nghĩa một tài liệu HTML và chứa toàn bộ nội dung của trang web.



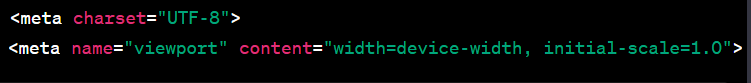
Hình 17:

<head>: Chứa các thành phần không hiển thị trực tiếp trên trang, như các thẻ meta, title, style, script, và nhiều thứ khác.



Hình 18:

Cung cấp thông tin mô tả về trang web, như bộ mã hóa ký tự, thẻ viewport cho responsive design, mô tả trang, v.v.



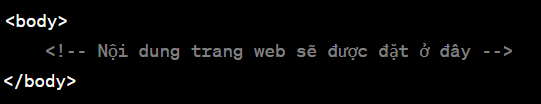
Hình 19:

Đặt tiêu đề của trang web, hiển thị trên thanh tiêu đề của trình duyệt.



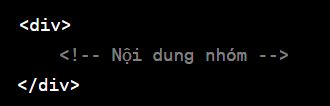
Hình 20:

<body>: Chứa nội dung hiển thị trực tiếp trên trang web.



Hình 21:

<div>: Dùng để nhóm các phần tử HTML và tạo cấu trúc cho trang web.



Hình 22:

<span>: Tương tự như <div>, nhưng được sử dụng để nhóm một phần nhỏ của văn bản hoặc một số phần tử khác.



Hình 23:

<a>: Tạo liên kết (hyperlink) đến một trang web khác hoặc tài nguyên.



Hình 24:

<img>: Nhúng hình ảnh vào trang web.



Hình 25:

<br>: Tạo một dòng mới trong văn bản.



Hình 26:

<hr>: Tạo một đường ngang để phân chia nội dung.



Hình 27:

### 2.6. HTML5

**- HTML5 là gì?**

HTML5 là phiên bản mới nhất và cập nhật của ngôn ngữ đánh dấu siêu văn bản HTML (Hypertext Markup Language), một phần quan trọng của các công nghệ web. HTML5 đã được phát triển để cải thiện khả năng đa phương tiện, tích hợp đa dạng thiết bị, cũng như để hỗ trợ các ứng dụng web phức tạp hơn.

Dưới đây là một số điểm chính về HTML5:

Hỗ trợ đa phương tiện: HTML5 đưa ra nhiều phần tử mới như <audio>, <video>, và <canvas> để hỗ trợ tích hợp âm thanh, video và đồ họa vector một cách thuận tiện hơn, không cần sự phụ thuộc vào các plugin bên ngoài như Flash.

Tích hợp các API mới: HTML5 cung cấp các API tích hợp như Geolocation API, Web Storage API, WebSockets, và nhiều API khác để làm cho việc phát triển ứng dụng web trở nên mạnh mẽ hơn và tương tác với thiết bị và dữ liệu nền.

Tích hợp hỗ trợ đa thiết bị: HTML5 được thiết kế để hỗ trợ trải nghiệm người dùng tốt trên nhiều thiết bị khác nhau, bao gồm cả máy tính, điện thoại di động và máy tính bảng.

Semantics cải tiến: Các phần tử mới như <header>, <nav>, <article>, <footer> giúp cải thiện semantics của trang web, giúp các công cụ tìm kiếm và các trình đọc màn hình hiểu rõ cấu trúc của trang web.

Form mới và cập nhật: HTML5 cung cấp các phần tử và thuộc tính mới để giúp việc xây dựng biểu mẫu trở nên linh hoạt và dễ bảo trì hơn.

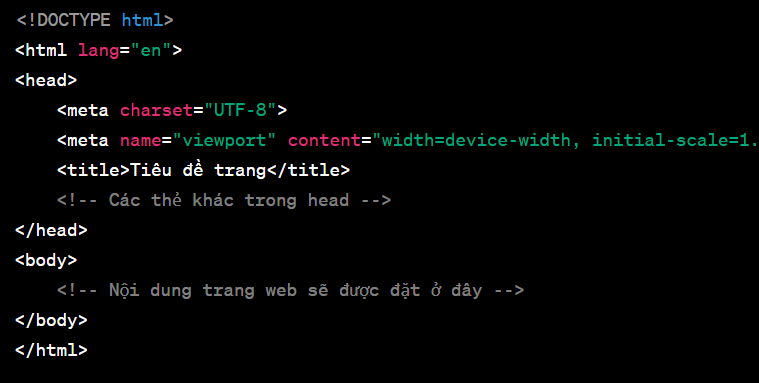
Hỗ trợ offline: Các tính năng như Web Storage và Application Cache giúp ứng dụng web có thể hoạt động offline, giảm thiểu sự phụ thuộc vào kết nối internet.

Tích hợp tốt với CSS3: HTML5 thường được kết hợp với CSS3 và JavaScript để tạo ra các trang web và ứng dụng web đẹp mắt và mạnh mẽ.

HTML5 đã trở thành một tiêu chuẩn ngành và là nền tảng cơ bản cho sự phát triển web hiện đại.

- Cú pháp HTML5

Cú pháp HTML5 giống với cú pháp HTML trước đó, nhưng nó bao gồm một số phần tử và thuộc tính mới. Dưới đây là một cú pháp cơ bản để bắt đầu một tài liệu HTML5:



Hình 28:

Dưới đây là một số điểm quan trọng về cú pháp này:

<!DOCTYPE html>: Định nghĩa loại tài liệu HTML, trong trường hợp này là HTML5.

<html>: Thẻ gốc của tài liệu HTML5. Nó định nghĩa một tài liệu HTML và chứa toàn bộ nội dung của trang web.

<head>: Chứa các thông tin không hiển thị trực tiếp trên trang web như các thẻ meta, title, style, script, và các thẻ khác.

<meta charset="UTF-8">: Đặt bộ mã ký tự của trang web là UTF-8, giúp hỗ trợ nhiều ngôn ngữ và ký tự.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Đặt thuộc tính viewport để làm cho trang web linh hoạt trên các thiết bị di động.

<title>: Đặt tiêu đề của trang web, hiển thị trên thanh tiêu đề của trình duyệt.

<body>: Chứa nội dung hiển thị trực tiếp trên trang web.

Cú pháp trên chỉ là một phần nhỏ của HTML5. Trong <head> và <body>, bạn sẽ thêm nhiều thẻ và thuộc tính khác tùy thuộc vào nhu cầu của trang web cụ thể bạn đang phát triển.

- Khai báo DOCTYPE trong HTML5

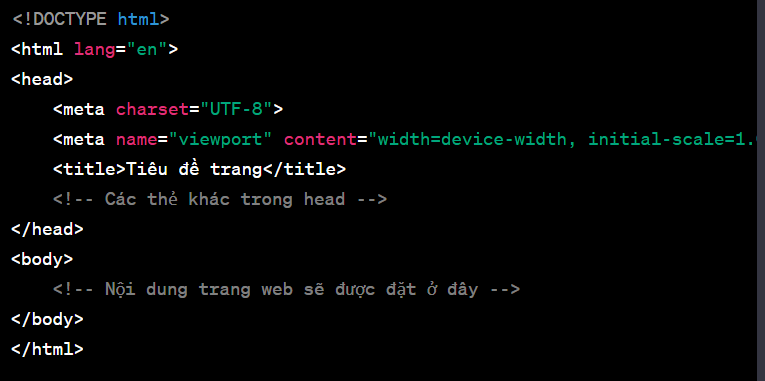
Trong HTML5, khai báo DOCTYPE (Document Type Declaration) không còn phức tạp như trong các phiên bản HTML trước đó. Trong HTML5, bạn chỉ cần sử dụng một DOCTYPE đơn giản như sau:



Hình 29:

Đoạn mã trên đặt ở đầu tài liệu HTML và thông báo cho trình duyệt rằng tài liệu sẽ sử dụng HTML5. Khai báo DOCTYPE HTML5 này rất ngắn gọn và dễ nhớ, giúp đơn giản hóa quá trình khai báo kiểu tài liệu.

Dưới đây là một ví dụ về cách sử dụng DOCTYPE HTML5 trong một tài liệu HTML đầy đủ:



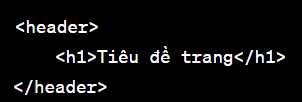
Hình 30:

Đoạn mã trên là một tài liệu HTML5 cơ bản, bắt đầu bằng khai báo DOCTYPE đơn giản và chứa các thẻ <html>, <head>, và <body> như đã mô tả trong câu hỏi trước đó.

- Thẻ ngữ nghĩa

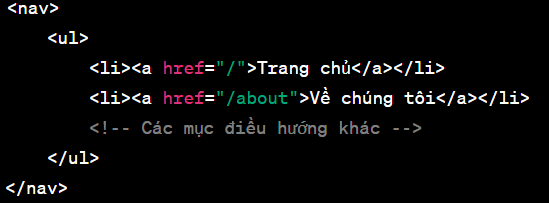
Thẻ ngữ nghĩa là các thẻ trong HTML5 được thiết kế để cung cấp ý nghĩa cụ thể về nội dung mà chúng bao gồm. Các thẻ này giúp máy tính, trình duyệt và công cụ tìm kiếm hiểu rõ hơn về cấu trúc của trang web và nội dung được hiển thị. Các thẻ ngữ nghĩa giúp cải thiện semantics (ngữ nghĩa) của trang web. Dưới đây là một số thẻ ngữ nghĩa quan trọng trong HTML5:

<header>: Định nghĩa phần tiêu đề của trang hoặc một phần của trang.



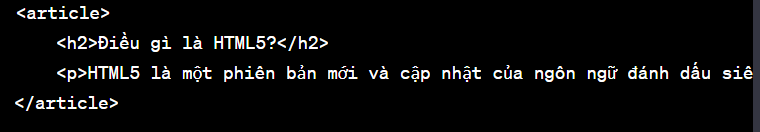
Hình 31:

<nav>: Định nghĩa một phần dành cho các liên kết điều hướng.



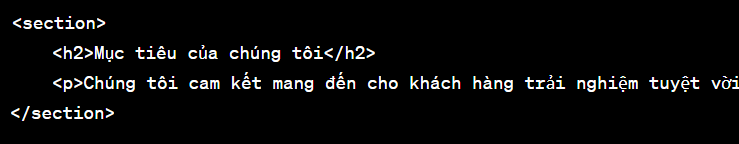
Hình 32:

<article>: Định nghĩa một phần của trang có thể tự đứng độc lập và mang nghĩa đầy đủ.



Hình 33:

<section>: Định nghĩa một phần hoặc khu vực của trang. Thường được sử dụng để nhóm các nội dung có liên quan.



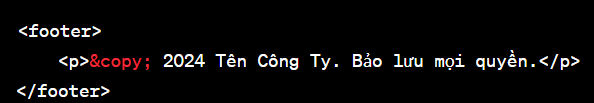
Hình 34:

<aside>: Định nghĩa một phần của trang có thể được xem như một chức năng bổ sung của trang.



Hình 35:

<footer>: Định nghĩa phần cuối trang hoặc một phần của trang, thường chứa thông tin về tác giả, liên hệ, và các liên kết khác.



Hình 36:

- Webform

Webform (biểu mẫu web) là một phần quan trọng trong phát triển web, cho phép người dùng nhập dữ liệu và tương tác với trang web. Các biểu mẫu web thường được sử dụng để thu thập thông tin từ người dùng, xử lý dữ liệu, và gửi thông tin đến máy chủ để xử lý.

Dưới đây là một ví dụ về cách tạo một biểu mẫu web đơn sử dụng HTML:



Hình 37:

<form>: Định nghĩa một biểu mẫu. Thuộc tính action xác định URL mà dữ liệu biểu mẫu sẽ được gửi đến, và method xác định phương thức gửi dữ liệu (thường là "post" hoặc "get").

<label>: Liên kết một nhãn với một trường nhập dữ liệu. Thuộc tính for của <label> phải trùng với id của trường nhập dữ liệu tương ứng.

<input>: Phần tử nhập dữ liệu. Có nhiều loại input như text, email, password được sử dụng cho các mục nhập thông thường.

required: Thuộc tính của <input> để bắt buộc người dùng nhập dữ liệu vào trường.

<submit>: Nút gửi để gửi dữ liệu biểu mẫu.

Một số thuộc tính và phần tử khác có thể được sử dụng để mở rộng chức năng và kiểu dáng của biểu mẫu web. JavaScript thường được sử dụng để thêm các chức năng tương tác nâng cao. Sau khi người dùng điền thông tin và nhấn nút Gửi, dữ liệu thường được gửi đến máy chủ để xử lý thông qua một ngôn ngữ lập trình phía máy chủ như PHP, Node.js, Python, hay một ngôn ngữ khác.

# 

# **KẾT LUẬN CHƯƠNG 2**

HTML (Hypertext Markup Language) và HTML5 đóng vai trò quan trọng trong phát triển web, cung cấp cấu trúc cơ bản và ngữ nghĩa cho trang web. Dưới đây là một số điểm kết luận về HTML và HTML5:

HTML:

Ngôn ngữ Đánh dấu Siêu văn bản: HTML là ngôn ngữ được sử dụng để tạo cấu trúc cơ bản của một trang web bằng cách sử dụng các thẻ và các phần tử.

Chức năng Cơ bản: HTML định nghĩa cấu trúc của một trang web thông qua các thẻ như <html>, <head>, <body>, <div>, <p>, và nhiều thẻ khác.

Điều khiển Nội dung và Hiển thị: HTML cung cấp khả năng điều khiển nội dung và hiển thị trang web trên trình duyệt.

Phiên bản Trước HTML5: Trước HTML5, các phiên bản HTML trước đó (như HTML 4.01 và XHTML) đã định rõ các quy tắc và cú pháp cho việc xây dựng trang web.

HTML5:

Cập Nhật và Mở Rộng Chức Năng: HTML5 là phiên bản mới nhất của HTML, mang lại nhiều cải tiến và mở rộng chức năng so với các phiên bản trước đó.

Hỗ Trợ Đa Phương Tiện: HTML5 cung cấp các phần tử như <audio> và <video> để tích hợp âm thanh và video mà không cần phải sử dụng các plugin bên ngoài.

API và Tích Hợp Đa Dạng Thiết Bị: HTML5 đi kèm với nhiều API như Geolocation API, Web Storage API, WebSockets, và khả năng tích hợp trên nhiều thiết bị.

Thẻ Ngữ Nghĩa: HTML5 giới thiệu các thẻ ngữ nghĩa như <header>, <nav>, <article>, <section>, <footer>, giúp cải thiện semantics của trang web.

Hỗ Trợ Offline: HTML5 cung cấp các tính năng như Web Storage và Application Cache để hỗ trợ ứng dụng web hoạt động offline.

HTML và HTML5 là các công nghệ cơ bản, là nền tảng cho phát triển web hiện đại và đang tiếp tục phát triển để đáp ứng những yêu cầu ngày càng phức tạp của trải nghiệm người dùng 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:

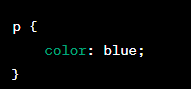


Hình 40:

### 3.4. Selectors

1. Selector Loại (Type Selector):

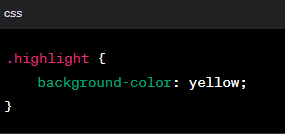
Chọn tất cả các phần tử cùng loại.



Hình 41:

2. Selector Class:

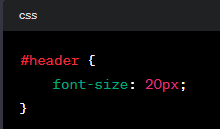
Chọn tất cả các phần tử có một class cụ thể.



Hình 42:

3. Selector ID:

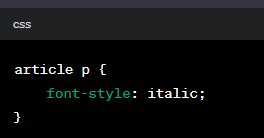
Chọn một phần tử có một ID cụ thể.



Hình 42:

4. Selector Phần Tử (Descendant Selector):

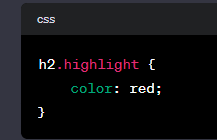
Chọn tất cả các phần tử con của một phần tử cụ thể



Hình 43:

5. Selector Kết hợp (Combination Selector):

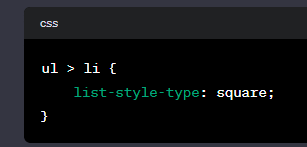
Kết hợp nhiều điều kiện chọn.



Hình 44:

6. Selector Phần Tử Con (Child Selector):

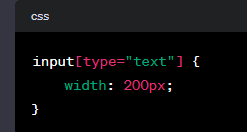
Chọn các phần tử con trực tiếp của một phần tử cụ thể.



Hình 45:

7. Selector Tiền Tố (Attribute Selector):

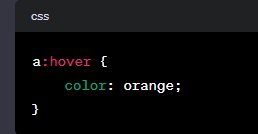
Chọn phần tử có một thuộc tính cụ thể.



Hình 46:

8. Selector Pseudo-class:

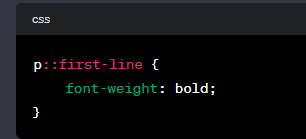
Chọn phần tử trong một trạng thái cụ thể hoặc vị trí trong văn bản.



Hình 47:

1. Selector Pseudo-element:

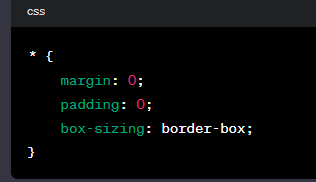
Chọn một phần của một phần tử, chẳng hạn như phần tử đầu tiên hoặc cuối cùng.



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.

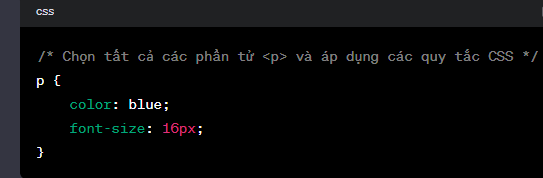
Dưới đây là một ví dụ sử dụng Universal Selector:



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:

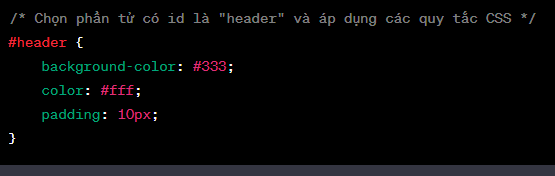


Hình 50:

### 3.4.3. Identity selector

ID Selector:

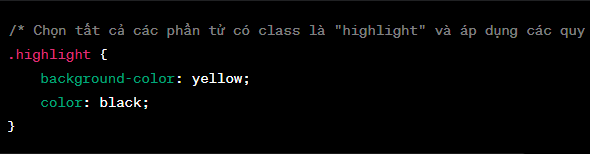
ID Selector được sử dụng để chọn một phần tử cụ thể dựa trên giá trị của thuộc tính id. Cú pháp của ID Selector là " #" sau đó là giá trị id. Ví dụ:



Hình 51:

### 3.4.4. Class selector

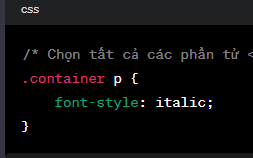
Class Selector trong CSS được sử dụng để chọn tất cả các phần tử có một class cụ thể. Cú pháp của Class Selector là dấu chấm (.) sau đó là tên của class. Ví dụ:



Hình 52:

**3.4.5. Descendant selector**

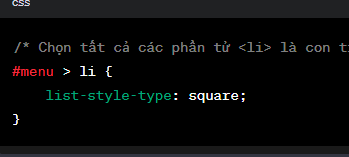
Descendant Selector (Selector Phần tử Con) trong CSS được sử dụng để chọn tất cả các phần tử con của một phần tử cụ thể. Cú pháp của Descendant Selector là sử dụng khoảng trắng ( ) giữa các tên phần tử. Ví dụ:



Hình 53:

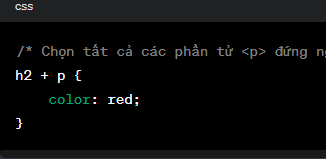
### 3.4.6. Child selector

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:

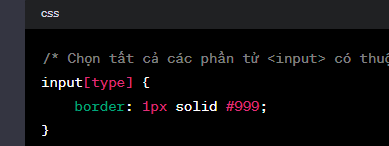


Hình 55:

### 3.4.8. Attribute selector

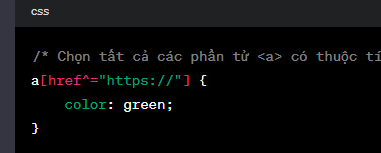
Attribute Selector (Selector Thuộc tính) trong CSS được sử dụng để chọn các phần tử dựa trên giá trị của thuộc tính của chúng. Có một số cách bạn có thể sử dụng Attribute Selector. Dưới đây là một số ví dụ:

Chọn phần tử có một thuộc tính cụ thể:



Hình 56:

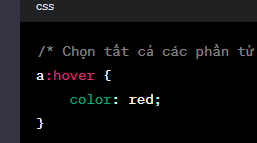
Chọn phần tử có một thuộc tính với giá trị cụ thể:



Hình 57:

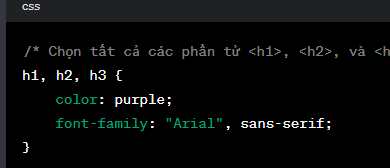
### 3.4.9.Pseudo class selector Pseudo-class Selector (Selector Pseudo-class) trong CSS được sử dụng để chọn các phần tử trong một trạng thái hoặc vị trí cụ thể. Cú pháp của Pseudo-class Selector là sử dụng dấu hai chấm : sau tên phần tử. Dưới đây là một số ví dụ:

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.

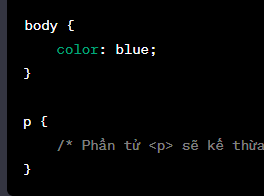
### 3.6. Kế thừa thuộc tính

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:

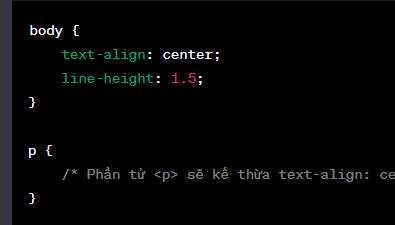
font-size: Kích thước font cũng có thể được kế thừa.



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

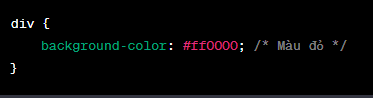
color:

Xác định màu chữ của văn bản.

- Thuộc tính định dạng nền (background)

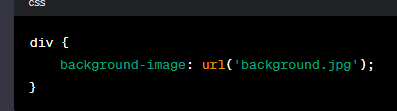
Thuộc tính định dạng nền (background) thường được sử dụng trong lập trình web và CSS để xác định hoặc tùy chỉnh nền của một phần tử HTML. Dưới đây là một số thuộc tính phổ biến liên quan đến định dạng nền:

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.



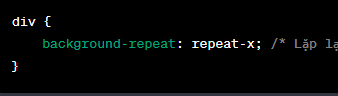
Hình 63:

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



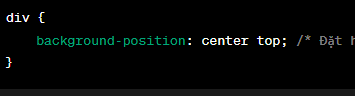
Hình 64:

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



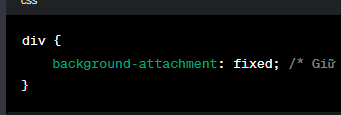
Hình 65:

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, %.



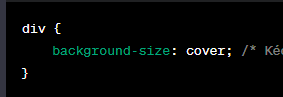
Hình 66:

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.

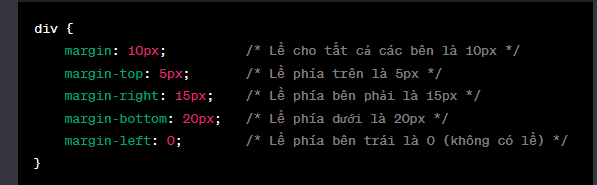
- Các thuộc tính định margin, padding, border

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.

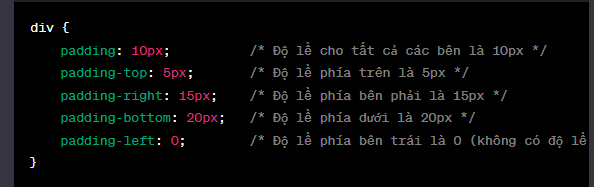


Hình 69

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.



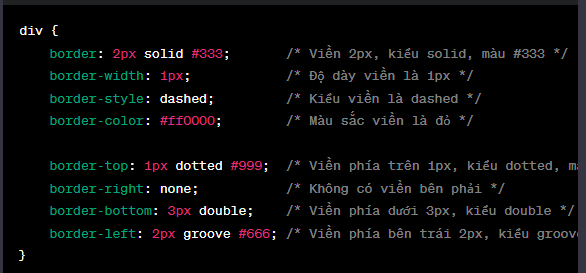
Hình 70:

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:

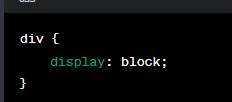
Những thuộc tính trên giúp bạn kiểm soát khoảng cách (margin), độ lề (padding), và trang trí viền (border) của các phần tử trên trang web của bạn.

- 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.

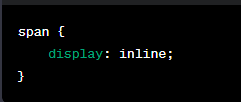
Dưới đây là một số giá trị phổ biến của thuộc tính display:

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.



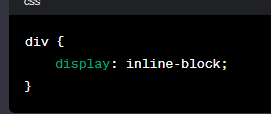
Hình 72:

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.



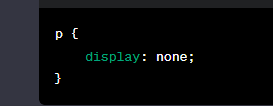
Hình 73:

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.



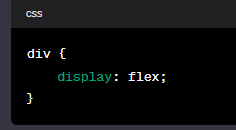
Hình 74:

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 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:

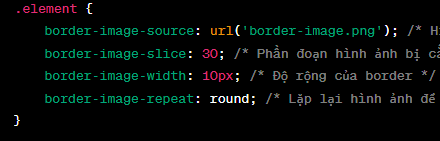
Bạn cũng có thể sử dụng phần trăm để đặt giá trị của border-radius, giúp tạo ra các góc cong linh hoạt hơn:

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:

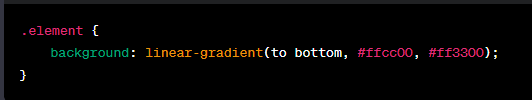
15. Background Blend Mode:

- 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:

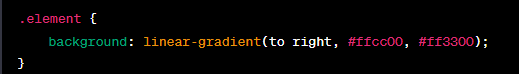
Linear Gradients:

1. Dọc (Top to Bottom):



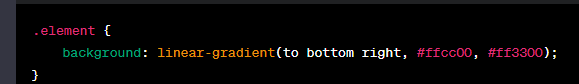
Hình 79:

1. Ngang (Left to Right):



Hình 80:

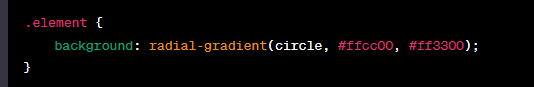
1. Đường chéo:



Hình 81:

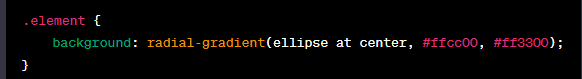
Radial Gradients:

1. Radial:



Hình 82:

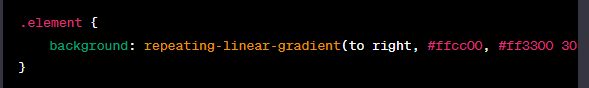
1. Elongated Radial:



Hình 83:

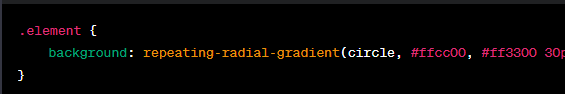
Repeating Gradients:

1. Linear Repeating:



Hình 84:

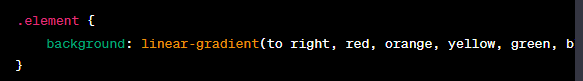
7.Radial Repeating:



Hình 85:

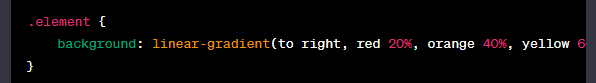
Color Stops:

8.Sử dụng Nhiều Màu:



Hình 86:

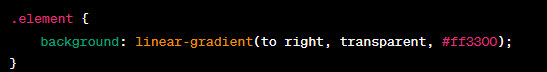
9.Sử dụng Màu và Điểm Dừng:



Hình 87:

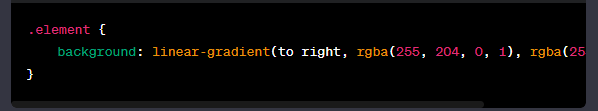
Transparent Gradients:

10.Màu Trong Suốt:



Hình 88:

11.Trộn Giữa Màu và Trong Suốt:



Hình 89:

Kết hợp với Background:

12.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:

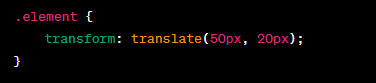
Font Feature Settings: Cho phép bật tắt các tính năng của font, như ligatures.

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:

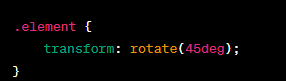
1.Translate (Dịch chuyển):



Hình 92:

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

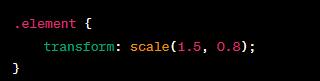
2. Rotate (Xoay):



Hình 93:

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

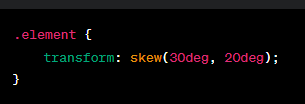
3. Scale (Thu phóng):



Hình 94:

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

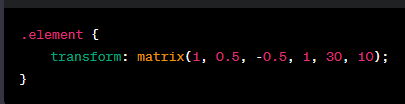
4. Skew (Nghiêng):



Hình 95:

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

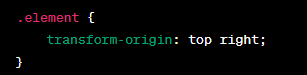
5. Matrix Transform:



Hình 96:

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

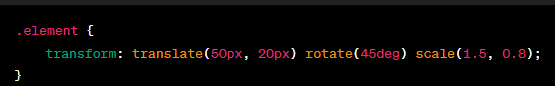
6. Transform Origin:



Hình 97:

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

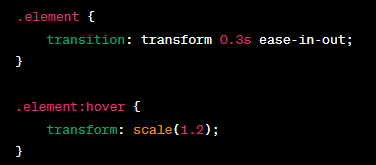
7. Multiple Transforms:



Hình 98:

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

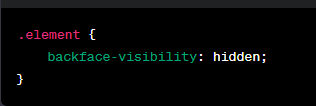
8. Transition (Chuyển động):



Hình 99:

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

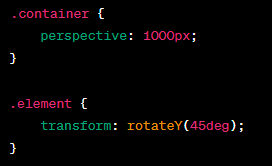
9. Backface Visibility:



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.

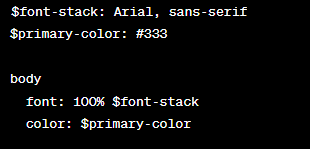
### 3.13. SASS

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.

Responsive Design:

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 TRANG WEB BÁN HÀNG THỜI TRANG**

### 4.1. Ý tưởng của Website

Việc xây dựng một website là cả một quá trình liên tục, nó không đơn thuần là một dự án duy nhất và một lần với các thông tin tĩnh. Việc biên tập, quản lý và duy trì kỹ thuật dài hạn phải bao trùm lên kế hoạch xây dựng website.

-Đối tượng khách hàng: Đây là website bán hàng thời trang nên đối tượng là không phân biệt là nam hay nữ và chủ yếu là sản phẩm cho độ tuổi thanh niên và mặt hàng chủ yếu của website sẽ là quần áo.

-Thời trang bền vững: Tạo ra một trang web tập trung vào việc bán các sản phẩm thời trang được làm từ vật liệu tái chế hoặc có ảnh hưởng ít đến môi trường. Cung cấp thông tin về nguồn gốc và quy trình sản xuất của các sản phẩm để khách hàng có thể lựa chọn mua sắm có trách nhiệm hơn.

-Thời trang cá nhân hóa: Phát triển một trang web cho phép người dùng tạo ra và tùy chỉnh sản phẩm thời trang theo ý thích của họ, từ áo thun, áo sơ mi đến giày dép và phụ kiện. Khách hàng có thể tải lên hình ảnh hoặc thiết kế của riêng mình để in lên sản phẩm.

Thời trang đa dạng về kích cỡ: Tạo ra một trang web bán thời trang chuyên về các sản phẩm dành cho mọi kích cỡ, từ plus size đến petite. Đảm bảo rằng mọi người đều có thể tìm thấy sản phẩm phù hợp với họ mà không gặp khó khăn.

-Quảng bá webssite: Mục tiêu của website là giới thiệu vàquảng bá cửa hàng, cung cấp các sản phẩm về thời trang mà hiện nay các bạn trẻ ưa thích. Khách hàng có thể biết thông tin

cụ thể của từng sản phẩm như: tên sản phẩm, giá cả... Ngoài ra website còn có mục đích giới thiệu cho nhiều khách hàng biết đến cửa hàng, thu hẹp khoảng cách địa lý, giải quyết được vấn đề đường xa, việc đi lại khó khăn... Làm cho việc kinh doanh ngày càng phát triển và được nhiều khách hàng biết đến.

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

Việc xây dựng bố cục cho một trang web tin tức quan trọng để tạo ra trải nghiệm người dùng thuận lợi và hấp dẫn. Dưới đây là một gợi ý về bố cục cho trang web thời trang:

*4.2.1. Header (Đầu trang)*

- Trước khi hiện thị logo thì ở đây em dùng thẻ “Maquee” để thu hút mọi người.

HTML

<header>

    <marquee behavior="scroll" direction="up" scrollamount="3" >

      <p> Đổi hàng miễn phí tại tất cả các của hàng </p>

    <p> Thêm vào giỏ hàng ngay 🛒</p>

    </marquee>

</header>

CSS

 marquee{

      background-color: rgba(171, 119, 85, 0.399);

  }

  marquee p{

      color:rgb(21, 21, 21);

      text-transform: uppercase;

      text-align: center;

      border-radius: 100px;

  }

Thành phẩm

**

- Logo: Hiện thị Logo của trang web để nhận dạng thương hiệu.

HTML

 <div>

        <ul class="menu">

            <li><img  src="logo.jpg" alt="Logo"></li>

<ul>

</div>

CSS

.menu{

      display:flex;

      margin-left: 80px;

      margin-right: 50px;

  }

  .menu img{

      width:80px;

      height:50px;

Thành phẩm:



- Menu: Bao gồm các mục như "Trang chủ ", "Quần áo nam ", "Quần áo nữ ", "Quần áo trẻ em ", v.v. để người dùng dễ dàng điều hướng đến các chuyên mục khác nhau.

Thanh menu được đặt cạnh logo để người mọi người dễ thấy và dễ nhớ khi nhấp vào các mục.

HTML

<main>

    <div>

        <ul class="menu">

            <li><img  src="logo.jpg" alt="Logo"></li>

            <li id="brick">TRANG CHỦ </li>

            <li id="brick">NỮ  </li>

            <li id="brick">NAM  </li>

            <li id="brick">TRẺ EM</li>

            <li id="brick">GIÁ TỐT</li>

            <li id="search">

                <form action="">

                    <input type="text" placeholder=" Tìm kiếm" >

                </form>

            </li>

            <li>🏙 Cửa hàng </li>

            <li>👩 Tài khoản</li>

            <li>🛒 Giỏ hàng </li>

        </ul>

    </div>

</main>

CSS

 .menu li{

      margin:auto;

      list-style-type: none;

      display: block;

    font-weight: 700;

    font-size: 16px;

    line-height: 18px;

    text-transform: uppercase;

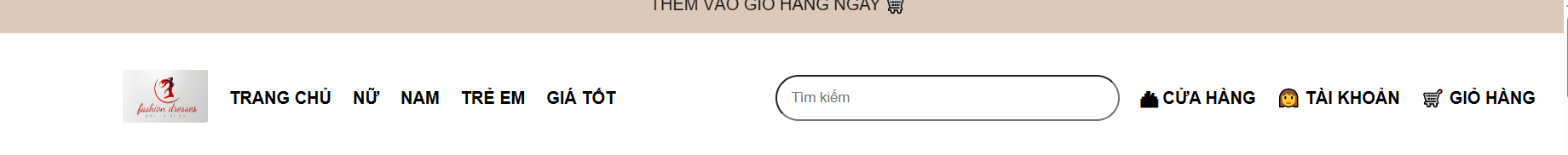
    padding: 0px 10px;

    white-space: nowrap;

    position: relative;

  }

Thành phẩm:



- Ở đầu trang thì không thể thiếu “Ô tìm kiếm” nó cho phép người dùng tìm kiếm các sản phẩm nhanh chóng và dễ dàng. Ở đây khi chúng ta cần sản phẩm gì thì chỉ cần ghi rõ sản phẩm mà ta muốn tìm thì nó sẽ hiện ra các sản phẩm mà ta mong muốn.

HTML

            <li id="search">

                <form action="">

                    <input type="text" placeholder=" Tìm kiếm" >

                </form>

            </li>

CSS

 .menu #search{

      margin-left: 130px;

  }

  #search input{

      width:300px;

      height:20px;

      border-radius: 100px;

      padding: 10px;

  }

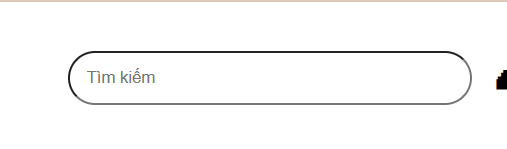
  #brick:hover{

      transition: box-shadow .4s, color.3s;

      box-shadow: 0 3px tomato;

  }

Thành phẩm:



#### 4.2.2. Carousel (Trình chiếu ảnh)

Trong phần trình chiếu ảnh em dùng thẻ div, thẻ label và thẻ input để tạo ra các slide thu hút người xem. Ở đây sẽ hiển thị các bộ sưu tập nổi bật các chủ đề sẽ có trong shop.

HTML:

<div class="slider-container">

    <div class="menu-1">

      <label for="slide-dot-1"></label>

      <label for="slide-dot-2"></label>

      <label for="slide-dot-3"></label>

    </div>

    <input id="slide-dot-1" type="radio" name="slides" checked>

      <div class="slide slide-1"></div>

    <input id="slide-dot-2" type="radio" name="slides">

      <div class="slide slide-2"></div>

    <input id="slide-dot-3" type="radio" name="slides">

      <div class="slide slide-3"></div>

  </div>

CSS:

 .slider-container{

      height: 65%;

      width: 100%;

      position: relative;

      overflow: hidden;

      text-align: center;

    }

    .menu-1 {

      position: absolute;

      left: 0;

      z-index: 900;

      width: 100%;

      bottom: 0;

    }

    .menu-1 label {

      cursor: pointer;

      display: inline-block;

      width: 16px;

      height: 16px;

      background: #fff;

      border-radius: 50px;

      margin: 0 .2em 1em;

      transition: all .3s ease;

      &:hover {

        background: red;

      }  }

    .slide {

      width: 100%;

      height: 80%;

      position: absolute;

      top: 0px;

      left: 50%;

      z-index: 10;

      padding: 8em 1em 0;

      background-size: cover;

      background-position:50% 50%;

      transition: left 0s .75s;

    }

    [id^="slide"]:checked + .slide {

      left: 0;

      z-index: 100;

      transition: left .65s ease-out;

    }

    .slide-1 {

      background-image: url("1.jpg");

    }

    .slide-2 {

      background-image: url("2.jpg");

    }

    .slide-3 {

      background-image: url("3.jpg");

    }

    main{

      padding: 30px;

  }

Khi HTML và CSS kết hợp với nhau thì nó sẽ tạo ra thành phẩm sau:



Dưới phần trình chiếu ảnh thì ta sẽ có các thông tin liên quan đến cách thức thanh toán, vận chuyển, đổi trả hàng đây là thứ tất yếu khi mua hàng. Mọi người sẽ quan tâm đến cách thức vận chuyển, thanh toán hay là đổi trả sản phẩm khi hàng lỗi hoặc không vừa ý.

HTML

<div class="item">

    <div class="icon">

        <div>

            <img src="thanhtoan.png" alt="">

       </div>

    </div>

 <div class="content">

 <h3>Thanh toán khi nhận hàng (COD)</h3>

 <span>Giao hàng toàn quốc. </span>

 </div>

    <div class="icon">

       <div>

           <img src="giao hàng.png" alt="">

      </div>

   </div>

<div class="content">

<h3>Miễn phí giao hàng</h3>

<span>Với đơn hàng trên 199.000đ. </span>

</div>

    <div class="icon">

       <div>

           <img src="đổi hàng.png" alt="">

      </div>

    </div>

      <div class="content">

<h3>Đổi hàng miễn phí</h3>

<span>Trong 30 ngày kể từ ngày mua. </span>

</div>

</div>

CSS

 .item{

    display:flex;

    justify-content: space-around;

    margin-left: 30px;

    margin-right: 100px;

    padding: 10px;

  }

.icon{

    margin-left:auto;

    margin-right: 10px;

}

.icon img{

    width:50px;

    height:50px;

}

.content h3{

    margin:auto;

    list-style-type: none;

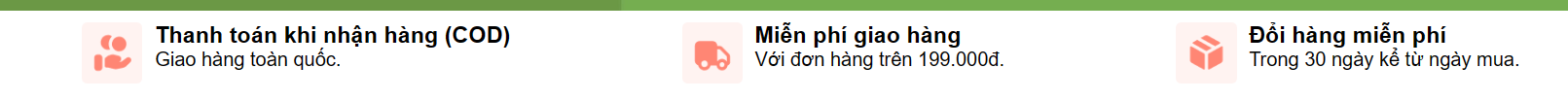
}

.content span{

    margin:0px;

}

Thành phẩm:



Và điều mọi người quan tâm nữa chính là các ưu đãi của shop. Khi mua hàng chúng ta sẽ áp dụng mã giảm giá để cho số tiền hợp với túi tiền của mình, điều này sẽ làm cho người mua ấn tượng khi họ mua sản phẩm với giá hời và họ sẽ quay lại mua nhiều lần nữa.

HTML

<h2> ƯU ĐÃI ĐỘC QUYỀN </h2>

<div class="meli">

    <div class="list">

        <img src="giảmgia1.png" alt="">

    </div>

    <div class="list">

        <img src="giảmgia2.png" alt="">

    </div>

    <div class="list">

        <img src="giảmgia3.png" alt="">

    </div>

</div>

CSS

h2{

    font-size: 30px;

    text-align: left;

    text-transform: uppercase;

    padding: 10px;

    margin-top: 10px;

}

.meli{

    display: flex;

    flex-wrap: wrap;

    justify-content: space-between;

}

.list{

    width: calc(33.3% - 10px);

    margin-bottom: 50px;

    text-align: center;

}

.list img{

    width: 70%;

    border: 2px solid rgb(247, 245, 245);

    margin-bottom: 20px;

    border-radius:10px;

}

Thành phẩm:



*4.2.3. Các nội dung chính:*

Danh mục sản phẩm: Trong phần danh mục thì sẽ chứa các danh mục con bao gồm” áo phông ”, “quần soóc”, “sản phẩm nỉ”. Mỗi tiêu đề em sẽ để chữ to, in đậm, và màu để làm nổi bật tiêu đề.

HTML

 <h1>Danh mục sản phẩm</h1>

  <!--Section1-->

  <section class="section">

      <h2>ÁO PHÔNG </h2>

CSS

 h1{

      font-size: 40px;

      text-align: center;

      text-transform: uppercase;

      padding: 5px;

      margin-top: 5px;

  }

Thành phẩm



Trong các phần danh mục con thì sẽ có các sản phẩm tuy nhiên ở phần trang chủ em chỉ nêu các sản phẩm nổi bật và muốn xem thêm thì ấn vào phần “Xem tất cả” khi ấn vào sẽ hiện ra rất nhiều sản phẩm liên quan.

HTML

 <div class="brick">

              <div class="kem">

                <a href="áo phông.html"><i class="fa-solid fa-seedling" style="color: black;"></i>Xem tất cả</a>

              </div>

          </div>

CSS

 .brick{

    text-align: center;

  }

  .brick .kem{

    display: inline-block;

    color: #e61717;

    font-size: 14px;

    font-weight: 700;

    line-height: 1.5;

    cursor: pointer;

    padding: 8px 16px;

    border-radius: 2px;

    border: 1px solid #da291c;

    background-color: #fff;

}



Sản phẩm: em sẽ để 4 sản phẩm cùng một hàng chia bố cục bằng nhau để khi nhìn vào sẽ vừa mắt hơn, không quá to và cũng không quá nhỏ

Dưới các hình ảnh là tên sản phẩm, giá bán, mua ngay và thêm vào giỏ hàng để mọi người biết được sản phẩm này có tên là gì giá cả ra sao họ sẽ chọn lựa để mua hoặc thêm vào giỏ hàng mà họ mong muốn.

HTML

 <div class="products">

          <!--Sản phẩm số 1-->

          <div class="product">

              <img src="áo thun nữ/anh1.jpg" alt="áo 1">

              <h3>1. Áo thun in form rộng dáng Unisex </h3>

              <li>Giá bán: 120 000 VNĐ</li>

              <button>Mua ngay </button>

              <button> Thêm vào giỏ hàng </button>

              <li>

CSS

.products{

      display: flex;

      flex-wrap: wrap;

      justify-content: space-around;

  }

  .product{

      width: calc(25.0% - 20px);

      margin-bottom: 50px;

      text-align: center;

  }

  .product img{

    width: 350px;

    height:400px;

    border: 2px solid rgb(17, 17, 17);

    margin-bottom: 20px;

    border-radius:10px;

}

  .product h3{

      margin: 0;

  }

  .product p{

      margin: 10px 0;

  }

  .product button{

      background-color: #222;

      color: #fff;

      border: 1px solid gray;

      border-radius: 5px;

      padding: 10px 20px;

      cursor: pointer;

  }

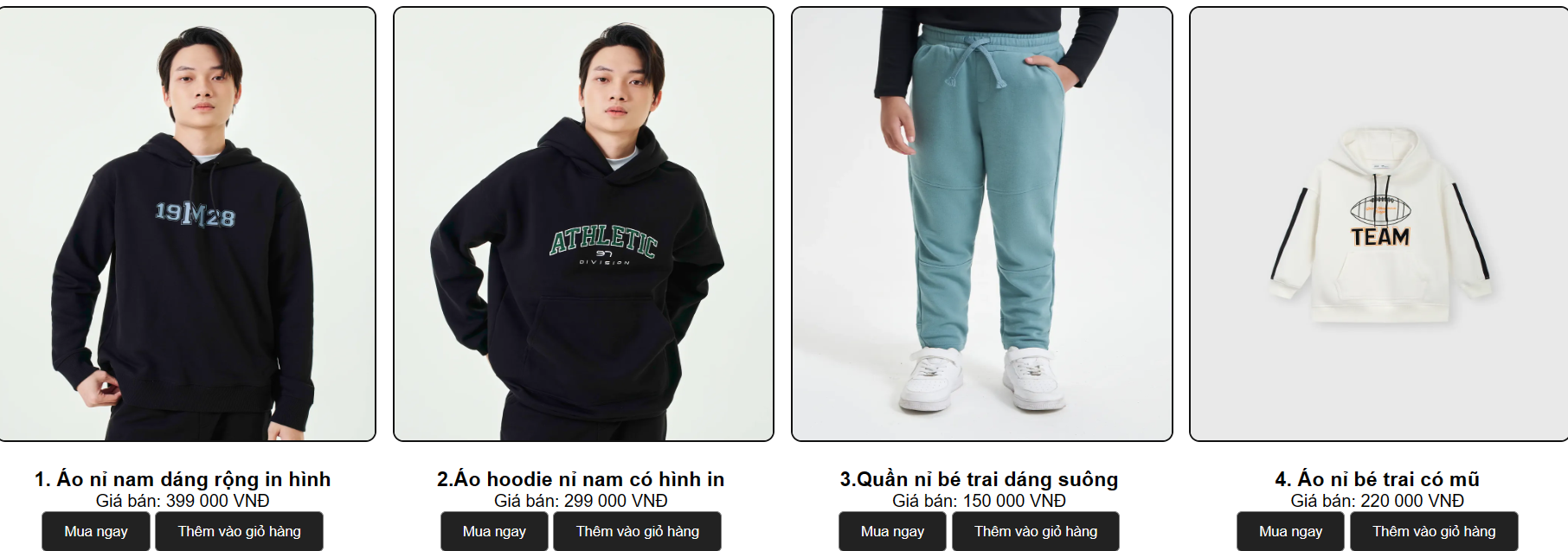
  .product li{

      color:rgb(0, 0, 0);

      margin:auto;

      list-style-type: none;

Thành phẩm



Thông tin chi tiết: Bất kì sản phẩm nào cũng cần những thông tin chi tiết để người mua hiểu rõ hơn về sản phẩm mà họ muốn mua:

HTML

 <details>

                      <summary><b>Thông tin sản phẩm</b>  </summary>

CSS



*4.2.4. Chân trang (Footer):*

Thông tin liên hệ: họ và tên , nội dung , email ở đây mọi người đăng kí hoặc gửi phản hồi về sản phẩm cho shop.

HTML

  <section>

   <h2>Đăng kí nhận bản tin</h2>

    <form action=""  >

        <label for="name"> Họ và tên</label>

        <input type=" text..."  name="name" id="name" placeholder="some text..."  > <br>

        <label for="email">Email</label>

        <input type=" text..."  name="email" id="email"  placeholder="upcoming events..."> <br>

        <label for="message">Nội dung</label>

        <input type=" text..."  name="message" id="message" placeholder="visit our blog..."> <br>

        <button type="submit">Gửi</button>

    </form>

</section>

CSS

#email{

    margin-left: 77px;

}

#message{

    margin-left: 52px;

}

section form button{

    margin-top: 10px;

    width: 50px;

    height: 25px;

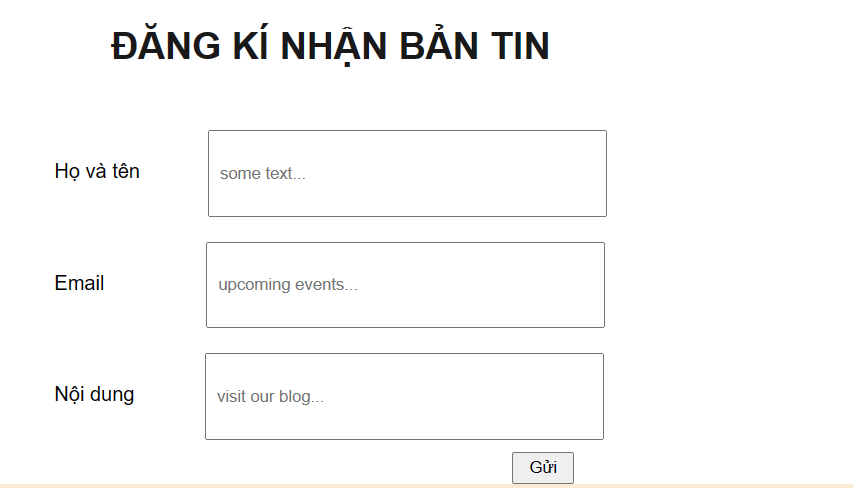
    margin-left: 366px;

}

section form input{

    padding: 8px;

}



Liên kết đến các trang mạng xã hội: Facebook, youtube, tiktok, ...vv.

HTML VÀ CSS

<table  style="background-color:antiquewhite" width="100%" height:100%>

        <tr>

            <th></th>

            <th></th>

            <th></th>

        </tr>

        <tr>

            <td></td>

            <td></td>

        </tr>

        <tr>

            <td width="100px"></td>

            <td width="500px" style="font-size: 20px;"><span><sup style="font-size:25px">Trụ Sở Hà Nội </sup></span>

                <br><br>

                Ngõ 8 ngách 8, Lê Trọng Tấn <br>Hà Đông,Hà Nội<br><br><span style="font-size:25px">Chi

                    Nhánh</span><br><br>

                    25 Trần Phú, Đông Ngàn, Từ Sơn, Bắc Ninh<br><br><br><br>

            </td>

            <td style="font-size: 20px;"><span style="font-size:25px">Hỗ trợ Khách Hàng</span><br><br>

                <span style="font-size:25px">Hotline </span><br><b style="color:rgb(226, 97, 51)">0393 364 283</b>(

                liên hệ trực tiếp )<br><br><span style="font-size:25px">Email </span><br><b

                    style="color:rgb(226, 97, 51)">hathiphuongthanhcntt1706@gmail.com</b><br><br>

                <span style="font-size:25px">Kết nối với chúng tôi </span><br><br>

                <a href="https://www.facebook.com/phuongthanh.hathi.35?mibextid=ibOpuV" target="\_blank">

                    <img src="face.png" alt="" width="35px">&ensp;&ensp;

                    <a href="https://www.youtube.com/@ThanhHa-nr5kx" target="\_blank">

                        <img src="youtube.png" alt="" width="48px">&ensp;&ensp;

                    <a href="https://www.tiktok.com/@user7412868165594?\_t=8kfqETzyXmD&\_r=1" target="\_blank">

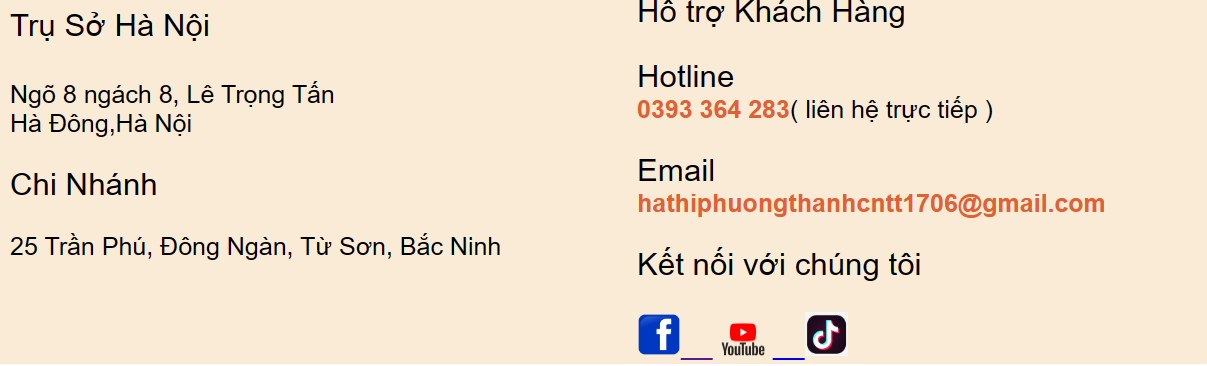
                            <img src="tiktok.jfif" alt="" width="35px">

            </td>

        </tr>

    </table><br><br>

  </main>



Thông tin về bản quyền: Bản quyền © [năm] [tên công ty hoặc cá nhân].

HTML

  <footer>

      © 2024 SHOP THANH🥰

  </footer>

CSS

footer{

      background-color: #b015157d;

      color: #fff;

      padding: 20px;

      text-align: center;

  }

  @media screen and(max-width: 768px) {

      header{

          padding: 10px;

      }

      nav ul{

          flex-wrap: wrap;

      }

      nav li{

          margin-right: 10px;

          margin-bottom: 10px;

      }

      .products{

          flex-wrap: wrap;

      }

      .product{

          width: calc(100%/2 - 10px);

          margin-bottom: 30px;

      }

  }



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

*4.3.1:Phần Header*

Dùng thẻ marquee, thẻ p, thẻ div, thẻ ul, thẻ li, thẻ img và thẻ form

<header>

    <marquee behavior="scroll" direction="up" scrollamount="3" >

      <p> Đổi hàng miễn phí tại tất cả các của hàng </p>

    <p> Thêm vào giỏ hàng ngay 🛒</p>

    </marquee>

</header>

<main>

    <div>

        <ul class="menu">

            <li><img  src="logo.jpg" alt="Logo"></li>

            <li id="brick">TRANG CHỦ </li>

            <li id="brick">NỮ  </li>

            <li id="brick">NAM  </li>

            <li id="brick">TRẺ EM</li>

            <li id="brick">GIÁ TỐT</li>

            <li id="search">

                <form action="">

                    <input type="text" placeholder=" Tìm kiếm" >

                </form>

            </li>

            <li>🏙 Cửa hàng </li>

            <li>👩 Tài khoản</li>

            <li>🛒 Giỏ hàng </li>

        </ul>

    </div>

</main>

Dùng thẻ label, thẻ input tạo ra slide hình ảnh

div class="slider-container">

    <div class="menu-1">

      <label for="slide-dot-1"></label>

      <label for="slide-dot-2"></label>

      <label for="slide-dot-3"></label>

    </div>

    <input id="slide-dot-1" type="radio" name="slides" checked>

      <div class="slide slide-1"></div>

    <input id="slide-dot-2" type="radio" name="slides">

      <div class="slide slide-2"></div>

    <input id="slide-dot-3" type="radio" name="slides">

      <div class="slide slide-3"></div>

  </div>

Dùng thẻ div và thẻ img để tạo ra các phiếu giảm giá và các hình thức giao hàng:

 <div class="item">

    <div class="icon">

        <div>

            <img src="thanhtoan.png" alt="">

       </div>

    </div>

 <div class="content">

 <h3>Thanh toán khi nhận hàng (COD)</h3>

 <span>Giao hàng toàn quốc. </span>

 </div>

<h2> ƯU ĐÃI ĐỘC QUYỀN </h2>

<div class="meli">

    <div class="list">

        <img src="giảmgia1.png" alt="">

    </div>

    <div class="list">

        <img src="giảmgia2.png" alt="">

    </div>

    <div class="list">

        <img src="giảmgia3.png" alt="">

    </div>

*4.3.2. Phần main*

Dùng thẻ section, thẻ p, thẻ img, thẻ h1, thẻ h2, thẻ h3, thẻ button, thẻ li, thẻ details, thẻ summary, thẻ b, thẻ br để tạo ra sản phẩm cùng với các chức năng như mua hàng, thêm vào giỏ hàng và các thông tin liên quan đến sản phẩm.

<h1>Danh mục sản phẩm</h1>

  <!--Section1-->

  <section class="section">

      <h2>ÁO PHÔNG </h2>

      <div class="products">

          <!--Sản phẩm số 1-->

          <div class="product">

              <img src="áo thun nữ/anh1.jpg" alt="áo 1">

              <h3>1. Áo thun in form rộng dáng Unisex </h3>

              <li>Giá bán: 120 000 VNĐ</li>

              <button>Mua ngay </button>

              <button> Thêm vào giỏ hàng </button>

              <li>

                  <details>

                      <summary><b>Thông tin sản phẩm</b>  </summary>

                      <p>

                        - Chất liệu: Vải Cotton 100% 2 chiều <br>

                          - Màu sắc: Đen<br>

                          - Form: Local Brand - Unisex        <br>

                   - Chất lượng in: In lụa dùng mực Nhật Bản chất lượng cao  <br>

                        - Bảo quản: Có thể giặt máy & giặt ngâm                <br>

                          Bảng size:                             <br>

                          S: từ 45kg đến 55kg - 1m50 đến 1m60          <br>

                          M: từ 55kg đến 65kg - 1m60 đến 1m7  <br>

                          L: từ 65kg đến 80kg - 1m70 đến 1m80 <br>

                          XL: từ 80kg trở lên - 1m80 trở lên<br>

                          </p>

                   </details>

                  </li>

          </div>

*4.3.3. Phần Footer*

Dùng thẻ table, tr, td, table, input, label,h2, form

 <section>

   <h2>Đăng kí nhận bản tin</h2>

    <form action=""  >

               <label for="name"> Họ và tên</label>

        <input type=" text..."  name="name" id="name" placeholder="some text..."  > <br>

        <label for="email">Email</label>

        <input type=" text..."  name="email" id="email"  placeholder="upcoming events..."> <br>

        <label for="message">Nội dung</label>

        <input type=" text..."  name="message" id="message" placeholder="visit our blog..."> <br>

        <button type="submit">Gửi</button>

    </form>

</section>

    <table  style="background-color:antiquewhite" width="100%" height:100%>

        <tr>

            <th></th>

            <th></th>

            <th></th>

        </tr>

        <tr>

            <td></td>

            <td></td>

        </tr>

        <tr>

            <td width="100px"></td>

            <td width="500px" style="font-size: 20px;"><span><sup style="font-size:25px">Trụ Sở Hà Nội </sup></span>

                <br><br>

                Ngõ 8 ngách 8, Lê Trọng Tấn <br>Hà Đông,Hà Nội<br><br><span style="font-size:25px">ChiNhánh</span><br><br>

                    25 Trần Phú, Đông Ngàn, Từ Sơn, Bắc Ninh<br><br><br><br>

            </td>

            <td style="font-size: 20px;"><span style="font-size:25px">Hỗ trợ Khách Hàng</span><br><br>

                <span style="font-size:25px">Hotline </span><br><b style="color:rgb(226, 97, 51)">0393 364 283</b>(

                liên hệ trực tiếp )<br><br><span style="font-size:25px">Email </span><br><b

    style="color:rgb(226, 97,51)">hathiphuongthanhcntt1706@gmail.com</b><br><br>

<span style="font-size:25px">Kết nối với chúng tôi </span><br><br>

<a href="https://www.facebook.com/phuongthanh.hathi.35?mibextid=ibOpuV"target="\_blank">

                    <img src="face.png" alt="" width="35px">&ensp;&ensp;

                    <a href="https://www.youtube.com/@ThanhHa-nr5kx" target="\_blank">

                    <img src="youtube.png" alt="" width="48px">&ensp;&ensp;

                    <a href="https://www.tiktok.com/@user7412868165594?\_t=8kfqETzyXmD&\_r=1" target="\_blank">

                  <img src="tiktok.jfif" alt="" width="35px">

            </td>

        </tr>

    </table><br><br>

  </main>

  <footer>

      © 2024 SHOP THANH🥰

  </footer>

### 4.4. Kết quả đạt được

Tăng doanh số bán hàng : Giao diện thân thiện và dễ dàng tìm kiếm sản phẩm giúp tăng hiệu suất bán hàng và doanh số.

Tăng tương tác với khách hàng: Cung cấp các tính năng như đánh giá sản phẩm, phản hồi từ khách hàng và các chương trình khuyến mãi có thể kích thích tương tác từ người dùng.

Tăng lượng khách hàng quay lại: Một trang web thời trang tốt sẽ tạo ra trải nghiệm mua sắm tích cực, khiến khách hàng muốn quay lại lần nữa.

Tăng tính nhận biết thương hiệu: Thiết kế độc đáo và chất lượng cao sẽ tạo ra ấn tượng mạnh mẽ về thương hiệu với khách hàng.

Tăng tính tương tác trên mạng xã hội: Một trang web thời trang phản ánh tốt trên các nền tảng mạng xã hội có thể thu hút sự chú ý từ cộng đồng trực tuyến, tăng cơ hội tiếp cận và tương tác với đối tượng mục tiêu.

Tăng tính đa dạng sản phẩm: Bố cục tốt giúp người dùng dễ dàng tìm kiếm và khám phá các sản phẩm đa dạng, tăng cơ hội mua hàng từ nhiều danh mục khác nhau.

Tăng tính theo dõi và phân tích: Bằng cách tích hợp các công cụ phân tích và theo dõi, doanh nghiệp có thể hiểu rõ hơn về hành vi của khách hàng và tối ưu hóa trang web để đáp ứng nhu cầu của họ.

**Kết luận chương 4**

*a)Một số điểm còn hạn chế của trang web:*

Trải nghiệm người dùng kém: Giao diện không thân thiện, khó sử dụng hoặc không tương thích với các thiết bị di động có thể làm giảm trải nghiệm người dùng.

Chất lượng hình ảnh và mô tả sản phẩm không tốt: Thông tin không đầy đủ hoặc không chính xác về sản phẩm có thể gây nghi ngờ và làm giảm lòng tin của khách hàng.

Tốc độ tải trang chậm: Thời gian tải trang dài có thể khiến người dùng mất kiên nhẫn và rời bỏ trang web.

Quá trình thanh toán phức tạp: Nếu quá trình thanh toán không thuận tiện hoặc an toàn, người dùng có thể từ chối hoàn tất giao dịch.

Khuyến mãi và ưu đãi không hấp dẫn: Thiếu chiến lược tiếp thị hiệu quả hoặc cung cấp khuyến mãi không hấp dẫn có thể làm giảm sự quan tâm của khách hàng.

Cạnh tranh cao: Sự cạnh tranh giữa các trang web bán thời trang là rất cao, vì vậy trang web cần phải nỗ lực để nổi bật và thu hút khách hàng

Quản lý hàng tồn kho không hiệu quả: Sự cập nhật chậm trễ hoặc thông tin không chính xác về tồn kho có thể gây ra thất vọng cho khách hàng khi sản phẩm họ muốn mua không có sẵn.

Chính sách vận chuyển và trả hàng không rõ ràng: Chính sách vận chuyển phức tạp hoặc không minh bạch có thể làm mất lòng tin của khách hàng.

*b) Hướng phát triển trang web*

Tối ưu hóa trải nghiệm người dùng: Tăng cường trải nghiệm người dùng bằng cách cải thiện giao diện người dùng, tối ưu hóa trang web cho các thiết bị di động, giảm thời gian tải trang, và tăng tính tương tác.

Nâng cao chất lượng sản phẩm và nội dung: Cung cấp mô tả sản phẩm chi tiết, hình ảnh chất lượng cao, và nội dung hấp dẫn để tăng cường lòng tin của khách hàng và thúc đẩy việc mua sắm.

Mở rộng dải sản phẩm: Đảm bảo rằng trang web cung cấp một loạt các sản phẩm thời trang đa dạng để đáp ứng nhu cầu của nhiều đối tượng khách hàng khác nhau. Tăng cường chiến lược tiếp thị trực tuyến: Sử dụng các kênh tiếp thị trực tuyến như quảng cáo trên mạng xã hội, email marketing, và tối ưu hóa công cụ tìm kiếm để tăng cường sự nhận biết và thu hút khách hàng mới.

Phát triển chương trình khách hàng trung thành: Tạo ra các chương trình khuyến mãi, ưu đãi đặc biệt, và điểm thưởng để khuyến khích sự trung thành của khách hàng và tăng cường lợi ích cho việc mua sắm trên trang web.

Mở rộng quy mô hoạt động: Xem xét việc mở rộng hoạt động kinh doanh bằng cách mở thêm cửa hàng trực tuyến, mở rộng quy mô vận chuyển quốc tế, hoặc phát triển mối quan hệ với các nhà cung cấp mới.

Liên tục cải thiện hệ thống: Theo dõi và phản hồi phản hồi của khách hàng, phân tích dữ liệu để hiểu rõ hành vi của khách hàng và điều chỉnh chiến lược kinh doanh và tiếp thị theo hướng đó.

# **Kết luận**

Trong thời kỳ công nghệ thông tin phát triển như hiện nay thì môn thiết kế web làkhông thể thiếu hiện nay lập trình web đang là một trong những công việc được nhiềungười yêu thích lập trình lựa chọn. Bởi nó không chỉ đem lại cơ hội việc làm cao mà nócòn đem lại nguồn thu nhập ổn định cho nhiều người. Nhiều người nghĩ rằng lập trìnhweb là một công việc khô khan và khó nhằn, chắc chỉ dành cho người có năng khiếubẩm sinh mà thôi. Thực tế thì lại không phải như vậy, hầu như ai cũng có thể học lậptrình web được. Miễn là chịu học một cách kiên trì, quyết tâm thực hành, theo đuổi đến cùng

HTML (HyperText Markup Language) và CSS (Cascading Style Sheets) là các khốixây dựng cơ bản nhất của mã hóa web. Nếu không có HTML và CSS, bạn có thể sẽkhông tạo ra một trang web. Và tất cả những gì bạn sẽ có là văn bản đơn giản khôngđược định dạng trên màn hình. Bạn thậm chí không thể thêm hình ảnh vào một trangmà không có HTMLTrước khi bạn bắt đầu trên bất kỳ con đường sự nghiệp phát triển web nào. Bạn sẽ phảithành thạo mã hóa với HTM; và CSS. Bạn có thể có kiến thức vững chắc về HTML vàCSS trong một vài tuần. Tuy nhiên bạn không nên chủ quan với nó và coi nó là dễdàng.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Lê Văn Phong, Bài giảng Thiết kế web và triển khai hệ thống phần mềm, trường Đại học Đại Nam, 2024.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.