**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, GIỚI THIỆU VỀ DU LỊCH VÙNG TÂY BẮC )**

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

**Sinh viên thực hiện:**  **Nguyễn Thanh Danh**

**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, GIỚI THIỆU VỀ DU LỊCH VÙNG TÂY BẮC)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020126 | Nguyễn Thanh Danh | 09/10/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

Trước hết em xin bày tỏ lòng biết ơn sâu sắc đến thầy ThS. Lê Văn Phong,

người thầy đã trực tiếp giảng dạy, động viên, trao đổi kiến thức và chỉ bảo em

trong suốt quá trình học tập để em hoàn thành nội dung đề tài này.

Em cũng xin gửi lời cảm ơn đến cô giáo chủ nhiệm, các thầy cô giảng dạy môn

Thiết kế web và triển khai hệ thống phần mềm và các thầy cô lãnh đạo khoa đã tận tình động tiên, góp ý, giúp đỡ em trong suốt thời gian vừa qua.

Cuối cùng em vô cùng tri ân sự hướng dẫn tận tình và theo dõi sát sao đầy tinh

thần trách nhiệm cùng lòng thương mến thầy cô trong suất quá trình em theo học

tại ngôi trường Đại Nam.

**Sinh viên**

Nguyễn Thanh Danh

**MỤC LỤC**

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 7](#_Toc161222193)

[**1.1. Các khái niệm cơ bản** 7](#_Toc161222194)

[Khái niệm cơ bản về thiết kế web và triển khai hệ thống phần mềm 7](#_Toc161222195)

[**1.2. Phân biệt Web tĩnh, Web động** 8](#_Toc161222196)

[Phân biệt Web Tĩnh và Web Động 8](#_Toc161222197)

[**Một số lưu ý:** 9](#_Toc161222198)

[**1.3**. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, **HomePage,** 9](#_Toc161222199)

[**Web Site, WebServer, URL-Uniform Resource Location, Browser, …)** 9](#_Toc161222200)

[**1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …)** 15](#_Toc161222201)

[Một số công cụ dùng thiết kế Web phổ biến: 15](#_Toc161222202)

[**Kết luận Chương 1** 16](#_Toc161222203)

[Kết luận về thiết kế web và triển khai hệ thống phần mềm: 16](#_Toc161222204)

[**CHƯƠNG II: HTML & HTML5** 18](#_Toc161222205)

[**2.1. Tổng quan về HTML** 18](#_Toc161222206)

[**2.2. Cấu trúc tổng quát trang HTML** 18](#_Toc161222207)

[**2.3. Các thẻ HTML thông dụng** 20](#_Toc161222208)

[**2.4. Các thẻ tạo biểu mẫu (form)** 21](#_Toc161222209)

[**2.5. Một số thẻ HTML đặc biệt** 23](#_Toc161222210)

[**2.6. HTML5** 25](#_Toc161222211)

[*2.6.1. HTML 5 là gì?* 25](#_Toc161222212)

[*2.6.2. Cú pháp HTML5* 26](#_Toc161222213)

[*2.6.3. Khai báo giữa DOCTYOE trong HTML5* 28](#_Toc161222214)

[*2.6.4. Thẻ ngữ nghĩa* 29](#_Toc161222215)

[*2.6.5. Webform* 31](#_Toc161222216)

[**Kết Luận Chương 2** 32](#_Toc161222217)

[**CHƯƠNG III: CSS và CSS3** 33](#_Toc161222218)

[**3.1. CSS là gì?** 33](#_Toc161222219)

[**3.2. Cú pháp CSS** 34](#_Toc161222220)

[**3.3. Áp dụng CSS vào trang HTML** 35](#_Toc161222222)

[**3.4. Selectors** 36](#_Toc161222223)

[*3.4.1. Universal selector* 37](#_Toc161222224)

[*3.4.2. Type selector* 38](#_Toc161222225)

[*3.4.3. Identity selector* 39](#_Toc161222226)

[*3.4.4. Class selector* 39](#_Toc161222227)

[***3.4.5. Descendant selector*** 40](#_Toc161222228)

[*3.4.6. Child selector* 41](#_Toc161222229)

[*3.4.7. Adjacent sibling selector* 42](#_Toc161222230)

[*3.4.8. Attribute selector* 43](#_Toc161222231)

[*3.4.9. Pseudo class selector* 44](#_Toc161222232)

[*3.4.10. Group selector* 45](#_Toc161222233)

[**3.5. Đơn vị đo lường CSS** 46](#_Toc161222234)

[1. Đơn vị tương đối - *Relative Units* 47](#_Toc161222235)

[2. Đơn vị tuyệt đối - *Absolute Units* 47](#_Toc161222236)

[**3.6. Kế thừa thuộc tính** 47](#_Toc161222237)

[**3.7. Các nhóm thuộc tính trong CSS** 48](#_Toc161222238)

[Các thuộc tính định margin 49](#_Toc161222240)

[Cấu trúc 49](#_Toc161222241)

[Cấu trúc 50](#_Toc161222242)

[Cấu trúc 51](#_Toc161222243)

[Display CSS là gì? 52](#_Toc161222244)

[Cấu trúc 53](#_Toc161222245)

[**3.8. Float & Clear** 53](#_Toc161222246)

[**Thuộc tính Float CSS là gì?** 53](#_Toc161222247)

[**Thuộc tính Float bao gồm các giá trị nào?** 54](#_Toc161222248)

[Phân biệt thuộc tính Float CSS và Clear CSS 54](#_Toc161222249)

[**3.9. Flex** 54](#_Toc161222250)

[Flex css là gì? 54](#_Toc161222251)

[Cấu trúc của flex css 55](#_Toc161222252)

[**3.10. Grid** 55](#_Toc161222253)

[Grid CSS là gì? 55](#_Toc161222254)

[Trình duyệt tương thích 55](#_Toc161222255)

[**3.11. CSS3** 55](#_Toc161222256)

[**3.12. SCSS** 57](#_Toc161222257)

[**3.13. SASS** 57](#_Toc161222258)

[**Kết luận Chương 3** 58](#_Toc161222259)

[CHƯƠNG 4: THIẾT KẾ WEBISTE, GIỚI THIỆU VỀ DU LỊCH VÙNG TÂY BẮC 59](#_Toc161222260)

[**4.1. Ý tưởng của Website** 59](#_Toc161222261)

[**4.2. Xây dựng bố cục của trang Web** 59](#_Toc161222262)

[**4.3. Thiết kế trang Web bằng HTML và CSS** 60](#_Toc161222263)

[**4.5. Kết quả đạt được** 61](#_Toc161222264)

[**Kết luận chương 4** 61](#_Toc161222265)

**MỤC LỤC HÌNH ẢNH**

[Hình 1 11](#_Toc160871033)

[Hình 2 11](#_Toc160871034)

[Hình 3 12](#_Toc160871035)

[Hình 4 12](#_Toc160871036)

[Hình 5 13](#_Toc160871037)

[Hình 6 13](#_Toc160871038)

[Hình 7 14](#_Toc160871039)

[Hình 8 14](#_Toc160871040)

[Hình 9 15](#_Toc160871041)

[Hình 10 15](#_Toc160871042)

[Hình 11 16](#_Toc160871043)

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

## **1.1. Các khái niệm cơ bản**

## Khái niệm cơ bản về thiết kế web và triển khai hệ thống phần mềm

**Thiết kế web** là quá trình tạo ra giao diện và cấu trúc cho một trang web. Nó bao gồm các yếu tố như bố cục, đồ họa, typography và khả năng sử dụng. Thiết kế web đóng vai trò quan trọng trong việc thu hút người dùng và mang lại trải nghiệm tốt cho họ.

**Triển khai hệ thống phần mềm** là quá trình cài đặt và cấu hình phần mềm để nó có thể hoạt động trong môi trường sản xuất. Quá trình này bao gồm các bước như:

* Cài đặt phần mềm
* Cấu hình phần mềm
* Di chuyển dữ liệu
* Kiểm tra và thử nghiệm
* Đào tạo người dùng

**Dưới đây là một số khái niệm cơ bản về thiết kế web:**

* **Giao diện người dùng (UI):** Là phần mà người dùng trực tiếp tương tác với trang web. Nó bao gồm các yếu tố như bố cục, màu sắc, hình ảnh, typography và các nút điều khiển.
* **Trải nghiệm người dùng (UX):** Là cảm nhận của người dùng khi sử dụng trang web. Nó bao gồm các yếu tố như khả năng sử dụng, hiệu suất và mức độ dễ dàng tìm kiếm thông tin.
* **Responsive design:** Là thiết kế web có thể tự động điều chỉnh giao diện để phù hợp với mọi kích thước màn hình, từ máy tính để bàn đến điện thoại thông minh.
* **Web tĩnh:** Là trang web có nội dung được lưu trữ cố định và không thay đổi theo thời gian.
* **Web động:** Là trang web có nội dung được tạo ra tự động dựa trên yêu cầu của người dùng.

**Dưới đây là một số khái niệm cơ bản về triển khai hệ thống phần mềm:**

* **Môi trường sản xuất:** Là môi trường mà phần mềm được sử dụng bởi người dùng thực tế.
* **Môi trường thử nghiệm:** Là môi trường được sử dụng để kiểm tra và thử nghiệm phần mềm trước khi triển khai.
* **Kiểm tra và thử nghiệm:** Là quá trình kiểm tra phần mềm để đảm bảo rằng nó hoạt động chính xác và đáp ứng các yêu cầu.
* **Đào tạo người dùng:** Là quá trình cung cấp cho người dùng kiến thức và kỹ năng cần thiết để sử dụng phần mềm.

**Sự khác biệt giữa thiết kế web và triển khai hệ thống phần mềm:**

* Thiết kế web tập trung vào giao diện và cấu trúc của trang web, trong khi triển khai hệ thống phần mềm tập trung vào việc cài đặt và cấu hình phần mềm để nó có thể hoạt động trong môi trường sản xuất.
* Thiết kế web là một phần của quá trình phát triển web, trong khi triển khai hệ thống phần mềm là bước cuối cùng trong quá trình phát triển phần mềm.
* Thiết kế web thường được thực hiện bởi các nhà thiết kế web, trong khi triển khai hệ thống phần mềm thường được thực hiện bởi các kỹ sư phần mềm.

**Kết luận:**

Thiết kế web và triển khai hệ thống phần mềm là hai khái niệm quan trọng trong lĩnh vực công nghệ thông tin. Thiết kế web đóng vai trò quan trọng trong việc thu hút người dùng và mang lại trải nghiệm tốt cho họ, trong khi triển khai hệ thống phần mềm đảm bảo rằng phần mềm có thể hoạt động hiệu quả trong môi trường sản xuất.

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

## Phân biệt Web Tĩnh và Web Động

**Web tĩnh:**

* **Nội dung:**
  + Không thay đổi thường xuyên.
  + Được tạo ra bằng HTML, CSS và JavaScript.
  + Ví dụ: trang web giới thiệu công ty, portfolio cá nhân.
* **Ưu điểm:**
  + Tốc độ tải trang nhanh.
  + Chi phí phát triển thấp.
  + Dễ dàng bảo trì.
* **Nhược điểm:**
  + Khó cập nhật nội dung.
  + Tính tương tác thấp.
  + Không phù hợp cho các trang web có nội dung thay đổi thường xuyên.

**Web động:**

* **Nội dung:**
  + Thay đổi thường xuyên.
  + Được tạo ra bằng các ngôn ngữ lập trình phía máy chủ như PHP, Python, Ruby on Rails.
  + Ví dụ: trang web tin tức, mạng xã hội, trang web bán hàng.
* **Ưu điểm:**
  + Dễ dàng cập nhật nội dung.
  + Tính tương tác cao.
  + Phù hợp cho các trang web có nội dung thay đổi thường xuyên.
* **Nhược điểm:**
  + Tốc độ tải trang có thể chậm hơn web tĩnh.
  + Chi phí phát triển cao hơn.
  + Bảo trì phức tạp hơn.

**Lựa chọn loại web nào:**

* **Web tĩnh:** Phù hợp cho các trang web có nội dung ít thay đổi, cần tốc độ tải trang nhanh và chi phí phát triển thấp.
* **Web động:** Phù hợp cho các trang web có nội dung thay đổi thường xuyên, cần tính tương tác cao và có thể cập nhật nội dung dễ dàng.

## **Một số lưu ý:**

* Có thể kết hợp cả web tĩnh và web động trong cùng một trang web.
* Việc lựa chọn loại web nào phụ thuộc vào nhu cầu cụ thể của từng dự án.

## **1.3**. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, **HomePage,**

## **Web Site, WebServer, URL-Uniform Resource Location, Browser, …)**

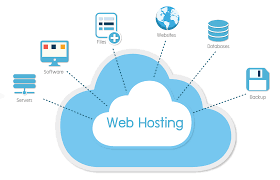
**Tên miền (Domain Name)**: Giống như địa chỉ nhà của trang web của bạn, là một tên duy nhất và dễ nhớ mà mọi người có thể sử dụng để truy cập trang web của bạn. Ví dụ: [đã xoá URL không hợp lệ], facebook.com.



Hình

Hình ảnh 1.3.1. Domain name

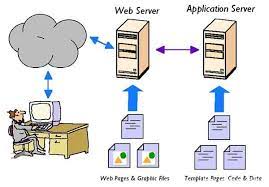
**Hosting:** Không gian lưu trữ trên máy chủ web nơi các tệp trang web của bạn được lưu trữ. Giống như việc thuê chỗ để đặt máy chủ của bạn và tất cả các tệp cần thiết để trang web của bạn hoạt động.



Hình

Hình ảnh 1.3.2. Hosting

**Máy chủ web (Web Server):** Máy tính mạnh mẽ được kết nối với internet 24/7, lưu trữ các tệp trang web của bạn và phục vụ chúng cho bất kỳ ai truy cập trang web của bạn thông qua tên miền của bạn.



Hình

Hình ảnh 1.3.3. Máy chủ web (Web Server)

**Trang chủ (HomePage):** Trang đầu tiên mọi người nhìn thấy khi họ truy cập trang web của bạn. Đây là cơ hội của bạn để tạo ấn tượng tốt và hướng dẫn người dùng đến các phần khác của trang web của bạn.



Hình

Hình ảnh 1.3.4. Trang chủ (Homepage)

**Trang web (Website):** Một bộ sưu tập các trang web được liên kết với nhau, được truy cập thông qua một tên miền duy nhất. Nó có thể chứa văn bản, hình ảnh, video, âm thanh và các nội dung khác được tổ chức thành các trang riêng biệt.



Hình

Hình ảnh 1.3.5. Trang web (Website)

**URL (Uniform Resource Locator):** Địa chỉ cụ thể của một trang web trên internet. Nó bao gồm tên miền, giao thức (ví dụ: http, https) và đường dẫn đến tệp hoặc tài nguyên cụ thể trên trang web. Ví dụ: [đã xoá URL không hợp lệ].



Hình

Hình ảnh 1.3.6. URL (Uniform Resource Locator)

**Trình duyệt (Browser):** Phần mềm cho phép bạn truy cập và tương tác với các trang web. Các trình duyệt phổ biến bao gồm Chrome, Firefox, Safari, Edge và Opera.

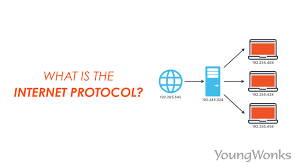
Ngoài ra, còn có một số thuật ngữ khác mà bạn có thể gặp phải:



Hình

Hình ảnh 1.3.7. Trình duyệt (Browser)

* **IP (Internet Protocol):** Một địa chỉ số duy nhất được gán cho mỗi thiết bị được kết nối với internet. Máy chủ web cũng có địa chỉ IP, nhưng tên miền dễ nhớ hơn nhiều.



Hình

Hình ảnh 1.3.8. IP (Internet Protocol)

* **HTML (HyperText Markup Language):** Ngôn ngữ đánh dấu được sử dụng để tạo cấu trúc cơ bản của trang web.



Hình

Hình ảnh 1.3.9. HTML (HyperText Markup Language)

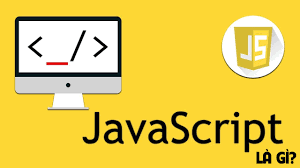
* **CSS (Cascading Style Sheets):** Ngôn ngữ được sử dụng để kiểm soát kiểu dáng của trang web, chẳng như màu sắc, phông chữ và bố cục.



Hình

Hình ảnh 1.3.10. CSS (Cascading Style Sheets)

* **JavaScript:** Ngôn ngữ lập trình được sử dụng để thêm các tính năng tương tác cho trang web của bạn.



Hình

Hình ảnh 1.3.11. JavaScript

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

## Một số công cụ dùng thiết kế Web phổ biến:

**1. Visual Studio Code:**

* Trình soạn thảo mã miễn phí, mã nguồn mở và nhẹ.
* Hỗ trợ nhiều ngôn ngữ lập trình, bao gồm HTML, CSS và JavaScript.
* Cung cấp nhiều tính năng hữu ích cho việc phát triển web, chẳng hạn như tô sáng cú pháp, tự động hoàn thành mã và gỡ lỗi.

**2. Adobe Dreamweaver:**

* Môi trường phát triển web tích hợp (IDE) trả phí.
* Cung cấp nhiều tính năng cho việc thiết kế và phát triển web, chẳng hạn như giao diện WYSIWYG, hỗ trợ Git và tích hợp với các công cụ Adobe khác.

**3. Sublime Text:**

* Trình soạn thảo mã trả phí với nhiều tính năng mạnh mẽ.
* Hỗ trợ nhiều ngôn ngữ lập trình, bao gồm HTML, CSS và JavaScript.
* Cung cấp nhiều tính năng hữu ích cho việc phát triển web, chẳng hạn như tô sáng cú pháp, tự động hoàn thành mã và gỡ lỗi.

**4. Atom:**

* Trình soạn thảo mã miễn phí, mã nguồn mở và có thể tùy chỉnh cao.
* Hỗ trợ nhiều ngôn ngữ lập trình, bao gồm HTML, CSS và JavaScript.
* Cung cấp nhiều tính năng hữu ích cho việc phát triển web, chẳng hạn như tô sáng cú pháp, tự động hoàn thành mã và gỡ lỗi.

**5. Webflow:**

* Nền tảng thiết kế web trực quan cho phép bạn tạo trang web mà không cần viết mã.
* Cung cấp giao diện kéo và thả dễ sử dụng, cũng như nhiều mẫu và thành phần để bạn lựa chọn.

**6. Wix:**

* Nền tảng thiết kế web dựa trên đám mây cho phép bạn tạo trang web mà không cần viết mã.
* Cung cấp giao diện kéo và thả dễ sử dụng, cũng như nhiều mẫu và thành phần để bạn lựa chọn.

**7. WordPress:**

* Hệ thống quản trị nội dung (CMS) mã nguồn mở cho phép bạn tạo trang web và blog.
* Cung cấp nhiều tính năng và plugin để bạn tùy chỉnh trang web của mình.

## **Kết luận Chương 1**

## Kết luận về thiết kế web và triển khai hệ thống phần mềm:

Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực quan trọng trong việc tạo ra một trang web hoặc ứng dụng web thành công.

**Thiết kế web** tập trung vào việc tạo ra giao diện và cấu trúc của trang web, đảm bảo tính thẩm mỹ, trực quan và dễ sử dụng cho người dùng.

**Triển khai hệ thống phần mềm** tập trung vào việc cài đặt và cấu hình phần mềm để nó có thể hoạt động hiệu quả trên máy chủ web.

**Cả hai lĩnh vực này đều cần có sự phối hợp chặt chẽ với nhau để tạo ra một sản phẩm hoàn chỉnh đáp ứng nhu cầu của người dùng.**

**Dưới đây là một số kết luận chính về thiết kế web và triển khai hệ thống phần mềm:**

* **Thiết kế web:**
  + Là yếu tố đầu tiên thu hút người dùng và tạo ấn tượng ban đầu về trang web.
  + Cần đảm bảo tính thẩm mỹ, trực quan và dễ sử dụng cho người dùng.
  + Có nhiều yếu tố cần quan tâm như bố cục, màu sắc, typography, hình ảnh, v.v.
* **Triển khai hệ thống phần mềm:**
  + Đảm bảo phần mềm hoạt động hiệu quả và ổn định trên máy chủ web.
  + Cần có kiến thức chuyên môn về phần mềm và máy chủ.
  + Bao gồm các bước như cài đặt phần mềm, cấu hình phần mềm, di chuyển dữ liệu, v.v.

**Để tạo ra một trang web hoặc ứng dụng web thành công, cần có sự kết hợp hài hòa giữa thiết kế web và triển khai hệ thống phần mềm.**

**Ngoài ra, cần lưu ý một số điểm sau:**

* **Lựa chọn công cụ phù hợp:** Có nhiều công cụ thiết kế web và triển khai hệ thống phần mềm khác nhau. Cần lựa chọn công cụ phù hợp với nhu cầu và kỹ năng của bạn.
* **Cập nhật xu hướng mới:** Lĩnh vực thiết kế web và triển khai hệ thống phần mềm luôn thay đổi và phát triển. Cần cập nhật xu hướng mới để tạo ra sản phẩm phù hợp với thị hiếu người dùng.
* **Bảo mật:** Cần đảm bảo bảo mật cho trang web hoặc ứng dụng web của bạn để tránh các nguy cơ tấn công mạng.

# **CHƯƠNG II: HTML & HTML5**

**(**Hyper Text Markup Language**)**

## **2.1. Tổng quan về HTML**

**HTML** là viết tắt của **HyperText Markup Language**, hay còn gọi là **Ngôn ngữ Đánh dấu Siêu văn bản**. Đây là ngôn ngữ nền tảng để tạo dựng trang web, được sử dụng để định nghĩa cấu trúc và nội dung của các trang web.

**Đặc điểm chính của HTML:**

* **Dễ học và sử dụng:** HTML có cú pháp đơn giản, dễ hiểu, phù hợp cho cả người mới bắt đầu.
* **Linh hoạt:** HTML cho phép bạn tạo ra các trang web với nhiều bố cục và chức năng khác nhau.
* **Có thể mở rộng:** HTML có thể được kết hợp với các ngôn ngữ khác như CSS và JavaScript để tạo ra các trang web phức tạp và tương tác cao.

**Cấu trúc cơ bản của một trang HTML:**

* **DOCTYPE:** Khai báo phiên bản HTML đang sử dụng.
* **Thẻ HTML:** Định nghĩa phần đầu và phần thân của trang web.
* **Thẻ HEAD:** Chứa thông tin về trang web, như tiêu đề, meta dữ liệu, liên kết đến các tập tin CSS.
* **Thẻ BODY:** Chứa nội dung chính của trang web, như văn bản, hình ảnh, video.

**Một số thẻ HTML phổ biến:**

* **Thẻ heading:** Định nghĩa các tiêu đề (H1, H2, H3...)
* **Thẻ paragraph:** Định nghĩa đoạn văn bản.
* **Thẻ image:** Chèn hình ảnh vào trang web.
* **Thẻ link:** Tạo liên kết đến các trang web khác.
* **Thẻ table:** Tạo bảng dữ liệu.
* **Thẻ form:** Tạo biểu mẫu để người dùng nhập dữ liệu.

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

\* Cấu trúc tổng quát của một trang HTML bao gồm 3 phần chính:

**1. Khai báo DOCTYPE:**

* Dòng đầu tiên của trang HTML là khai báo DOCTYPE.
* DOCTYPE cho trình duyệt biết trang web sử dụng phiên bản HTML nào.
* Ví dụ:

HTML

<!DOCTYPE **html**>

**2. Phần HEAD:**

* Phần HEAD nằm giữa thẻ <head> và </head>.
* Phần HEAD chứa các thông tin về trang web, như:
  + Tiêu đề trang web
  + Meta dữ liệu
  + Liên kết đến các tập tin CSS
  + Script JavaScript
* Ví dụ:

HTML

<head>

<title>Trang web của tôi</title>

<meta charset="utf-8">

<link rel="stylesheet" href="style.css">

</head>

**3. Phần BODY:**

* Phần BODY nằm giữa thẻ <body> và </body>.
* Phần BODY chứa nội dung chính của trang web, như:
  + Văn bản
  + Hình ảnh
  + Video
  + Bảng
  + Biểu mẫu
* Ví dụ:

HTML

<body>

<h1>Đây là tiêu đề trang web</h1>

<p>Đây là nội dung trang web</p>

<img src="image.jpg" alt="Hình ảnh">

</body>

**Ngoài 3 phần chính trên, trang HTML có thể bao gồm các phần khác như:**

* **Phần COMMENT:** Dùng để ghi chú trong code HTML, không hiển thị trên trang web.
* **Phần SCRIPT:** Dùng để chèn JavaScript vào trang web.

**Dưới đây là ví dụ về một trang HTML đơn giản:**

HTML

<!DOCTYPE **html**>

<html>

<head>

<title>Trang web của tôi</title>

<meta charset="utf-8">

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Đây là tiêu đề trang web</h1>

<p>Đây là nội dung trang web</p>

<img src="image.jpg" alt="Hình ảnh">

</body>

</html>

**Lưu ý:**

* Cấu trúc HTML phải tuân theo quy tắc cú pháp.
* Các thẻ HTML phải được viết đúng và đóng mở đầy đủ.
* Có thể sử dụng các công cụ hỗ trợ để viết code HTML, như Sublime Text, Visual Studio Code.

## **2.3. Các thẻ HTML thông dụng**

Dưới đây là danh sách các thẻ HTML thông dụng được sử dụng để xây dựng cấu trúc và định dạng nội dung cho trang web:

**Cấu trúc trang web:**

* **<!DOCTYPE html>**: Khai báo phiên bản HTML.expand\_more
* **<html>**: Thẻ mở đầu của trang HTML.expand\_more
* **<head>**: Chứa thông tin về trang web, như tiêu đề, metadata, stylesheet.expand\_more
* **<title>**: Xác định tiêu đề của trang web.expand\_more
* **<meta>**: Cung cấp thông tin metadata về trang web, như charset, author, description.expand\_more
* **<link>**: Liên kết đến các tài nguyên bên ngoài, như stylesheet, JavaScript.expand\_more
* **<body>**: Chứa nội dung chính của trang web.expand\_more
* **</html>**: Thẻ đóng cuối của trang HTML.expand\_more

**Định dạng văn bản:**

* **<h1>** đến **<h6>**: Xác định các cấp độ heading khác nhau.expand\_more
* **<p>**: Định nghĩa một đoạn văn bản.expand\_more
* **<br>**: Tạo dòng mới.expand\_more
* **<strong>**: Định dạng văn bản đậm.
* **<em>**: Định dạng văn bản nghiêng.expand\_more

**Liên kết:**

* **<a>**: Tạo liên kết đến một trang web khác hoặc một phần khác trong cùng trang.expand\_more

**Hình ảnh:**

* **<img>**: Chèn hình ảnh vào trang web.expand\_more

**Danh sách:**

* **<ul>**: Tạo danh sách không thứ tự.
* **<ol>**: Tạo danh sách có thứ tự.
* **<li>**: Định nghĩa một mục trong danh sách.

**Bảng:**

* **<table>**: Tạo bảng.
* **<tr>**: Định nghĩa một hàng trong bảng.
* **<td>**: Định nghĩa một ô trong bảng.

**Khác:**

* **<div>**: Định nghĩa một phần tử block-level.
* **<span>**: Định nghĩa một phần tử inline.expand\_more
* **<hr>**: Tạo đường kẻ ngang.expand\_more

**Lưu ý:**

* Hầu hết các thẻ HTML đều có thẻ đóng tương ứng.expand\_more Ví dụ: <p> có thẻ đóng </p>.
* Một số thẻ HTML không cần thẻ đóng.expand\_more Ví dụ: <br>.
* Có thể sử dụng thuộc tính để định dạng thêm cho các thẻ HTML. Ví dụ: <p style="color: red;">.

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

Biểu mẫu HTML cho phép người dùng tương tác và gửi thông tin đến máy chủ. Dưới đây là các thẻ HTML thông dụng để tạo biểu mẫu:

**1. Thẻ <form>:**

* Xác định một biểu mẫu.
* Thuộc tính quan trọng:
  + action: Xác định URL để gửi dữ liệu biểu mẫu.
  + method: Xác định phương thức gửi dữ liệu (GET hoặc POST).

**2. Thẻ <input>:**

* Tạo các phần tử đầu vào khác nhau trong biểu mẫu.
* Các loại input phổ biến:
  + text: Nhập văn bản.
  + password: Nhập mật khẩu.
  + radio: Nút radio.
  + checkbox: Hộp kiểm.
  + submit: Nút gửi biểu mẫu.
  + reset: Nút đặt lại biểu mẫu.
* Thuộc tính quan trọng:
  + type: Xác định loại input.
  + name: Xác định tên của phần tử đầu vào.
  + value: Xác định giá trị mặc định của phần tử đầu vào.

**3. Thẻ <label>:**

* Nhãn cho các phần tử đầu vào.
* Thuộc tính quan trọng:
  + for: Liên kết nhãn với phần tử đầu vào bằng ID.

**4. Thẻ <textarea>:**

* Tạo vùng nhập văn bản nhiều dòng.
* Thuộc tính quan trọng:
  + name: Xác định tên của vùng nhập văn bản.
  + cols: Xác định số lượng cột.
  + rows: Xác định số lượng hàng.

**5. Thẻ <select>:**

* Tạo menu chọn.
* Thuộc tính quan trọng:
  + name: Xác định tên của menu chọn.
  + size: Xác định số lượng lựa chọn hiển thị đồng thời.

**6. Thẻ <option>:**

* Xác định các lựa chọn trong menu chọn.
* Thuộc tính quan trọng:
  + value: Xác định giá trị của lựa chọn.
  + selected: Xác định lựa chọn mặc định.

**Ví dụ:**

HTML

<form action="xuly.php" method="post">

<label for="name">Tên:</label>

<input type="text" id="name" name="name" required>

<br>

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

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

<br>

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

<textarea id="message" name="message" rows="5" cols="30"></textarea>

<br>

<input type="submit" value="Gửi">

<input type="reset" value="Đặt lại">

</form>

**Lưu ý:**

* Có thể sử dụng thêm nhiều thẻ HTML khác để tạo biểu mẫu phức tạp hơn.
* Cần sử dụng JavaScript để xử lý dữ liệu biểu mẫu được gửi đến máy chủ.

## **2.5. Một số thẻ HTML đặc biệt**

Ngoài các thẻ HTML phổ biến được sử dụng để xây dựng cấu trúc và định dạng nội dung cho trang web, còn có một số thẻ HTML đặc biệt với chức năng chuyên biệt hơn. Dưới đây là một số ví dụ:

**1. Thẻ <DOCTYPE>:**

* Khai báo phiên bản HTML của trang web.
* Ví dụ:

HTML

<!DOCTYPE **html**>

**2. Thẻ <meta>:**

* Cung cấp thông tin metadata về trang web, như charset, author, description.
* Ví dụ:

HTML

<meta charset="utf-8">

<meta name="author" content="Bard">

<meta name="description" content="Đây là trang web của tôi">

**3. Thẻ <link>:**

* Liên kết đến các tài nguyên bên ngoài, như stylesheet, JavaScript.
* Ví dụ:

HTML

<link rel="stylesheet" href="style.css">

<script src="script.js"></script>

**4. Thẻ <img>:**

* Chèn hình ảnh vào trang web.
* Thuộc tính quan trọng:
  + src: Xác định URL của hình ảnh.
  + alt: Cung cấp mô tả thay thế cho hình ảnh.
* Ví dụ:

HTML

<img src="image.jpg" alt="Đây là một hình ảnh">

**5. Thẻ <iframe>:**

* Chèn nội dung từ một trang web khác vào trang web hiện tại.
* Thuộc tính quan trọng:
  + src: Xác định URL của trang web cần chèn.
* Ví dụ:

HTML

<iframe src="https://www.google.com"></iframe>

**6. Thẻ <video>:**

* Chèn video vào trang web.
* Thuộc tính quan trọng:
  + src: Xác định URL của video.
* Ví dụ:

HTML

<video src="video.mp4"></video>

**7. Thẻ <audio>:**

* Chèn âm thanh vào trang web.
* Thuộc tính quan trọng:
  + src: Xác định URL của âm thanh.
* Ví dụ:

HTML

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

**8. Thẻ <canvas>:**

* Vẽ đồ họa trên trang web.
* Cần sử dụng JavaScript để vẽ đồ họa.
* Ví dụ:

HTML

<canvas width="500" height="300"></canvas>

**9. Thẻ <svg>:**

* Vẽ đồ họa vector trên trang web.
* Ví dụ:

HTML

<svg width="500" height="300">

<circle cx="250" cy="150" r="100" fill="red" />

</svg>

**10. Thẻ <script>:**

* Chèn mã JavaScript vào trang web.
* Ví dụ:

HTML

<script>

alert("Hello, world!");

</script>

## **2.6. HTML5**

### *2.6.1. HTML 5 là gì?*

HTML5 là phiên bản thứ năm và **là bản sửa đổi chính cuối cùng** của Tiêu chuẩn Ngôn ngữ Đánh dấu Siêu văn bản (HTML). Nó được phát triển bởi World Wide Web Consortium (W3C) với mục đích cải thiện khả năng của HTML để hỗ trợ các công nghệ và ứng dụng web mới nhất. Mặc dù không còn là tiêu chuẩn đang được cập nhật tích cực, nhưng nó vẫn là nền tảng cho phần lớn các trang web ngày nay.

**Dưới đây là một số tính năng và cải tiến chính được giới thiệu trong HTML5:**

**1. Các phần tử ngữ nghĩa mới:** HTML5 giới thiệu các phần tử ngữ nghĩa mới cung cấp thêm ý nghĩa và cấu trúc cho các trang web. Các phần tử này mô tả nội dung chúng chứa, giúp các công cụ tìm kiếm và công nghệ hỗ trợ dễ hiểu trang web hơn. Ví dụ bao gồm <header>, <nav>, <section>, <article>, <aside> và <footer>.

**2. Hỗ trợ đa phương tiện:** HTML5 cung cấp hỗ trợ tích hợp để nhúng nội dung đa phương tiện như âm thanh và video trực tiếp trong các trang web, loại bỏ nhu cầu về các plugin bên ngoài như Flash. Điều này đạt được bằng cách sử dụng các phần tử <audio> và <video>.

**3. Khả năng ngoại tuyến:** HTML5 giới thiệu các tính năng như Web Storage (localStorage và sessionStorage) và Application Cache, cho phép các ứng dụng web lưu trữ dữ liệu cục bộ trên thiết bị của người dùng. Điều này cho phép chúng hoạt động ở một mức độ nào đó ngay cả khi người dùng ngoại tuyến.

**4. Canvas và SVG:** HTML5 giới thiệu phần tử <canvas> để tạo đồ họa động và tương tác bằng JavaScript và phần tử <svg> để nhúng và thao tác đồ họa vector có thể mở rộng.

**5. Biểu mẫu:** HTML5 giới thiệu các loại đầu vào mới cho biểu mẫu, chẳng hạn như email, search, url, date và time, cùng với các tính năng như xác thực biểu mẫu và văn bản giữ chỗ, giúp việc tạo và tương tác biểu mẫu hiệu quả hơn.

**6. Web Sockets:** HTML5 giới thiệu WebSockets, cho phép giao tiếp hai chiều thời gian thực giữa máy khách web và máy chủ. Điều này cho phép các tính năng như trò chuyện trực tiếp, chỉnh sửa cộng tác và cập nhật dữ liệu thời gian thực.

**7. Định vị địa lý:** HTML5 cung cấp quyền truy cập vào vị trí địa lý của người dùng thông qua API Định vị địa lý, cho phép các dịch vụ và tính năng dựa trên vị trí.

**Nhìn chung, HTML5 đã cải thiện đáng kể khả năng phát triển web, mở đường cho trải nghiệm web phong phú, tương tác và năng động hơn.**

***2.6.2. Cú pháp HTML5***

Cú pháp HTML5 là tập hợp các quy tắc chi phối cách viết mã HTML. Cú pháp này được thiết kế để đơn giản, dễ đọc và dễ hiểu.

Dưới đây là một số điểm chính về cú pháp HTML5:

**1. Khai báo DOCTYPE:**

Dòng đầu tiên của tài liệu HTML5 phải là khai báo DOCTYPE. Khai báo này cho trình duyệt biết rằng tài liệu được viết bằng HTML5.

HTML

<!DOCTYPE **html**>

**2. Cấu trúc cơ bản:**

Tài liệu HTML5 được cấu trúc bởi các thẻ HTML. Thẻ HTML là các từ khóa được bao bọc trong dấu ngoặc nhọn (< >).

Ví dụ:

HTML

<h1>Đây là tiêu đề</h1>

<p>Đây là đoạn văn bản.</p>

**3. Thẻ đóng:**

Hầu hết các thẻ HTML đều có thẻ đóng tương ứng. Thẻ đóng có cùng tên với thẻ mở nhưng có thêm dấu gạch chéo (/).

Ví dụ:

HTML

<h1>Đây là tiêu đề</h1>

**4. Thuộc tính:**

Thẻ HTML có thể có các thuộc tính để định dạng hoặc cung cấp thêm thông tin về nội dung của thẻ. Thuộc tính được viết sau tên thẻ trong dấu ngoặc kép ("").

Ví dụ:

HTML

<p style="color: red;">Đây là đoạn văn bản màu đỏ.</p>

**5. Rỗng thẻ:**

Một số thẻ HTML không có nội dung và không cần thẻ đóng. Những thẻ này được gọi là thẻ rỗng. Thẻ rỗng được viết bằng cách thêm dấu gạch chéo (/) sau tên thẻ.

Ví dụ:

HTML

<img src="image.jpg" alt="Đây là một hình ảnh">

**6. Phân biệt chữ hoa chữ thường:**

Cú pháp HTML5 không phân biệt chữ hoa chữ thường. Điều này có nghĩa là bạn có thể viết tên thẻ và thuộc tính bằng chữ hoa hoặc chữ thường.

Ví dụ:

HTML

<h1>Đây là tiêu đề</h1>

**7. Các phần tử ngữ nghĩa:**

HTML5 giới thiệu các phần tử ngữ nghĩa mới để cung cấp thêm ý nghĩa và cấu trúc cho các trang web. Ví dụ:

* <header>: Chứa thông tin tiêu đề của trang web.
* <nav>: Chứa các liên kết điều hướng.
* <section>: Chứa một phần nội dung của trang web.
* <article>: Chứa một bài báo hoặc mục độc lập.
* <aside>: Chứa nội dung bổ sung cho nội dung chính.
* <footer>: Chứa thông tin chân trang của trang web.

**8. Đa phương tiện:**

HTML5 cung cấp hỗ trợ tích hợp cho âm thanh và video. Bạn có thể nhúng âm thanh và video vào trang web của mình bằng cách sử dụng các thẻ <audio> và <video>.

**9. Biểu mẫu:**

HTML5 cung cấp các tính năng mới để tạo biểu mẫu. Bạn có thể sử dụng các loại đầu vào mới, chẳng hạn như email, search, url, date và time.

**10. Web Sockets:**

HTML5 giới thiệu Web Sockets, cho phép giao tiếp hai chiều thời gian thực giữa máy khách web và máy chủ.

***2.6.3. Khai báo giữa DOCTYOE trong HTML5***

**Khai báo DOCTYPE** là dòng đầu tiên của một trang HTML5. Nó cho trình duyệt biết rằng tài liệu được viết bằng HTML5 và giúp trình duyệt hiển thị trang web chính xác.

**Cú pháp:**

HTML

<!DOCTYPE **html**>

**Giải thích:**

* <!DOCTYPE>: Khai báo bắt đầu một DOCTYPE.
* html: Cho trình duyệt biết rằng tài liệu được viết bằng HTML.

**Ví dụ:**

HTML

<!DOCTYPE **html**>

<html>

<head>

<title>Đây là trang web của tôi</title>

</head>

<body>

<h1>Đây là tiêu đề</h1>

<p>Đây là đoạn văn bản.</p>

</body>

</html>

**Lưu ý:**

* Khai báo DOCTYPE **không** bắt buộc trong HTML5. Tuy nhiên, **khuyến nghị** sử dụng nó để đảm bảo khả năng tương thích với các trình duyệt cũ hơn.
* Khai báo DOCTYPE **phải** được đặt trên dòng đầu tiên của tài liệu HTML.
* Không được có bất kỳ khoảng trắng nào trước khai báo DOCTYPE.

**Ngoài ra, còn có một số loại DOCTYPE khác:**

* **HTML 4.01:**

HTML

<!DOCTYPE **html** **PUBLIC** "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

* **XHTML 1.0 Strict:**

HTML

<!DOCTYPE **html** **PUBLIC** "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

**Tuy nhiên, HTML5 là phiên bản HTML phổ biến nhất hiện nay và nên được sử dụng cho hầu hết các trang web.**

**Tài liệu tham khảo:**

* **Tài liệu web MDN:** [đã xoá URL không hợp lệ]
* **W3Schools:** [đã xoá URL không hợp lệ]
* **HTML Dog:** [đã xoá URL không hợp lệ]

***2.6.4. Thẻ ngữ nghĩa***

Thẻ ngữ nghĩa là những thẻ HTML cung cấp thêm thông tin về ý nghĩa và cấu trúc của nội dung trang web. Việc sử dụng thẻ ngữ nghĩa giúp cải thiện khả năng truy cập, khả năng đọc và khả năng hiểu của trang web đối với các trình duyệt, công cụ tìm kiếm và người dùng, đặc biệt là người dùng hỗ trợ công nghệ.

**Dưới đây là một số thẻ ngữ nghĩa phổ biến trong HTML5:**

* **header:** Chứa thông tin tiêu đề của trang web.
* **nav:** Chứa các liên kết điều hướng.
* **section:** Chứa một phần nội dung của trang web.
* **article:** Chứa một bài báo hoặc mục độc lập.
* **aside:** Chứa nội dung bổ sung cho nội dung chính.
* **footer:** Chứa thông tin chân trang của trang web.

**Ví dụ:**

HTML

<header>

<h1>Đây là trang web của tôi</h1>

</header>

<nav>

<a href="index.html">Trang chủ</a>

<a href="about.html">Giới thiệu</a>

<a href="contact.html">Liên hệ</a>

</nav>

<section>

<h2>Đây là tiêu đề phần 1</h2>

<p>Đây là nội dung phần 1.</p>

</section>

<section>

<h2>Đây là tiêu đề phần 2</h2>

<p>Đây là nội dung phần 2.</p>

</section>

<footer>

<p>Bản quyền &copy; 2023</p>

</footer>

**Lợi ích của việc sử dụng thẻ ngữ nghĩa:**

* **Cải thiện khả năng truy cập:** Thẻ ngữ nghĩa giúp các công nghệ hỗ trợ, như trình đọc màn hình, hiểu rõ hơn về nội dung trang web và cung cấp trải nghiệm tốt hơn cho người dùng.
* **Cải thiện khả năng đọc:** Thẻ ngữ nghĩa giúp trình duyệt hiển thị trang web một cách có cấu trúc và dễ đọc hơn.
* **Cải thiện khả năng hiểu:** Thẻ ngữ nghĩa giúp công cụ tìm kiếm hiểu rõ hơn về nội dung trang web và cải thiện thứ hạng tìm kiếm.

### *2.6.5. Webform*

Webform là một mô hình phát triển web sử dụng các trang web được tạo ra động dựa trên đầu vào của người dùng. Nó thường được sử dụng để tạo các ứng dụng web tương tác, chẳng hạn như biểu mẫu, bảng điều khiển và trang web thương mại điện tử.

**Có hai loại Webform chính:**

* **Webform truyền thống:** Sử dụng các công nghệ web cũ hơn như ASP.NET Web Forms và PHP để tạo các trang web động.
* **Webform hiện đại:** Sử dụng các công nghệ web mới hơn như JavaScript và các framework JavaScript như React, Angular và Vue.js để tạo các ứng dụng web một trang (SPA).

**Lợi ích của Webform:**

* **Dễ sử dụng:** Webform có thể được sử dụng bởi các nhà phát triển web ở mọi trình độ kỹ năng.
* **Linh hoạt:** Webform có thể được sử dụng để tạo nhiều loại ứng dụng web khác nhau.
* **Có thể mở rộng:** Webform có thể được mở rộng để đáp ứng nhu cầu của các ứng dụng web lớn và phức tạp.

**Nhược điểm của Webform:**

* **Có thể phức tạp:** Việc tạo các Webform phức tạp có thể yêu cầu nhiều mã và có thể khó bảo trì.
* **Hiệu suất:** Webform truyền thống có thể không hiệu quả như các ứng dụng web hiện đại.
* **Bảo mật:** Webform có thể dễ bị tấn công bảo mật hơn các ứng dụng web hiện đại.

**Tóm lại:**

Webform là một mô hình phát triển web linh hoạt và có thể mở rộng có thể được sử dụng để tạo nhiều loại ứng dụng web khác nhau. Tuy nhiên, nó có thể phức tạp và không hiệu quả như các ứng dụng web hiện đại.

## **Kết Luận Chương 2**

**HTML** là ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language) được sử dụng để tạo cấu trúc cho các trang web. **HTML5** là phiên bản mới nhất của HTML và cung cấp nhiều tính năng mới và cải tiến so với các phiên bản trước đây.

**Dưới đây là một số điểm chính về HTML và HTML5:**

**HTML:**

* Là ngôn ngữ đánh dấu được sử dụng để tạo cấu trúc cho các trang web.
* Gồm các thẻ HTML để định dạng nội dung trang web.
* Dễ học và sử dụng.
* Là nền tảng cho hầu hết các trang web trên thế giới.

**HTML5:**

* Là phiên bản mới nhất của HTML.
* Cung cấp nhiều tính năng mới và cải tiến so với các phiên bản trước đây.
* Hỗ trợ đa phương tiện, đồ họa vector, biểu mẫu nâng cao, lưu trữ ngoại tuyến, v.v.
* Giúp tạo các trang web web hiện đại, tương tác và năng động hơn.

**Kết luận:**

*HTML5 là lựa chọn tốt nhất cho việc phát triển web hiện đại. Nó cung cấp nhiều tính năng và cải tiến giúp tạo các trang web web hiện đại, tương tác và năng động hơn. Tuy nhiên, HTML vẫn là một ngôn ngữ quan trọng và được sử dụng rộng rãi để tạo các trang web cơ bản.*

# **CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

# **3.1. CSS là gì?**

CSS là viết tắt của Cascading Style Sheets, là ngôn ngữ lập trình được sử dụng để định dạng và tạo kiểu cho các trang web. CSS giúp bạn kiểm soát giao diện của trang web, bao gồm phông chữ, màu sắc, bố cục, v.v.

**Dưới đây là một số tính năng chính của CSS:**

* **Kiểm soát kiểu dáng:** CSS cho phép bạn kiểm soát mọi khía cạnh của giao diện trang web, bao gồm phông chữ, màu sắc, kích thước, bố cục, v.v.
* **Tính linh hoạt:** CSS có thể được sử dụng để định dạng các trang web HTML, XHTML và XML.
* **Dễ học:** CSS tương đối dễ học so với các ngôn ngữ lập trình khác.
* **Tiết kiệm thời gian:** CSS giúp bạn tiết kiệm thời gian bằng cách cho phép bạn định dạng nhiều trang web cùng lúc.

**Ví dụ:**

Bạn có thể sử dụng CSS để thay đổi màu sắc của văn bản trên trang web của mình:

CSS

h1 {

color: red;

}

Mã CSS này sẽ thay đổi màu sắc của tất cả các tiêu đề h1 trên trang web của bạn thành màu đỏ.

**Có ba loại CSS chính:**

* **CSS nội tuyến:** Được nhúng trực tiếp vào thẻ HTML.
* **CSS nhúng:** Được đặt trong thẻ <style> trong phần <head> của tài liệu HTML.
* **CSS liên kết:** Được lưu trữ trong tệp CSS riêng biệt và được liên kết với tài liệu HTML bằng thẻ <link>.

**Kết luận:**

CSS là một ngôn ngữ lập trình quan trọng giúp bạn kiểm soát giao diện của trang web. Nó dễ học và sử dụng, và có thể giúp bạn tạo các trang web đẹp và chuyên nghiệp.

## **3.2. Cú pháp CSS**

Cấu trúc cơ bản của CSS bao gồm 3 phần chính:

**1. Bộ chọn (Selector)**: xác định phần tử HTML nào sẽ được áp dụng kiểu CSS. Ví dụ:

* h1: chọn tất cả thẻ h1
* .button: chọn tất cả phần tử có class "button"
* #header: chọn phần tử có ID "header"

**2. Thuộc tính (Property)**: xác định đặc điểm nào của phần tử sẽ được thay đổi. Ví dụ:

* color: màu sắc
* font-size: kích thước chữ
* background-color: màu nền

**3. Giá trị (Value)**: xác định giá trị cụ thể cho thuộc tính. Ví dụ:

* color: red: đặt màu chữ đỏ
* font-size: 16px: đặt kích thước chữ 16px
* background-color: #ffffff: đặt màu nền trắng

**Ví dụ:**

CSS

h1 {

color: red;

font-size: 16px;

}

.button {

background-color: #000000;

color: #ffffff;

}

**Lưu ý:**

* Mỗi khai báo CSS kết thúc bằng dấu chấm phảy (;).
* Khai báo CSS có thể được viết trên một dòng hoặc nhiều dòng.
* Thứ tự của các khai báo không quan trọng.

## Các loại bộ chọn

Có nhiều loại bộ chọn khác nhau trong CSS, bao gồm:

* **Bộ chọn theo thẻ**: chọn các phần tử dựa trên tên thẻ. Ví dụ: h1, p, div.
* **Bộ chọn theo class**: chọn các phần tử dựa trên class được gán. Ví dụ: .button, .header.
* **Bộ chọn theo ID**: chọn phần tử dựa trên ID được gán. Ví dụ: #header, #footer.
* **Bộ chọn con**: chọn các phần tử con của một phần tử khác. Ví dụ: div p, .container .button.
* **Bộ chọn anh chị em**: chọn các phần tử cùng cấp với một phần tử khác. Ví dụ: h1 + p, .button ~ .link.
* **Bộ chọn giả**: chọn các phần tử dựa trên trạng thái của chúng. Ví dụ: :hover, :active.Một số lưu ý
* Cú pháp CSS có thể thay đổi theo phiên bản CSS.
* Nên sử dụng các công cụ hỗ trợ viết CSS như trình soạn thảo mã có tích hợp chức năng gợi ý cú pháp.
* Việc sử dụng CSS hiệu quả đòi hỏi cần có sự hiểu biết về các thuộc tính và giá trị CSS.

## **3.3. Áp dụng CSS vào trang HTML**

Có 3 cách chính để áp dụng CSS vào trang HTML:

**1. CSS nội tuyến (Inline CSS)**

* Viết CSS trực tiếp vào thuộc tính style của thẻ HTML.
* Cách này chỉ nên sử dụng cho một vài phần tử cụ thể.
* Ví dụ:

HTML

<h1 style="color: red; font-size: 24px;">Tiêu đề</h1>

**2. CSS nhúng (Internal CSS)**

* Viết CSS trong thẻ <style> trong phần <head> của trang HTML.
* Cách này phù hợp cho các trang HTML đơn giản.
* Ví dụ:

HTML

<!DOCTYPE **html**>

<html lang="vi">

<head>

<meta charset="UTF-8">

<title>Trang web của tôi</title>

<style>

h1 {

color: red;

font-size: 24px;

}

</style>

</head>

<body>

<h1>Tiêu đề</h1>

</body>

</html>

**3. CSS liên kết (External CSS)**

* Viết CSS trong một tệp tin .css riêng biệt và liên kết đến tệp tin đó trong phần <head> của trang HTML.
* Cách này phù hợp cho các trang HTML phức tạp hoặc các trang web có nhiều trang.
* Ví dụ:

HTML

<!DOCTYPE **html**>

<html lang="vi">

<head>

<meta charset="UTF-8">

<title>Trang web của tôi</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Tiêu đề</h1>

</body>

</html>

**Lưu ý:**

* Thứ tự ưu tiên của các cách áp dụng CSS là: CSS nội tuyến > CSS nhúng > CSS liên kết.
* Nên sử dụng CSS liên kết để tối ưu hóa hiệu suất trang web.

## **3.4. Selectors**

Trong lĩnh vực thiết kế web với Cascading Style Sheets (CSS), **Selectors (bộ chọn)** đóng vai trò quan trọng như những **cánh tay nhắm trúng** các thành phần (element) cụ thể trên trang HTML để áp dụng kiểu dáng (style) mong muốn.

Nói một cách dễ hiểu, Selectors giống như **công cụ định vị** giúp bạn "chỉ" cho trình duyệt biết nên **áp dụng các thuộc tính CSS** (màu sắc, font chữ, kích thước,...) vào **những phần tử nào** trên trang web.

Có nhiều loại Selectors khác nhau, mỗi loại có cách thức hoạt động riêng, giúp bạn linh hoạt hơn trong việc lựa chọn thành phần để tùy chỉnh:

* **Chọn theo tên thẻ (Element Selectors):** Dựa vào tên thẻ HTML (ví dụ: h1, p, div). Ví dụ: h1 { color: red; } sẽ làm tất cả các tiêu đề <h1> có màu đỏ.
* **Chọn theo class (Class Selectors):** Dùng tên class được gán cho các thành phần (ví dụ: .button, .header). Class được phân biệt bằng dấu chấm (.) phía trước. Ví dụ: .button { background-color: blue; } sẽ đổi màu nền xanh cho tất cả thành phần có class "button".
* **Chọn theo ID (ID Selectors):** Dựa vào ID duy nhất của một thành phần (ví dụ: #main, #footer). ID được phân biệt bằng dấu thăng (#) phía trước. Ví dụ: #main { text-align: center; } sẽ căn giữa nội dung trong thành phần có ID "main".

Ngoài ra, còn có các loại Selectors phức tạp hơn, giúp bạn chọn các thành phần theo mối quan hệ, trạng thái,...:

* **Combinators (Bộ kết hợp):** Giúp kết hợp các Selectors đơn giản để chọn chính xác hơn.
* **Pseudo-classes (Lớp giả):** Chọn thành phần dựa trên trạng thái của chúng (ví dụ: khi rê chuột qua, click chuột,...).
* **Pseudo-elements (Phần tử giả):** Chọn các phần cụ thể của một thành phần (ví dụ: trước nội dung, sau nội dung,...).

Hiểu và sử dụng thành thạo Selectors là **khởi đầu quan trọng** để bạn làm chủ việc tùy chỉnh giao diện trang web, biến những trang HTML thô sơ thành những trang web đẹp mắt và giàu tính tương tác.

### *3.4.1. Universal selector*

Universal selector, hay còn gọi là **bộ chọn phổ quát**, là một loại selector đặc biệt trong CSS có khả năng **chọn tất cả các thành phần HTML** trên trang web. Nó được biểu thị bằng **dấu sao (\*)**.

**Ví dụ:**

CSS

\* {

color: red;

}

Mã CSS trên sẽ **làm cho tất cả các nội dung** trên trang web có màu đỏ, bao gồm cả văn bản, hình ảnh, liên kết, v.v.

**Lưu ý:**

* Universal selector là một công cụ mạnh mẽ, nhưng **nên sử dụng cẩn thận**. Việc sử dụng nó quá nhiều có thể dẫn đến việc khó kiểm soát và thay đổi kiểu dáng trang web.
* Nên sử dụng các loại selector khác để **chọn lọc** các thành phần cụ thể thay vì sử dụng universal selector.

**Sử dụng universal selector hiệu quả:**

* **Kết hợp với các selector khác:** Ví dụ, div \* { color: red; } sẽ chỉ làm cho **văn bản bên trong các thẻ div** có màu đỏ.
* **Thay đổi thuộc tính cụ thể:** Ví dụ, \* { font-size: 16px; } sẽ thay đổi **kích thước font chữ** của tất cả các thành phần.

### *3.4.2. Type selector*

**Type selector**, hay còn gọi là **bộ chọn kiểu**, là một loại selector trong CSS được sử dụng để **chọn tất cả các thành phần HTML có cùng tên thẻ**. Nó được biểu thị bằng **tên thẻ** của thành phần.

**Ví dụ:**

* h1: Chọn tất cả các thẻ <h1>.
* p: Chọn tất cả các thẻ <p>.
* div: Chọn tất cả các thẻ <div>.

**Sử dụng type selector:**

* **Thay đổi kiểu dáng cho tất cả các thành phần cùng loại:** Ví dụ, h1 { color: red; } sẽ làm cho **tất cả các tiêu đề h1** có màu đỏ.
* **Kết hợp với các selector khác:** Ví dụ, div p { font-size: 16px; } sẽ thay đổi **kích thước font chữ** cho **tất cả các đoạn văn bên trong các thẻ div**.

**Lưu ý:**

* Type selector **không phân biệt class, ID hay trạng thái** của thành phần.
* Type selector có thể **kết hợp với các selector khác** để tăng độ chính xác khi chọn lọc thành phần.

**Ví dụ nâng cao:**

* h1:first-child: Chọn **thẻ h1 đầu tiên** trên trang.
* p:hover: Chọn **thẻ p** khi người dùng **di chuột qua**.
* div.container p: Chọn **tất cả các thẻ p** bên trong **thẻ div có class "container"**.

**Tóm lại:**

Type selector là một công cụ đơn giản nhưng hiệu quả để chọn lọc và thay đổi kiểu dáng cho các thành phần HTML cùng loại. Việc sử dụng kết hợp với các selector khác giúp bạn kiểm soát giao diện trang web một cách linh hoạt và chính xác.

### *3.4.3. Identity selector*

Cụm từ "Identity selector" (Bộ chọn danh tính) **không phải là một thuật ngữ chính thức trong CSS** (Cascading Style Sheets). Do đó, nó không có ý nghĩa tiêu chuẩn và cách sử dụng nhất định.

**Các khả năng:**

1. **Sai sót hoặc lỗi đánh máy:** Cụm từ này có thể là **sai sót** hoặc **lỗi đánh máy** cho một thuật ngữ CSS chính xác khác, chẳng hạn như **ID selector** (Bộ chọn ID). ID selector, được ký hiệu bằng #, cho phép bạn chọn một phần tử duy nhất trên trang web dựa trên thuộc tính ID của nó.
2. **Sử dụng tùy chỉnh:** Trong một số trường hợp hiếm gặp, "Identity selector" có thể được sử dụng trong một **kiểu framework hoặc ngữ cảnh tùy chỉnh** cụ thể. Nếu bạn gặp trường hợp cụ thể này, cách tốt nhất để hiểu ý nghĩa của nó là **tham khảo tài liệu của framework** đó hoặc **liên hệ với người tạo ra framework** để làm rõ.

**Tóm lại:**

* **Không sử dụng "Identity selector" trong CSS theo chuẩn.**
* **Kiểm tra xem có phải lỗi đánh máy cho ID selector hay không.**
* **Nếu gặp trong framework/ngữ cảnh tùy chỉnh, tham khảo tài liệu hoặc liên hệ người tạo.**

### *3.4.4. Class selector*

**Class selector (bộ chọn class)** là một loại selector trong CSS cho phép bạn **chọn tất cả các phần tử HTML có cùng tên class**. Class selector được sử dụng bằng cách **chèn dấu chấm (.)** trước tên class.

**Ví dụ:**

* **HTML:**

HTML

<div class="button">Click me</div>

<p class="text">This is some text.</p>

<a class="link" href="#">Visit Google</a>

* **CSS:**

CSS

.button {

background-color: blue;

color: white;

}

.text {

font-size: 16px;

font-weight: bold;

}

.link {

color: red;

text-decoration: underline;

}

**Kết quả:**

* Tất cả các phần tử có class button sẽ có màu nền xanh và chữ màu trắng.
* Tất cả các phần tử có class text sẽ có font chữ 16px, đậm.
* Tất cả các phần tử có class link sẽ có màu đỏ và gạch chân.

**Ưu điểm của class selector:**

* **Sử dụng lại:** Class có thể được sử dụng lại nhiều lần cho nhiều phần tử khác nhau.
* **Tính linh hoạt:** Class cho phép bạn dễ dàng thay đổi kiểu dáng cho nhiều phần tử cùng lúc.
* **Bảo trì:** Class giúp code CSS dễ đọc và dễ bảo trì hơn.

**Lưu ý:**

* Class selector **không phân biệt ID** của phần tử.
* Class selector **không phân biệt trạng thái** của phần tử.

**Tóm lại:**

Class selector là một công cụ mạnh mẽ giúp bạn **kiểm soát kiểu dáng** cho các phần tử HTML một cách hiệu quả. Sử dụng class selector giúp code CSS **gọn gàng, dễ đọc và dễ bảo trì**.

***3.4.5. Descendant selector***

**Descendant selector (bộ chọn hậu duệ)** là một loại selector trong CSS cho phép bạn **chọn tất cả các phần tử con** của một phần tử được chỉ định. Descendant selector được sử dụng bằng cách **cách hai phần tử bởi dấu cách**.

**Ví dụ:**

* **HTML:**

HTML

<div class="container">

<h1>This is a heading</h1>

<p>This is some text.</p>

</div>

* **CSS:**

CSS

.container h1 {

color: red;

}

**Kết quả:**

* Tất cả các thẻ <h1> bên trong thẻ div có class container sẽ có màu đỏ.

**Descendant selector có thể sử dụng để:**

* **Chọn tất cả các phần tử con** của một phần tử, bất kể độ sâu lồng nhau.
* **Chọn các phần tử con cụ thể** dựa trên tên tag.
* **Tạo kiểu cho các phần tử con** dựa trên vị trí của chúng trong cấu trúc HTML.

**Lưu ý:**

* Descendant selector **không phân biệt class** của phần tử con.
* Descendant selector **không phân biệt trạng thái** của phần tử con.

**Tóm lại:**

Descendant selector là một công cụ hữu ích để **chọn lọc và thay đổi kiểu dáng** cho các phần tử con của một phần tử được chỉ định. Sử dụng descendant selector giúp code CSS **gọn gàng và dễ đọc**.

### *3.4.6. Child selector*

**Child selector (bộ chọn con)** là một loại selector trong CSS cho phép bạn **chọn tất cả các phần tử con trực tiếp** của một phần tử được chỉ định. Child selector được sử dụng bằng cách **cách hai phần tử bởi dấu >**.

**Ví dụ:**

* **HTML:**

HTML

<div class="container">

<h1>This is a heading</h1>

<p>This is some text.</p>

</div>

* **CSS:**

CSS

.container > h1 {

color: red;

}

**Kết quả:**

* **Chỉ** thẻ <h1> **trực tiếp** bên trong thẻ div có class container sẽ có màu đỏ.

**Child selector khác với descendant selector ở chỗ:**

* Child selector chỉ chọn **các phần tử con trực tiếp**, descendant selector chọn **tất cả các phần tử con** bất kể độ sâu lồng nhau.

**Child selector có thể sử dụng để:**

* **Chọn các phần tử con trực tiếp** của một phần tử.
* **Tạo kiểu cho các phần tử con trực tiếp** dựa trên vị trí của chúng trong cấu trúc HTML.

**Lưu ý:**

* Child selector **không phân biệt class** của phần tử con.
* Child selector **không phân biệt trạng thái** của phần tử con.

**Tóm lại:**

Child selector là một công cụ hữu ích để **chọn lọc và thay đổi kiểu dáng** cho các phần tử con trực tiếp của một phần tử được chỉ định. Sử dụng child selector giúp code CSS **gọn gàng và dễ đọc**.

**Ví dụ nâng cao:**

* div > p:first-child: Chọn **thẻ p đầu tiên** **trực tiếp** bên trong **thẻ div**.
* h1:hover > p: Chọn **thẻ p** **trực tiếp** bên trong **thẻ h1** khi người dùng **di chuột qua**.
* div.container > p: Chọn **tất cả các thẻ p** **trực tiếp** bên trong **thẻ div có class "container"**.

**Kết hợp child selector với các selector khác giúp bạn tăng độ chính xác khi chọn lọc phần tử.**

### *3.4.7. Adjacent sibling selector*

**Adjacent sibling selector (bộ chọn anh chị em kề)** là một loại selector trong CSS cho phép bạn **chọn một phần tử nằm ngay sau** một phần tử khác **cùng cấp** (cùng nằm trong một phần tử cha). Adjacent sibling selector được sử dụng bằng cách **đặt dấu cộng (+)** giữa tên selector của hai phần tử.

**Ví dụ:**

* **HTML:**

HTML

<div class="container">

<p>This is some text.</p>

<h1 class="title">This is a heading</h1>

</div>

* **CSS:**

CSS

p + h1 {

color: red;

}

**Kết quả:**

* **Chỉ** thẻ <h1> **ngay sau** thẻ <p> **cùng cấp** (bên trong thẻ div có class container) sẽ có màu đỏ. Các thẻ <h1> khác trên trang sẽ không bị ảnh hưởng.

**Lưu ý:**

* Adjacent sibling selector chỉ chọn được **phần tử anh chị em kề** nhau **cùng cấp**.
* Adjacent sibling selector **chỉ chọn được một phần tử**.
* Thứ tự của các selector quan trọng: p + h1 sẽ chọn phần tử <h1> **sau** phần tử <p>, không ngược lại.

**Tóm lại:**

Adjacent sibling selector là một công cụ hữu ích để **chọn lọc và thay đổi kiểu dáng** cho một phần tử anh chị em kề với một phần tử khác. Sử dụng adjacent sibling selector giúp code CSS **linh hoạt** hơn trong việc tạo kiểu dựa trên vị trí tương đối của các phần tử.

***3.4.8. Attribute selector***

**Attribute selector (bộ chọn thuộc tính)** là một loại selector trong CSS cho phép bạn **chọn các phần tử HTML dựa trên các thuộc tính của chúng**. Thuộc tính là một đặc điểm bổ sung của thẻ HTML, cung cấp thêm thông tin về thẻ đó.

**Cấu trúc cơ bản:**

[element\_name][attribute\_name][operator][attribute\_value]

**Ví dụ:**

* **HTML:**

HTML

<img src="image.jpg" alt="Image description">

<a href="https://www.example.com" title="Visit our website">Click here</a>

* **CSS:**

CSS

img[alt] {

border: 1px solid #ddd; /\* Add border to all images with alt attribute \*/

}

a[href^="https://"] {

color: blue; /\* Add blue color to links starting with "https://" \*/

}

**Kết quả:**

* Tất cả các thẻ <img> có **thuộc tính alt** sẽ có đường viền 1px màu xám.
* Tất cả các thẻ <a> có **thuộc tính href bắt đầu bằng "https://""** sẽ có màu xanh.

**Các toán tử thường dùng:**

* =: Kiểm tra giá trị chính xác của thuộc tính.
* ~=: Kiểm tra xem giá trị thuộc tính **chứa** chuỗi cụ thể, ngăn cách bởi dấu cách.
* ^=: Kiểm tra xem giá trị thuộc tính **bắt đầu** với chuỗi cụ thể.
* $=: Kiểm tra xem giá trị thuộc tính **kết thúc** bằng chuỗi cụ thể.
* \*=: Kiểm tra xem giá trị thuộc tính **chứa** chuỗi cụ thể.

**Ưu điểm:**

* **Linh hoạt:** Chọn phần tử dựa trên nhiều thuộc tính khác nhau.
* **Kiểu dáng có điều kiện:** Kiểu dáng phụ thuộc vào giá trị của thuộc tính.

**Lưu ý:**

* **Kiểm tra hỗ trợ trình duyệt:** Một số thuộc tính cũ hơn có thể không được hỗ trợ đầy đủ trên tất cả các trình duyệt.
* **Sử dụng thận trọng:** Quá nhiều attribute selector có thể khiến code CSS khó đọc và bảo trì.

**Tóm lại:**

Attribute selector là một công cụ mạnh mẽ cho phép bạn **tạo kiểu dáng có mục tiêu** cho các phần tử HTML dựa trên các thuộc tính của chúng. Sử dụng attribute selector một cách hợp lý giúp code CSS **linh hoạt, chính xác và dễ bảo trì**.

### *3.4.9. Pseudo class selector*

**Pseudo class selector (bộ chọn trạng thái giả)** là một loại selector đặc biệt trong CSS dùng để **chú thích cho các trạng thái tạm thời** của một phần tử HTML. Khác với các class selector bình thường, **pseudo class không tương ứng với bất kỳ thuộc tính HTML nào** và **không thể được thêm vào như một class** cho các phần tử khác.

**Ví dụ:**

* **HTML:**

HTML

<a href="#">This is a link</a>

<button>Click me</button>

* **CSS:**

CSS

a:hover {

text-decoration: underline; /\* Underline links on hover \*/

}

button:active {

background-color: #ddd; /\* Change button background on click \*/

}

**Kết quả:**

* Khi người dùng **di chuột qua** thẻ <a>, nó sẽ được **gạch chân**.
* Khi người dùng **nhấn vào** nút <button>, nền của nó sẽ **chuyển sang màu xám nhạt**.

**Các Pseudo Class Selector phổ biến:**

* :hover: Khi người dùng di chuột qua phần tử.
* :active: Khi người dùng kích hoạt phần tử (nhấn chuột, chạm, v.v.).
* :focus: Khi phần tử nhận được tiêu điểm (ví dụ, khi người dùng tab đến một input field).
* :link: Khi phần tử là một liên kết chưa được truy cập.
* :visited: Khi phần tử là một liên kết đã được truy cập.
* :first-child: Phần tử con đầu tiên của một phần tử khác.
* :last-child: Phần tử con cuối cùng của một phần tử khác.
* :nth-child(n): Phần tử con thứ n của một phần tử khác (n là một số nguyên).

**Ưu điểm:**

* **Kiểu dáng linh hoạt:** Tạo hiệu ứng phong phú dựa trên hành vi người dùng.
* **Cải thiện trải nghiệm người dùng:** Cung cấp phản hồi trực quan cho người dùng.
* **Giảm code lặp lại:** Không cần tạo class riêng cho các trạng thái tạm thời.

**Lưu ý:**

* **Không lạm dụng:** Quá nhiều pseudo class selector có thể khiến code CSS khó đọc và bảo trì.
* **Kiểm tra hỗ trợ trình duyệt:** Một số pseudo class selector cũ hơn có thể không được hỗ trợ đầy đủ trên tất cả các trình duyệt.

### *3.4.10. Group selector*

Group selector (bộ chọn nhóm) là một loại selector trong CSS **chưa được triển khai** và **không được khuyến khích sử dụng**. Nó được đề xuất trong bản dự thảo CSS Selectors Level 4 nhưng chưa được áp dụng rộng rãi.

**Cấu trúc:**

[selector1][selector2]

**Ví dụ:**

* **HTML:**

HTML

<div class="container">

<p>This is some text.</p>

<h1 class="title">This is a heading</h1>

</div>

* **CSS:**

CSS

/\* Group selector không được hỗ trợ rộng rãi \*/

p[class="text"] h1[class="title"] {

color: red; /\* Chọn h1 có class "title" bên trong p có class "text" \*/

}

**Kết quả:**

**Theo dự thảo CSS Selectors Level 4:**

* Mã CSS trên sẽ chọn **thẻ h1 có class "title"** **bên trong** thẻ **p có class "text"**.

**Tuy nhiên:**

* Hầu hết các trình duyệt hiện nay **không hỗ trợ** group selector.
* Sử dụng **descendant selector** (p h1) hoặc **child selector** (p > h1) thay thế cho group selector.

**Lưu ý:**

* **Tránh sử dụng group selector** trong code CSS của bạn.
* Sử dụng các selector khác thay thế để đảm bảo khả năng tương thích với nhiều trình duyệt.

## **3.5. Đơn vị đo lường CSS**

Theo những tìm hiểu của mình và trong quá trình làm việc thực tế ở các dự án thì có các đơn vị tính trong CSS: px, pt, in, pc, mm, cm, em, ex, ch, rem, vw, vh, vmin, vmax, %...

#### **Quy định về đơn vị trong CSS:**

* Giá trị số rồi đến đơn vị, VD: 10px, 2em, 50%, ...
* Tuyệt đối không có dấu cách giữa số và đơn vị tính.
* Nếu giá trị số là 0, có thể bỏ qua đơn vị tính.
* Một số đơn vị cho phép sử dụng số âm. VD: padding-top:-20px; margin-top: -20%; Nhưng phần lớn các đơn vị khác là số dương.

Trong khuân khổ bài chia sẻ này mình sẽ cùng các bạn tìm hiểu những đơn vị tính thường sử dụng trong CSS. Mặc dù sẽ không đề cập hết nhưng không có nghĩa là những đơn vị đó sẽ không được sử dụng, các bạn nên tìm hiểu qua để biết và lựa chọn đơn vị tính phù hợp.

Trong CSS có hai loại đơn vị là **Đơn vị tương đối** (Relative Units) và **Đơn vị tuyệt đối** (Absolute Units). Trong đó:

### 1. Đơn vị tương đối - Relative Units

**Đơn vị tương đối** là đơn vị được tính một cách tương đối so với phần tử mẹ hoặc các phần tử khác. Đơn vị tương đối cơ động và hoạt động tốt hơn cho những thiết bị, những màn hình có kích thước và độ phân giải khác nhau.

#### **Một số đơn vị tương đối chính được sử dụng trong CSS:**

* **% (percentages)**: Đơn vị tương đối về kích thước của phần tử con so với phần tử mẹ (parrent). Ví dụ: Nếu phần tử mẹ có thuộc tính width:500px, phần tử con có thuộc tính width:50% thì có nghĩa là phần tử con có thuộc tính width:250px
* **em**: Là đơn vị tương đối dựa vào thuộc tính font-size của phần tử mẹ. Ví dụ: Nếu phần tử mẹ có thuộc tính font-size:16px, phần tử con có thuộc tính font-size:2em thì có nghĩa là phần tử con có thuộc tính font-size:32px
* **rem(root em)**: Là đơn vị tương đối dựa vào thuộc tính font-size của phần html (root element). Ví dụ: Nếu phần tử html có thuộc tính font-size:12px phần tử mẹ có thuộc tính font-size:16px, phần tử con có thuộc tính font-size:2rem thì có nghĩa là phần tử con có thuộc tính font-size:24px; Tức là đơn vị rem không phụ thuộc vào phần tử mẹ.

### 2. Đơn vị tuyệt đối - Absolute Units

Khác với đơn vị tương đối, như tên gọi, đơn vị tuyệt đối sẽ có những giá trị cố định không thay đổi. Đơn vị tuyệt đối không được khuyến khích sử dụng cho màn hình, vì có rất nhiều các kích thước màn hình khác nhau. Nó chỉ nên dùng cho những trường hợp biết chắc kích thước của giao diện, như xuất ra máy in chẳng hạn.

#### **Một số đơn vị tuyệt đối được sử dụng trong CSS:**

* px \*: pixels (1px = 1/96th of 1in)
* pt: points (1pt = 1/72 of 1in)
* pc: picas (1pc = 12 pt)
* cm: centimeters
* mm: millimeters
* \* Pixels (px) là khác nhau giữa các thiết bị. Với các thiết bị có độ phân giải thấp (low-dpi devices), 1px là một chấm điểm hiển thị trên màn hình của thiết bị. Với máy in và các thiết bị có độ phân giải cao, 1px lại là nhiều chấm điểm hiển thị trên màn hình của thiết bị.in: inches (1in = 96px = 2.54cm)

**3.6. Kế thừa thuộc tính**

* Thuộc tính kế thừa (inherited properties): là những thuộc tính ảnh hưởng tới thuộc tính con của nó. Tất cả các thuộc tính ảnh hưởng tới text đều là thuộc tính kế thừa. Ví dụ, nếu chúng ta chọn font-size cho một phần tử HTML, giá trị đó sẽ được áp dụng cho tất cả các phần tử con của nó

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

* - Thuộc tính định dạng font chữ, văn bản (font, text)

## **Thiết lập KÍCH CỠ CHỮ**

* Thuộc tính font-size dùng để thiết lập kích cỡ chữ của văn bản
* Ví dụ:
* p {
* font-size:40px;
* }
* - Thuộc tính định dạng nền (background)
* Với thuộc tính và giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **giá trị** | **Ví dụ** | **Mô tả** |
| [background-attachment](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr01) | fixed  scroll | background-attachment :fixed; | Xác định thành phần nền được cố định hoặc cuộn so với trang.  Được sử dụng kèm với giá trị background-image |
| [background-color](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr02) | mã màu  tên màu  Giá trị rgb | background-color :#ff0000;  background-color: red;  background-color: rgb(255,0,0); | Xác định màu nền cho thành phần. |
| [background-image](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr03) | url(đường dẫn hình) | background-image: url(ico\_arrow.gif); | Xác định hình ảnh nền cho thành phần. |
| [background-position](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr04) | left  right  top  bottom  px  % | background-position: left top; | Xác định vị trí hình ảnh nền cho thành phần.  Được sử dụng kèm với giá trị background-image. |
| [background-repeat](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr05) | repeat-x  repeat-y  repeat  no-repeat | background-repeat: repeat-x; | Xác định hình ảnh nền được lặp như thế nào. |
| [background](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr06) | Một hoặc nhiều giá trị của các thuộc tính trên | background: url(ico\_arrow.gif) repeat-x left top; | Có giá trị riêng lẻ hoặc tổng hợp các thuộc tính trên. |

* - Mô hình hộp (box model)

Một phần tử HTML được coi là một Box.

Mô hình hộp (Box model) có nghĩa là một phần tử HTML sẽ bao gồm các thành phần: margin, border, padding, và content (nội dung của phần tử).

Mô hình hộp (Box model) cho phép chúng ta thêm bo viền xung quanh phần tử , và xác định khoảng cách giữa các phần tử.

- Các thuộc tính định margin, padding, border

### Các thuộc tính định margin

### Cấu trúc

tag {

margin: giá trị;

}

Với giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **giá trị** | **Ví dụ** | **Mô tả** |
| [margin](https://hocwebchuan.com/reference/cssSection/example/ex_margin.php#pr05) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  auto  inherit | margin: 5px 10px 3px 20px; | Đây là cách viết ngắn gọn các thuộc tính bên dưới, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [margin-top](https://hocwebchuan.com/reference/cssSection/example/ex_margin.php#pr01) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  auto  inherit | margin-top: 5px; | Canh lề bên trên cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [margin-right](https://hocwebchuan.com/reference/cssSection/example/ex_margin.php#pr02) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  auto  inherit | margin-right: 5px; | Canh lề bên phải cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [margin-bottom](https://hocwebchuan.com/reference/cssSection/example/ex_margin.php#pr03) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  auto  inherit | margin-bottom: 5px; | Canh lề bên dưới cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [margin-left](https://hocwebchuan.com/reference/cssSection/example/ex_margin.php#pr04) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  auto  inherit | margin-left: 5px; | Canh lề bên trái cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |

Các thuộc tính định padding

### Cấu trúc

tag {

padding: giá trị;

}

Với giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **giá trị** | **Ví dụ** | **Mô tả** |
| [padding](https://hocwebchuan.com/reference/cssSection/example/ex_padding.php#pr05) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  inherit | padding: 5px 10px 3px 20px; | Đây là cách viết ngắn gọn các thuộc tính bên dưới, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [padding-top](https://hocwebchuan.com/reference/cssSection/example/ex_padding.php#pr01) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  inherit | padding-top: 5px; | Thêm vào khoảng không bên trên cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [padding-right](https://hocwebchuan.com/reference/cssSection/example/ex_padding.php#pr02) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  inherit | padding-right: 5px; | Thêm vào khoảng không bên phải cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [padding-bottom](https://hocwebchuan.com/reference/cssSection/example/ex_padding.php#pr03) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  inherit | padding-bottom: 5px; | Thêm vào khoảng không bên dưới cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |
| [padding-left](https://hocwebchuan.com/reference/cssSection/example/ex_padding.php#pr04) | [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php)  inherit | padding-left: 5px; | Thêm vào khoảng không bên trái cho thành phần, [đơn vị](https://hocwebchuan.com/reference/cssSection/css_units.php) có thể là px, em, %, ... |

Các thuộc tính định border

### Cấu trúc

tag {

Thuộc tính: giá trị;

}

Với thuộc tính và giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **giá trị** | **Ví dụ** | **Mô tả** |
| Thuộc tính định dạng | | | |
| [border-color](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr01) | mã màu  tên màu  Giá trị rgb | border-color: #ff0000;  border-color: red;  border-color: rgb(255,0,0); | Xác định màu sắc của đường viền. |
| [border-style](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr02) | [kiểu border](https://hocwebchuan.com/reference/cssSection/pr_bdr.php#bdr01) | border-style: solid; | Xác định hình dạng của đường viền. |
| [border-width](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr03) | px  thin  medium  thick | border-width: 5px; | Xác định bề dày của đường viền. |
| Thuộc tính vị trí | | | |
| [border-top](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr04) | Một hoặc nhiều giá trị của các thuộc tính định dạng. | border-top: #ff0000 solid 2px; | Xác định đường viền bên trên. |
| [border-right](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr05) | Một hoặc nhiều giá trị của các thuộc tính định dạng. | border-right: #ff0000 solid 2px; | Xác định đường viền bên phải. |
| [border-bottom](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr06) | Một hoặc nhiều giá trị của các thuộc tính định dạng. | border-bottom: #ff0000 solid 2px; | Xác định đường viền bên dưới. |
| [border-left](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr07) | Một hoặc nhiều giá trị của các thuộc tính định dạng. | border-left: #ff0000 solid 2px; | Xác định đường viền bên trái. |
| [border](https://hocwebchuan.com/reference/cssSection/example/ex_bdr.php#pr08) | Một hoặc nhiều giá trị của các thuộc tính định dạng. | border: #ff0000 solid 2px; | Xác định tất cả các đường viền xung quanh. |

- Thuộc tính định dạng cách hiển thị (display)

## Display CSS là gì?

Display CSS là thuộc tính quan trọng nhất của CSS được sử dụng để kiểm soát bố cục của phần tử. Nó chỉ định cách phần tử được hiển thị.

Display CSS cho phép bạn xác định sự xuất hiện của các phần tử trang một cách khác đi so với cài đặt mặc định của chúng. Đây là một khả năng mạnh mẽ và hầu hết các trang web hiện đại – bao gồm cả những trang web được xây dựng bằng Bootstrap CSS- đều tận dụng nó ở một mức độ nào đó.

Mọi phần tử HTML được đại diện bởi một box chứa nội dung và xác định khoảng cách xung quanh nội dung. Thuộc tính display CSS chỉ định cách hộp này xuất hiện trên trang web so với các phần tử khác, cũng như hành vi của các phần tử con của nó (tức là các phần tử bên trong nó).

Mọi phần tử đều có giá trị hiển thị mặc định theo bản chất của nó. Mọi phần tử trên trang web là một hình hộp chữ nhật và thuộc tính CSS xác định hành vi của hình hộp chữ nhật đó.

Thuộc tính CSS hiển thị đặt xem một phần tử được coi là khối hay phần tử nội dòng và bố cục được sử dụng cho các phần tử con của nó, chẳng hạn như bố cục luồng, lưới hoặc linh hoạt.

Về mặt hình thức, thuộc tính display thiết lập các kiểu hiển thị bên trong và bên ngoài của một phần tử. Loại bên ngoài thiết lập sự tham gia của một phần tử vào bố cục luồng; loại bên trong thiết lập bố cục của trẻ em. Một số giá trị của màn hình được xác định đầy đủ trong các thông số kỹ thuật riêng của chúng; ví dụ, chi tiết về những gì sẽ xảy ra khi khai báo display: flex được xác định trong đặc tả Mô hình hộp linh hoạt CSS.

Trong CSS có hai “cấp độ” mà box này có thể áp dụng: block và inline:

Các phần tử cấp block tồn tại trên dòng riêng của chúng và kéo dài toàn bộ chiều rộng của trang (hoặc chiều rộng được chỉ định của block đó). <div> và <p> là các ví dụ về các phần tử cấp block.

Các phần tử cấp inline thì ngược lại, chúng có thể tồn tại trên cùng một dòng. <span>, <b> và <a> là các phần tử cấp inline.

- Thuộc tính xác định vị trí (position)

### Cấu trúc

tag {

position: giá trị;

}

Với giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **giá trị** | **Ví dụ** | **Mô tả** |
| [position](https://hocwebchuan.com/reference/cssSection/example/ex_position.php#pr01) | static | position: static; | Thành phần sẽ nằm theo thứ tự trong văn bản, đây là dạng mặc định. |
| [relative](https://hocwebchuan.com/reference/cssSection/example/ex_position.php#pr02) | position: relative; | Định vị trí tuyệt đối cho thành phần. |
| [absolute](https://hocwebchuan.com/reference/cssSection/example/ex_position.php#pr03) | position: absolute; | Định vị trí tuyệt đối cho thành phần theo thành phần bao ngoài (thành phần định vị trí tương đối position: relative;) hoặc theo cửa sổ trình duyệt. |
| [fixed](https://hocwebchuan.com/reference/cssSection/example/ex_position.php#pr04) | position: fixed; | Định vị trí tương đối cho thành phần theo cửa sổ trình duyệt. |
| inherit | position: inherit; | Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài). |

## **3.8. Float & Clear**

## **Thuộc tính Float CSS là gì?**

Thuộc tính Float CSS là một thuộc tính được sử dụng để chuyển một phần tử sang góc trái hoặc góc bên phải của không gian bao quanh nó.

Đây là một thuộc tính cơ bản và rất cần thiết khi định dạng lại bố cục trang. Trên thực tế, tất cả các phần tử HTML ở dạng mặc định và không có float.

## **Thuộc tính Float bao gồm các giá trị nào?**

Thuộc tính Float có thể bao gồm các giá trị dưới đây:

* Left: Cố định phần tử HTML về phía bên trái.
* Right: Cố định phần tử HTML về bên phải.
* None: Phần tử nằm ở vị trí của nó, trạng thái mặc định.
* Inherit: Phần tử kế thừa giá trị float của phần tử cha.

## **Phân biệt thuộc tính Float CSS và Clear CSS**

Thuộc tính Clear là một thuộc tính ngược lại hoàn toàn với Float. Thuộc tính Clear CSS có tác dụng ngăn chặn các thành phần A chiếm không gian của thành phần B. Đôi khi người ta sử dụng thuộc tính Clear khi không muốn Float CSS ở một tình huống nào đó.

Nói theo cách dễ hiểu hơn thì thuộc tính Clear dùng để giải quyết vấn đề trong phần lưu ý phía bên trên mà chúng tôi đã nêu.

Cách sử dụng phổ biến nhất để dùng thuộc tính Clear chính là dùng sau khi sử dụng thuộc tính Float. Nếu một phần tử được float sang bên trái thì bạn nên Clear về bên phải. Phần tử HTML sẽ tiếp tục nổi nhưng phần tử bị xóa sẽ không bị mất mà vẫn xuất hiện.

Thuộc tính Clear có những giá trị sau tương tự như Float CSS:

* Left/Right: tràn phần tử về bên trái/phải.
* None: Cho phép tràn lên cả hai phía của phần tử.
* Both: Không cho phép tràn về phía nào.
* Inherit: Kế thừa từ phẩn tử cha.

## **3.9. Flex**

### **Flex css là gì?**

Flex css là một kiểu dàn trang mà khi sử dụng nó sẽ tự cân đối kích thước của các thành phần để hiển thị trong  mọi thiết bị. Điều đó có nghĩa là bạn không cần ngồi căn chỉnh kích thước phần tử, không cần ngồi float chỉ cần thiết lập nó theo chiều ngang dọc là lúc đó các phần tử bên trong sẽ tự hiển thị theo mong muốn.

Nguyên lý hoạt động của flex css dựa trên hình ảnh 1 chiếc hộp chứa nhiều thành phần nhỏ bên trong do đó việc căn chỉnh sẽ dễ dàng hơn nhiều.Hiện nay, flex css thường được sử dụng để thiết lập bố cục phạm vi nhỏ còn các bố cục có phạm vi rộng thì thường được sử dụng kiểu dàn trang thông thường.

### **Cấu trúc của flex css**

Trong flex css có hai thành phần chính quan trọng là container và item:

* ***Container*** là phần tử lớn bao quanh toàn bộ các phần tử con, container sẽ thiết lập cách sắp xếp theo chiều dọc hoặc chiều ngang, các phần tử con bên trong sẽ hiển thị theo thiếp lập đó.
* ***Item*** là các phần tử con, bạn có thể thiết lập số cột cũng như thứ tự hiển thị của các item.

Ngoài ra còn có các thành phần khác như :

* ***Main start, main end, cross start, cross end*** : Điểm bắt đầu và kết thúc của container theo main axis được. gọi lần lượt là main start, main end ; đối với cross axis cũng tương tự là cross start, cross end.
* ***Man axis*** hướng của item hiển thị được mặc định chạy từ trái qua phải.
* ***Cross axis*** Trục vuông góc với main axis, chạy theo chiều trên xuống dưới.
* ***Cross size*** là kích thước của item dựa theo trục cross axis.
* ***Main size*** là kích thước của item dựa theo trục main axis.

## **3.10. Grid**

### **Grid CSS là gì?**

Grid layout là một hệ thống layout 2 chiều (x,y) được dùng trong thiết kế UI. Theo định nghĩa từ Mozilla, grid (lưới) là một tổ hợp của những đường ngang và dọc cắt nhau – một nhóm xác định các cột và nhóm kia xác định các hàng. Các phần tử có thể được đặt lên lưới, dựa vào các đường hàng và cột này.

### **Trình duyệt tương thích**

Hầu hết các trình duyệt hiện đại ngày nay đều có hỗ trợ Grid CSS. Vậy nên việc áp dụng grid khi thiết kế sẽ sớm trở thành xu thế chung.

## **3.11. CSS3**

- *CSS3 là gì?*

CSS3 là bản nâng cấp mới nhất của CSS với mục đích mở rộng CSS2. 1 . CSS3 bổ sung, mang đến nhiều tính năng mới như bo tròn góc, tạo độ bóng, gradient, transitions hay animations cũng như các bố cục mới như multiple-column, flexbox hay grid layout

- *CSS3 Selectors*

CSS selector là một thành phần của bộ quy tắc CSS, được sử dụng để xác định các phần tử bạn muốn tạo style bằng CSS. CSS selector cho trình duyệt biết các phần tử HTML nào để áp dụng các giá trị thuộc tính CSS. Nó lựa chọn các phần tử HTML dựa trên id, class, loại, thuộc tính của chúng

- *CSS3 Rounded Corners*

Rounded Corner trong CSS là việc sử dụng thuộc tính border-radius để bo tròn góc cạnh các phần tử.

- *CSS3 Border Images*

Border Image trong CSS là việc thêm hình ảnh xuất hiện trên đường viền cho các phần tử. Bạn không cần sử dụng bất cứ HTML code nào để gọi Border Image

- *CSS3 Backgrounds*

Background-color

Thuộc tính background-color dùng để đặt màu nền cho một thành phần. Nó chấp nhận tất cả giá trị các mã màu hoặc thuộc tính transparent

- *CSS Gradients*

CSS gradient là sự chuyển đổi mượt mà giữa hai hoặc nhiều màu sắc khác nhau. CSS gradient cung cấp khả năng kiểm soát và hiệu quả tốt hơn so với việc sử dụng tệp hình ảnh màu gradient thực tế (mà bạn có thể tạo ra với một công cụ như Adobe Illustrator)

*- CSS3 Shadows*

Box-shadow là thuộc tính mới trong CSS, nằm trong phiên bản CSS3. Tính năng này có tác dụng tạo bóng cho hộp. Thuộc tính Shadow trong CSS được sử dụng để thiết lập bóng cho các phần tử thành phần trong website

- *CSS3 Fonts*

CSS3 mới nhất cũng có các tính năng tạo kiểu đường viền mới như border-radius, image-slice, image-source và các giá trị cho “width stretch”,…

- *CSS3 2D Transforms*

2D Transform CSS là những thuộc tính được dùng để xử lý các hiệu ứng di chuyển 2D của các phần tử trên website. Để tạo một transform 2D bạn sẽ sử dụng cú pháp sau đây: transform: value(argument); Các giá trị thường được sử dụng trong 2D Transform gồm có: translate(), rotate(), scale(), skewX(), skewY(), matrix().

- *CSS3 3D Transforms*

3D Transform là gì? 3D Transform là những thuộc tính dùng để xử lý hiệu ứng di chuyển 3D cho các phần tử theo 3 trục: x, y và z.

- *CSS3 Transitions*

Trong CSS3 có 2 thuộc tính quan trọng đó là Transition và Animation. Đây là một kỹ thuật rất hay giúp làm chuyển động cho phần tử HTML mà không cần Javascript hay Flash. Trong bài viết này mình sẽ giới thiệu với các bạn thuộc tính Animation và ứng dụng đơn giản của nó.

- CSS3 Multiple Columns

Multi-column là gì? Ý tưởng cơ bản của Multi-column là bạn có thể lấy một đoạn nội dung và chuyển nó thành nhiều cột như trong một tờ báo. Bạn làm điều này bằng cách sử dụng một trong hai thuộc tính. Thuộc tính column-count xác định số cột mà bạn muốn nội dung hiển thị.

- CSS3 Animations

CSS animation là công nghệ được giới thiệu trong phiên bản CSS3. Nó cho phép chúng ta tạo hiệu ứng chuyển động mà không phải sử dụng Javascript hay Flash. Trong bài học này bạn sẽ tìm hiểu cách tạo chuyển động thẳng và quay vòng đồng thời cho một thẻ div nhờ sử dụng HTML5 và CSS3.

## **3.12. SCSS**

**SCSS** sử dụng cú pháp giống với [Ruby](https://topdev.vn/blog/ruby-on-rails-la-gi/) (vì đơn giản nó được thiết kế bởi các lập trình viên Ruby)**.**Có phần mở rộng là **.scss**, SCSS ra đời sau SASS và có cú pháp viết tương tự như cách viết CSS. Cú pháp này được tạo ra nhằm thu hẹp khoảng cách giữa SASS và CSS bằng cách mang lại một thứ gì đó thân thiện với CSS.

## **3.13. SASS**

Sass là chữ viết tắt của Syntactically Awesome Style Sheets, chương trình tiền xử lý bằng ngôn ngữ kịch bản (Preprocessor Scripting Language ), sẽ được biên dịch thành CSS. Nghĩa là, mình sẽ làm style bằng SASS, rồi SASS sẽ render việc mình làm thành file CSS.

**SASS** bản thân có hai kiểu viết khác nhau, một kiểu như là HAML, Pug – Sử dụng indent (cách thụt đầu dòng) để phân tách các khối code , sử dụng xuống dòng để phân biệt rules , có phần mở rộng là **.sass.**

**Kết luận Chương 3**

* CSS và CSS3 là hai công cụ thiết yếu trong thiết kế web.
* CSS3 cung cấp nhiều tính năng mới mạnh mẽ giúp nâng cao khả năng thiết kế và trình bày trang web.
* Việc sử dụng CSS3 ngày càng phổ biến trong phát triển web hiện nay.

### CHƯƠNG 4: THIẾT KẾ WEBISTE, GIỚI THIỆU VỀ DU LỊCH VÙNG TÂY BẮC

## **4.1. Ý tưởng của Website**

**-** Vì một phần rất đam mê về mảng du lịch và cũng muốn tìm hiểu văn hoá vùng cao nên em quyết định chọn đề tài thiết kế website, giới thiệu về du lịch vùng Tây Bắc để làm bài báo cáo trong học phần này.

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

- Bố cục trang web đóng vai trò quan trọng trong việc thu hút người dùng và mang lại trải nghiệm tốt nhất cho họ. Một bố cục hợp lý sẽ giúp người dùng dễ dàng tìm kiếm thông tin, tăng khả năng tương tác và cải thiện hiệu quả hoạt động của trang web.

**Dưới đây là một số nguyên tắc cơ bản để xây dựng bố cục trang web:**

**1. Xác định mục tiêu của trang web:**

* Xác định mục tiêu chính của trang web là gì (thông tin, bán hàng, giới thiệu sản phẩm, v.v.)
* Xác định đối tượng mục tiêu của trang web

**2. Lên kế hoạch bố cục:**

* Chia trang web thành các khu vực chính như header, menu, nội dung chính, footer
* Xác định vị trí và kích thước của các khu vực này
* Lựa chọn bố cục phù hợp với mục tiêu và đối tượng mục tiêu của trang web

**3. Sử dụng các yếu tố thiết kế:**

* Sử dụng màu sắc, hình ảnh, font chữ phù hợp với mục tiêu và đối tượng mục tiêu của trang web
* Đảm bảo tính nhất quán trong thiết kế
* Sử dụng các yếu tố thiết kế để thu hút sự chú ý của người dùng vào những nội dung quan trọng

**4. Kiểm tra và tối ưu hóa:**

* Mở trang web trên nhiều thiết bị khác nhau để đảm bảo bố cục hiển thị tốt trên mọi màn hình
* Tiến hành thử nghiệm A/B để đánh giá hiệu quả của các bố cục khác nhau
* Thu thập phản hồi từ người dùng và cải thiện bố cục dựa trên phản hồi đó

**Dưới đây là một số loại bố cục trang web phổ biến:**

* **Bố cục một cột:** Bố cục đơn giản, phù hợp cho các trang web có nội dung ít.
* **Bố cục hai cột:** Bố cục phổ biến, phù hợp cho các trang web có nhiều nội dung.
* **Bố cục ba cột:** Bố cục phù hợp cho các trang web có nhiều nội dung và cần phân chia rõ ràng các khu vực.
* **Bố cục lưới:** Bố cục linh hoạt, phù hợp cho các trang web có nhiều nội dung đa dạng.

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

**HTML** (HyperText Markup Language) và **CSS** (Cascading Style Sheets) là hai ngôn ngữ lập trình cơ bản để thiết kế trang web.

**HTML** được sử dụng để tạo cấu trúc và nội dung cho trang web, bao gồm các thẻ như:

* <html>: Khai báo đầu trang HTML
* <head>: Chứa thông tin về trang web như tiêu đề, mô tả, liên kết đến các tập tin CSS
* <body>: Chứa nội dung chính của trang web như văn bản, hình ảnh, video
* <h1>, <h2>, <h3>, ...: Định dạng tiêu đề
* <p>: Định dạng đoạn văn
* <img>: Chèn hình ảnh
* <a>: Tạo liên kết

**CSS** được sử dụng để định dạng giao diện cho trang web, bao gồm các thuộc tính như:

* color: Màu sắc
* font-family: Kiểu chữ
* font-size: Cỡ chữ
* text-align: Căn lề văn bản
* margin: Khoảng cách lề
* padding: Khoảng cách đệm
* width: Chiều rộng
* height: Chiều cao

**Để thiết kế trang web bằng HTML và CSS, bạn cần thực hiện các bước sau:**

1. **Xác định bố cục trang web**: Xác định các khu vực chính trên trang web như header, menu, nội dung chính, footer.
2. **Viết mã HTML**: Sử dụng các thẻ HTML để tạo cấu trúc và nội dung cho trang web.
3. **Viết mã CSS**: Sử dụng các thuộc tính CSS để định dạng giao diện cho trang web.
4. **Kiểm tra và chỉnh sửa**: Mở trang web trên trình duyệt web để kiểm tra kết quả và chỉnh sửa mã HTML và CSS cho đến khi đạt được giao diện mong muốn.

## **4.5. Kết quả đạt được**

**- Kết quả đạt được sau khi thiết kế trang web bằng HTML và CSS có thể khác nhau tùy thuộc vào mục tiêu và yêu cầu của trang web.**

**1. Giao diện trang web đẹp mắt và thu hút:**

* Sử dụng HTML và CSS để tạo bố cục trang web hợp lý, khoa học.
* Sử dụng màu sắc, hình ảnh, font chữ phù hợp với mục tiêu và đối tượng mục tiêu của trang web.
* Đảm bảo tính nhất quán trong thiết kế.
* Sử dụng các yếu tố thiết kế để thu hút sự chú ý của người dùng vào những nội dung quan trọng.

**2. Trải nghiệm người dùng tốt:**

* Trang web dễ dàng truy cập và sử dụng trên mọi thiết bị.
* Tốc độ tải trang nhanh chóng.
* Nội dung trang web được trình bày rõ ràng, dễ đọc.
* Người dùng có thể dễ dàng tìm kiếm thông tin cần thiết.

**3. Tăng hiệu quả hoạt động của trang web:**

* Tăng số lượng người truy cập trang web.
* Tăng thời gian lưu lại trang web của người dùng.
* Tăng tỷ lệ chuyển đổi (conversion rate).
* Góp phần nâng cao thương hiệu và uy tín của doanh nghiệp.

**Dưới đây là một số ví dụ cụ thể về kết quả đạt được sau khi thiết kế trang web bằng HTML và CSS:**

* **Trang web bán hàng:** Tăng doanh thu bán hàng, thu hút thêm khách hàng tiềm năng.
* **Trang web giới thiệu công ty:** Nâng cao hình ảnh và uy tín của công ty, thu hút nhân tài.
* **Trang web tin tức:** Tăng lượng truy cập, thu hút độc giả.
* **Trang web blog:** Tăng lượng người theo dõi, xây dựng cộng đồng.

## **Kết luận chương 4**

-Sau khi học xong môn Thiết kế web và triển khai hệ thống phần mềm dưới sự chỉ dạy của thầy ThS. Lê Văn Phong, em đã tạo thành công trang web giới thiệu về du lịch vùng Tây Bắc.

**KẾT LUẬN**

## Qua bài báo cáo trên, ta thấy được Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực quan trọng trong phát triển phần mềm. Hai lĩnh vực này có mối liên hệ mật thiết với nhau và ảnh hưởng trực tiếp đến chất lượng và hiệu quả của sản phẩm phần mềm.

Một số ưu điểm của thiết kế web có thể kể đến như tạo ra giao diện người dùng đẹp mắt và thu hút. Nâng cao tair nghiệm người dùng và sự tương tác với trang web. Góp phần tăng hiệu quả kinh doanh và quảng bá thương hiệu. Bên cạnh đó nhược điểm là yêu cầu kiến thức chuyên môn về thiết kế và lập trình web cũng như tốn thời gian và chi phí để thiết kế và phát triển trang web. Tương tự đối với triển khai hệ thống phần mềm có ưu điểm là tự động hoá các quy trình thủ công và nâng cao hiệu quả công việc. Giúp quản lý dữ liệu và thông tin hiệu quả hơn. Ngược lại nó có nhược điểm là cần có đội ngũ nhân viên chuyên môn để triển khai và vận hành hệ thống và có thể gặp rủi ro trong quá trình triển khai và vận hành hệ thống.

Tổng thể môn học Thiết kế web và triển khai hệ thống phần mềm giúp chúng ta hiểu rõ hơn về công nghệ web. Qua đó chúng ta thấy được vai trò của công nghệ thông tin trong đời sống. Nó đóng một vai trò không thể thiếu trong cuộc sống của chúng ta hiện nay.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
3. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.

[4]. <https://www.w3schools.com/html/default.asp>

[5]. <https://developer.mozilla.org/en-US/docs/Web/HTML>

[6]. <https://www.htmldog.com/>

[7]. <https://dotnetguru.org/uu-nhuoc-diem-cua-web-forms/>

[8]. <https://toidicodedao.com/2016/06/21/tranh-xa-asp-web-form/>

[9]. <https://topdev.vn/blog/html-la-gi/>

[10]. <https://topdev.vn/blog/html-la-gi/>

[11]. <https://html.com/>

[12]. Chat gpt