**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 WEB BÁN ĐỒNG HỒ**

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

**Sinh viên thực hiện:**  **NGUYỄN SĨ ĐẠI**

**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 WEB BÁN GIÀY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020121 | Nguyễn Sĩ Đại | 10/10/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

**Triển khai hệ thống phần mềm** là một quá trình quan trọng và đầy thử thách, đánh dấu giai đoạn chuyển tiếp từ việc phát triển phần mềm sang sử dụng thực tế để mang lại giá trị cho doanh nghiệp. Quá trình này bao gồm nhiều hoạt động, từ cài đặt phần mềm, cấu hình hệ thống, đào tạo nhân viên, cho đến vận hành và bảo trì.Thiết kế web là một dự án quan trọng, góp phần nâng cao hình ảnh và thương hiệu của tên doanh nghiệp,tổ chức trên mạng Internet. Chúng tôi tin tưởng rằng với sự cam kết của ban lãnh đạo, sự phối hợp chặt chẽ của các bên liên quan và nỗ lực của nhóm dự án, dự án sẽ được triển khai thành công và mang lại hiệu quả cao.

MỤC LỤC

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 7](#_Toc159943579)

[1.1. Các khái niệm cơ bản 7](#_Toc159943580)

[1.2. Phân biệt Web tĩnh, Web động 7](#_Toc159943581)

[1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage, 8](#_Toc159943582)

[Web Site, WebServer, URL-Uniform Resource Location, Browser, …) 8](#_Toc159943583)

[1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …) 9](#_Toc159943584)

[**CHƯƠNG II: HTML & HTML5** 11](#_Toc159943585)

[2.1. Tổng quan về HTML 11](#_Toc159943586)

[2.3. Các thẻ HTML thông dụng 12](#_Toc159943587)

[2.4. Các thẻ tạo biểu mẫu (form) 13](#_Toc159943588)

[2.5. Một số thẻ HTML đặc biệt 15](#_Toc159943589)

[2.6. HTML5 16](#_Toc159943590)

[2.6. HTML5 18](#_Toc159943591)

[Cú pháp HTML5 19](#_Toc159943592)

[Khai báo DOCTYPE trong HTML5 19](#_Toc159943593)

[Thẻ ngữ nghĩa 20](#_Toc159943594)

[Webform 20](#_Toc159943595)

[Kết luận chương 2 20](#_Toc159943596)

[**CHƯƠNG III: CSS và CSS3** 21](#_Toc159943597)

[3.1. CSS là gì? 21](#_Toc159943598)

[3.2. Cú pháp CSS 21](#_Toc159943599)

[3.3. Áp dụng CSS vào trang HTML 22](#_Toc159943600)

[3.4. Selectors 24](#_Toc159943601)

[3.4.1. Universal selector 26](#_Toc159943602)

[3.4.2. Type selector 27](#_Toc159943603)

[3.4.3. Identity selector 27](#_Toc159943604)

[3.4.4. Class selector 28](#_Toc159943605)

[3.4.5. Descendant selector 28](#_Toc159943606)

[3.4.6. Child selector 29](#_Toc159943607)

[3.4.7. Adjacent sibling selector 29](#_Toc159943608)

[3.4.8. Attribute selector 30](#_Toc159943609)

[3.4.9. Pseudo class selector 30](#_Toc159943610)

[3.4.10. Group selector 31](#_Toc159943611)

[3.5. Đơn vị đo lường CSS 31](#_Toc159943612)

[3.6. Kế thừa thuộc tính 32](#_Toc159943613)

[3.7. Các nhóm thuộc tính trong CSS 33](#_Toc159943614)

[3.9. Flex 36](#_Toc159943615)

[3.10. Grid 37](#_Toc159943616)

[3.11. CSS3 38](#_Toc159943617)

[Tính năng nổi bật của CSS3: 38](#_Toc159943618)

[Lưu ý: 39](#_Toc159943619)

[3.12. SCSS 39](#_Toc159943620)

[3.13. SASS 39](#_Toc159943621)

[Kết luận chương 3 40](#_Toc159943622)

[**CHƯƠNG 4: THIẾT KẾ WEBISTE THEO ĐỀ TÀI BẠN CHỌN** 41](#_Toc159943623)

[4.1. Ý tưởng của Website 41](#_Toc159943624)

[4.2. Xây dựng bố cục của trang Web 42](#_Toc159943625)

[4.3. Thiết kế trang Web bằng HTML và CSS 43](#_Toc159943626)

[4.5. Kết quả đạt được 45](#_Toc159943627)

[Kết luận chương 4 46](#_Toc159943628)

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

## Các khái niệm cơ bản

Thiết kế web là quá trình tạo ra các trang web. Nó bao gồm nhiều khía cạnh khác nhau, bao gồm bố cục trang, đồ họa, kiểu chữ và khả năng sử dụng. Thiết kế web tốt sẽ vừa đẹp mắt vừa dễ sử dụng.

Triển khai phần mềm là quá trình đưa phần mềm vào sử dụng. Nó bao gồm nhiều bước khác nhau, chẳng hạn như cài đặt phần mềm, cấu hình phần mềm và đào tạo người dùng.

Các loại triển khai phần mềm

Có hai loại triển khai phần mềm chính:

**Triển khai tại chỗ:** Phần mềm được cài đặt trên máy tính hoặc máy chủ của tổ chức.

**Triển khai phần mềm dựa trên đám mây:** Phần mềm được lưu trữ trên máy chủ của bên cung cấp dịch vụ và được truy cập thông qua internet.

## 1.2. Phân biệt Web tĩnh, Web động

Web tĩnh

Khái niệm là website có nội dung được lưu trữ dưới dạng các file HTML, CSS và Javascript cố định. Khi người dùng truy cập, server sẽ gửi nguyên vẹn các file này đến trình duyệt của họ.

Ưu điểm:

Tốc độ tải trang nhanh,chi phí phát triển và bảo trì thấp,an toàn và bảo mật cao,dễ dàng tối ưu hóa cho SEO.

Nhược điểm:

Nội dung không thể thay đổi tự động,khó khăn trong việc cập nhật thông tin và tính tương tác với người dùng thấp.

Ứng dụng phù hợp cho các website có nội dung ít thay đổi, như website giới thiệu công ty, website portfolio, website brochure,...

Web động

Khái niệm là website sử dụng ngôn ngữ lập trình phía máy chủ (như PHP, Python, Ruby) để tạo nội dung trang web dựa trên yêu cầu của người dùng. Khi người dùng truy cập, server sẽ xử lý dữ liệu và tạo ra trang web tương ứng.

Ưu điểm:

Nội dung có thể thay đổi tự động,dễ dàng cập nhật thông tin,tính tương tác với người dùng cao.

Có thể tích hợp nhiều chức năng phức tạp.

Nhược điểm:

Tốc độ tải trang có thể chậm hơn web tĩnh,chi phí phát triển và bảo trì cao hơn,an toàn và bảo mật cần được quan tâm kỹ lưỡng.

Ứng dụng phù hợp cho các website có nội dung thay đổi thường xuyên, như website tin tức, website bán hàng, website mạng xã hội,...

## 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 bạn trên internet, cho phép người dùng truy cập website của bạn dễ dàng. Ví dụ: google.com.

**Web Server Name:** Tên máy chủ thực tế của website, thường không cần người dùng biết vì họ truy cập qua tên miền.

**Hosting:** Dịch vụ cho thuê không gian lưu trữ website của bạn trên internet. Tương tự như việc thuê mặt bằng để đặt máy chủ của bạn.

**Web Site:** Tổng thể các trang web liên kết với nhau và được đặt dưới một tên miền chung. Ví dụ: trang web của FPT Software bao gồm fpt.com.vn, fptsoftware.com, ...

**Web Server:** Phần mềm chạy trên máy chủ lưu trữ website, xử lý yêu cầu của người dùng và gửi lại nội dung website.

**URL (Uniform Resource Locator):** Đường dẫn xác định vị trí duy nhất của một tài nguyên (tài liệu, hình ảnh, trang web) trên internet. Ví dụ: <https://www.fpt.com.vn/> là URL của trang chủ FPT Software.

**Browser:** Trình duyệt web, công cụ cho phép người dùng truy cập và tương tác với các website thông qua URL. Ví dụ: Chrome, Firefox, Safari.

**HomePage:** Trang chính (trang đầu tiên) của một website.

**Các thuật ngữ khác:**

**Subdomain:** Một phần nhỏ của tên miền chính. Ví dụ: [đã xoá URL không hợp lệ].vn là subdomain của fpt.com.vn.

**CMS (Content Management System):** Hệ thống quản trị nội dung, giúp người dùng dễ dàng tạo, chỉnh sửa và xuất bản nội dung website mà không cần biết coding.

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

Visual Studio Code:

Trình chỉnh sửa mã miễn phí, mã nguồn mở, đa nền tảng,hỗ trợ nhiều ngôn ngữ lập trình web phổ biến như HTML, CSS, JavaScript, PHP, Python,...Cung cấp nhiều tiện ích mở rộng giúp nâng cao hiệu quả công việc,phù hợp cho cả người mới bắt đầu và lập trình viên web chuyên nghiệp.

Adobe Dreamweaver:

Phần mềm thiết kế web trả phí với nhiều tính năng mạnh mẽ,hỗ trợ chỉnh sửa trực quan (WYSIWYG) và viết mã,cung cấp nhiều công cụ hỗ trợ thiết kế giao diện web, quản lý nội dung, tối ưu hóa SEO,...

Phù hợp cho các nhà thiết kế web chuyên nghiệp.

WYSIWYG Web Editors:

Cho phép thiết kế web trực quan mà không cần viết mã,một số công cụ phổ biến: Wix, Squarespace, Weebly,...phù hợp cho người mới bắt đầu hoặc những người muốn tạo website đơn giản.

CMS (Content Management System):

Hệ thống quản trị nội dung giúp tạo và quản lý website dễ dàng,một số CMS phổ biến: WordPress, Joomla, Drupal,...Phù hợp cho website có nhiều nội dung và cần cập nhật thường xuyên.

Framework CSS:

Giúp tạo giao diện web đẹp mắt và responsive (tự động thích ứng với mọi thiết bị),một số framework phổ biến: Bootstrap, Foundation, Materialize,...Phù hợp cho các nhà thiết kế web muốn tạo giao diện web chuyên nghiệp.

Lựa chọn công cụ phù hợp:

Lựa chọn công cụ thiết kế web nào phụ thuộc vào nhiều yếu tố, bao gồm:

Mức độ kỹ năng của bạn: Nếu bạn mới bắt đầu, hãy chọn công cụ dễ sử dụng như WYSIWYG Web Editors hoặc CMS.

Loại website bạn muốn tạo: Nếu bạn muốn tạo website đơn giản, WYSIWYG Web Editors hoặc CMS là lựa chọn phù hợp. Nếu bạn muốn tạo website phức tạp hơn, hãy chọn Visual Studio Code hoặc Adobe Dreamweaver.

Ngân sách của bạn: Một số công cụ như Visual Studio Code và WYSIWYG Web Editors là miễn phí, trong khi Adobe Dreamweaver là phần mềm trả phí.

Kết luận chương 1

Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực quan trọng trong việc tạo ra các ứng dụng web hiệu quả và đáp ứng nhu cầu của người dùng.

Thiết kế web tập trung vào việc tạo ra giao diện người dùng trực quan và dễ sử dụng, đảm bảo tính thẩm mỹ và thu hút người dùng. Triển khai hệ thống phần mềm bao gồm việc xây dựng cấu trúc và chức năng của website, đảm bảo tính bảo mật và hiệu quả hoạt động.

Để tạo ra một website thành công, cần kết hợp hài hòa giữa thiết kế web và triển khai hệ thống phần mềm. Việc lựa chọn công nghệ và phương pháp phù hợp sẽ phụ thuộc vào mục đích sử dụng, ngân sách và yêu cầu cụ thể của từng dự án.

# **CHƯƠNG II: HTML & HTML5**

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

## 2.1. Tổng quan về HTML

**HTML** là viết tắt của **HyperText Markup Language**, là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo ra các trang web. Nó là ngôn ngữ cơ bản nhất của World Wide Web (WWW) và được sử dụng để định nghĩa cấu trúc và nội dung của các trang web.



2.2. Cấu trúc tổng quát trang HTML

**1. Phần khai báo (head):**

Nằm giữa thẻ <html> và </html>.

Chứa thông tin về trang web, không hiển thị trực tiếp trên trang.

Một số thẻ HTML phổ biến trong phần head:

<meta>: Cung cấp thông tin về trang web như charset, author, keywords,...

<title>: Định nghĩa tiêu đề trang web.

<link>: Liên kết đến các tập tin CSS và JavaScript.

<script>: Chèn mã JavaScript vào trang web.

**2. Phần nội dung (body):**

Nằm giữa thẻ <body> và </body>.

Chứa nội dung chính hiển thị trên trang web.

Một số thẻ HTML phổ biến trong phần body:

<p>: Định nghĩa đoạn văn bản.

<h1> đến <h6>: Định nghĩa heading (tiêu đề) từ cấp 1 đến cấp 6.

<img>: Chèn hình ảnh vào trang web.

<a>: Tạo liên kết đến một trang web khác.

<table>: Định nghĩa bảng.

<ul>: Định nghĩa danh sách không thứ tự.

<ol>: Định nghĩa danh sách có thứ tự.

## 2.3. Các thẻ HTML thông dụng

**1. Các thẻ xử lý văn bản:**

**Thẻ**<p>**:** Định nghĩa đoạn văn bản.

**Thẻ**<h1>**đến**<h6>**:** Định nghĩa heading (tiêu đề) từ cấp 1 đến cấp 6.

**Thẻ**<br>**:** Tạo dòng mới.

**Thẻ**<pre>**:** Định nghĩa định dạng văn bản cố định.

**Thẻ**<strong>**:** Định nghĩa văn bản đậm.

**Thẻ**<em>**:** Định nghĩa văn bản in nghiêng.

**Thẻ**<mark>**:** Đánh dấu văn bản.

**Thẻ**<ins>**:** Chèn văn bản mới.

**Thẻ**<del>**:** Xóa văn bản.

**2. Các thẻ tạo bảng:**

**Thẻ**<table>**:** Định nghĩa bảng.

**Thẻ**<tr>**:** Định nghĩa hàng trong bảng.

**Thẻ**<td>**:** Định nghĩa ô trong bảng.

**Thẻ**<thead>**:** Định nghĩa phần đầu bảng.

**Thẻ**<tbody>**:** Định nghĩa phần thân bảng.

**Thẻ**<tfoot>**:** Định nghĩa phần chân bảng.

**3. Thẻ liên kết:**

**Thẻ**<a>**:** Tạo liên kết đến một trang web khác.

**Thuộc tính**href**:** Cung cấp địa chỉ URL của trang web liên kết.

**Thuộc tính**target**:** Xác định vị trí mở trang web liên kết.

**4. Các thẻ đa phương tiện:**

**Thẻ**<img>**:** Chèn hình ảnh vào trang web.

**Thuộc tính**src**:** Cung cấp địa chỉ URL của hình ảnh.

**Thuộc tính**alt**:** Cung cấp văn bản thay thế cho hình ảnh.

**Thẻ**<video>**:** Chèn video vào trang web.

**Thuộc tính**src**:** Cung cấp địa chỉ URL của video.

**Thẻ**<audio>**:** Chèn âm thanh vào trang web.

**Thuộc tính**src**:** Cung cấp địa chỉ URL của âm thanh.

**5. Thẻ tạo khung**

**Thẻ**<frameset>**:** Định nghĩa khung cho trang web.

**Thẻ**<frame>**:** Định nghĩa nội dung cho từng khung.

**Thuộc tính**src**:** Cung cấp địa chỉ URL của nội dung cho khung.

## 2.4. Các thẻ tạo biểu mẫu (form)

**Thẻ** <form>**:** Định nghĩa biểu mẫu.

**Thuộc tính**action**:** Cung cấp địa chỉ URL để xử lý dữ liệu biểu mẫu.

**Thuộc tính**method**:** Xác định phương thức gửi dữ liệu biểu mẫu.

**Thẻ** <input>**:** Tạo các trường nhập liệu trong biểu mẫu.

**Thuộc tính**type**:** Xác định loại trường nhập liệu.

text: Trường nhập văn bản.

password: Trường nhập mật khẩu.

radio: Nút radio.

checkbox: Hộp kiểm.

submit: Nút gửi.

reset: Nút đặt lại.

**Thuộc tính**name**:** Cung cấp tên cho trường nhập liệu.

**Thuộc tính**value**:** Cung cấp giá trị mặc định cho trường nhập liệu.

**Thẻ** <label>**:** Gắn nhãn cho các trường nhập liệu.

**Thuộc tính**for**:** Cung cấp ID của trường nhập liệu liên quan.

**Thẻ** <textarea>**:** Tạo trường nhập văn bản đa dòng.

**Thuộc tính**name**:** Cung cấp tên cho trường nhập liệu.

**Thuộc tính**rows**:** Xác định số lượng hàng của trường nhập liệu.

**Thuộc tính**cols**:** Xác định số lượng cột của trường nhập liệu.

**Thẻ** <select>**:** Tạo danh sách lựa chọn.

**Thuộc tính**name**:** Cung cấp tên cho danh sách lựa chọn.

**Thẻ**<option>**:** Định nghĩa các lựa chọn trong danh sách.

**Thuộc tính**value**:** Cung cấp giá trị cho lựa chọn.

**Thẻ** <button>**:** Tạo nút bấm.

**Thuộc tính**type**:** Xác định loại nút bấm.

submit: Nút gửi.

reset: Nút đặt lại.

**Thuộc tính**name**:** Cung cấp tên cho nút bấm.

**Thuộc tính**value**:** Cung cấp giá trị cho nút bấm.

## 2.5. Một số thẻ HTML đặc biệt

DOCTYPE html> Đây là thẻ HTML đặc biệt xác định rằng tài liệu là tài liệu HTML.

<html> Đây là thẻ HTML đặc biệt xác định đầu trang HTML.

<head> Đây là thẻ HTML đặc biệt chứa thông tin về trang HTML, chẳng hạn như tiêu đề và tác giả.

<title> Đây là thẻ HTML đặc biệt xác định tiêu đề của trang HTML.

<meta> Đây là thẻ HTML đặc biệt cung cấp thông tin bổ sung về trang HTML, chẳng hạn như mô tả và từ khóa.

<body> Đây là thẻ HTML đặc biệt xác định phần thân của trang HTML, chứa nội dung chính của trang.

<h1> Đây là thẻ HTML đặc biệt xác định tiêu đề cấp 1.

<h2> Đây là thẻ HTML đặc biệt xác định tiêu đề cấp 2.

<h3> Đây là thẻ HTML đặc biệt xác định tiêu đề cấp 3.

<p> Đây là thẻ HTML đặc biệt xác định một đoạn văn bản.

<br> Đây là thẻ HTML đặc biệt xác định một ngắt dòng.

<img> Đây là thẻ HTML đặc biệt chèn hình ảnh vào trang HTML.

<table> Đây là thẻ HTML đặc biệt xác định bảng.

<tr> Đây là thẻ HTML đặc biệt xác định một hàng trong bảng.

<td> Đây là thẻ HTML đặc biệt xác định một ô trong bảng.

<a> Đây là thẻ HTML đặc biệt xác định một liên kết.

<form> Đây là thẻ HTML đặc biệt xác định một biểu mẫu.

<input> Đây là thẻ HTML đặc biệt xác định một trường nhập trong biểu mẫu.

<button> Đây là thẻ HTML đặc biệt xác định một nút trên biểu mẫu.

</body> Đây là thẻ HTML đặc biệt xác định phần cuối của phần thân trang HTML.

</html> Đây là thẻ HTML đặc biệt xác định phần cuối của trang HTML.

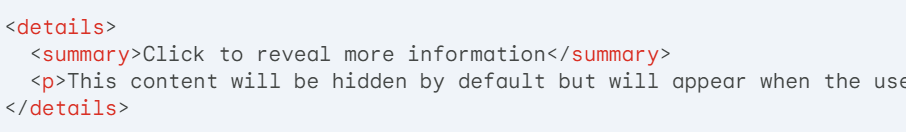
## 2.6. HTML5

**Thẻ** <details> **và** <summary>**:**

Dùng để tạo phần nội dung có thể thu gọn/mở rộng.

<details>: Bao quanh phần nội dung có thể thu gọn.

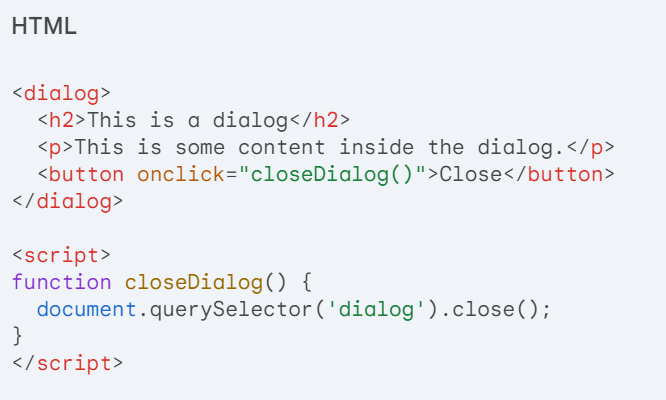
<summary>: Hiển thị tiêu đề cho phần nội dung có thể thu gọn.



**Thẻ** <dialog>**:**

Dùng để tạo hộp thoại modal.

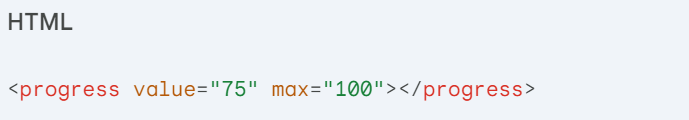
Hiển thị nội dung chồng lên trên trang web hiện tại.



**Thẻ** <progress>**:**

Dùng để hiển thị thanh tiến trình.

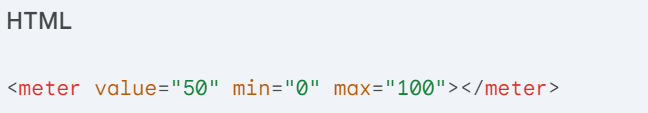
Hiển thị tiến trình của một tác vụ đang thực hiện.



**Thẻ** <meter>**:**

Dùng để hiển thị đồng hồ đo.

Hiển thị giá trị hiện tại trong một phạm vi giá trị.



**Thẻ** <template>**:**

Dùng để định nghĩa một mẫu nội dung HTML.

Nội dung được định nghĩa trong thẻ <template> không được hiển thị trực tiếp.

Có thể sử dụng JavaScript để chèn nội dung từ thẻ <template> vào trang web.



**Thẻ** <slot>**:**

Dùng để định vị điểm chèn nội dung trong một mẫu HTML.

Có thể sử dụng JavaScript để chèn nội dung khác nhau vào các vị trí <slot> khác nhau.



## 2.6. HTML5

HTML5 là phiên bản mới nhất của ngôn ngữ HTML (HyperText Markup Language) được sử dụng để tạo trang web. Nó được World Wide Web Consortium (W3C) công bố vào năm 2014.

**Lợi ích của HTML5:**

**Cải thiện khả năng tương tác:** HTML5 cung cấp nhiều thẻ ngữ nghĩa mới giúp trình duyệt hiểu rõ hơn về nội dung trang web, từ đó nâng cao khả năng tương tác và truy cập cho người dùng.

**Hỗ trợ đa phương tiện:** HTML5 cung cấp các API tích hợp để xử lý video, âm thanh và đồ họa trực tiếp trong trang web, loại bỏ sự phụ thuộc vào trình cắm thêm (plugin).

**Dễ học và sử dụng:** HTML5 có cú pháp đơn giản và dễ hiểu hơn so với các phiên bản HTML trước đây.

**Tương thích với các thiết bị di động:** HTML5 được thiết kế để tương thích với các thiết bị di động, giúp tạo ra trang web hiển thị đẹp mắt trên mọi màn hình.

### Cú pháp HTML5

Cú pháp HTML5 dựa trên XML, vì vậy nó có một số quy tắc mới so với HTML4:

**Tất cả các thẻ phải được đóng.**

**Tên thẻ phải viết bằng chữ thường.**

**Thuộc tính phải được đặt trong dấu ngoặc kép.**

**Giá trị thuộc tính phải được đặt trong dấu ngoặc kép hoặc đơn.**



### Khai báo DOCTYPE trong HTML5

DOCTYPE là dòng đầu tiên của trang HTML, nó thông báo cho trình duyệt về phiên bản HTML được sử dụng.

Khai báo DOCTYPE trong HTML5:



### Thẻ ngữ nghĩa

HTML5 cung cấp nhiều thẻ ngữ nghĩa mới giúp trình duyệt hiểu rõ hơn về nội dung trang web. Ví dụ:

header: Thẻ header chứa thông tin tiêu đề trang web.

nav: Thẻ nav chứa các liên kết điều hướng trang web.

section: Thẻ section chứa một phần nội dung của trang web.

article: Thẻ article chứa một bài viết.

aside: Thẻ aside chứa nội dung bổ sung cho nội dung chính của trang web.

footer: Thẻ footer chứa thông tin chân trang web.

### Webform

HTML5 cung cấp các API mới để tạo biểu mẫu web (webform) một cách dễ dàng và hiệu quả hơn so với HTML4. Ví dụ:

input type="date": Cho phép người dùng chọn ngày tháng.

input type="time": Cho phép người dùng chọn thời gian.

input type="number": Cho phép người dùng nhập số.

input type="email": Cho phép người dùng nhập email.

input type="url": Cho phép người dùng nhập URL.

Kết luận chương 2

**HTML** là ngôn ngữ lập trình cơ bản để tạo trang web. Nó cung cấp các công cụ để định dạng văn bản, chèn hình ảnh, video, liên kết và các nội dung khác vào trang web.

**HTML5** là phiên bản mới nhất của HTML với nhiều cải tiến và tính năng mới. Nó giúp tạo ra trang web đẹp mắt, dễ sử dụng và tương thích với nhiều thiết bị khác nhau.

**CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

## 3.1. CSS là gì?

CSS là viết tắt của Cascading Style Sheets, là một ngôn ngữ lập trình được sử dụng để định dạng và tạo kiểu cho các trang web. Nó hoạt động song song với HTML, ngôn ngữ dùng để tạo cấu trúc và nội dung cho trang web.

## 3.2. Cú pháp CSS

**1. Bộ chọn (Selector)**

Bộ chọn xác định các phần tử HTML mà bạn muốn định dạng. Có nhiều loại bộ chọn khác nhau, bao gồm:

**Bộ chọn thẻ:** Chọn các phần tử HTML dựa trên tên thẻ của chúng. Ví dụ: h1, p, div.

**Bộ chọn lớp:** Chọn các phần tử HTML dựa trên lớp được gán cho chúng. Ví dụ: .header, .content, .footer.

**Bộ chọn ID:** Chọn các phần tử HTML dựa trên ID được gán cho chúng. Ví dụ: #main, #sidebar, #footer.

**Bộ chọn con:** Chọn các phần tử HTML con của một phần tử khác. Ví dụ: div p, ul li.

**Bộ chọn chung:** Chọn tất cả các phần tử HTML trên trang.

**2. Thuộc tính (Property)**

Thuộc tính là các đặc điểm mà bạn muốn định dạng cho các phần tử HTML được chọn. Có rất nhiều thuộc tính CSS khác nhau, bao gồm:

**Màu sắc:** color

**Phông chữ:** font-family, font-size, font-weight

**Kích thước:** width, height

**Vị trí:** margin, padding, text-align

**Hiệu ứng:** border, background, box-shadow

**3. Giá trị (Value)**

Giá trị là các giá trị cụ thể mà bạn gán cho các thuộc tính. Ví dụ:

color: red;

font-family: Arial, sans-serif;

font-size: 16px;

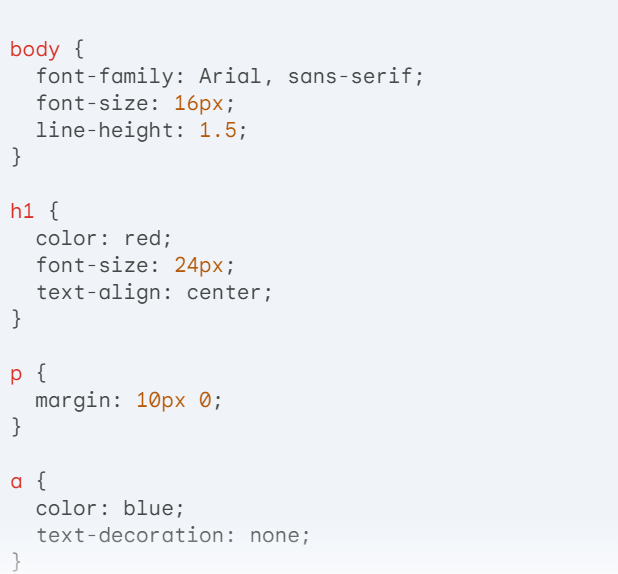
width: 500px;

margin: 10px;

**Cấu trúc:**

Một quy tắc CSS bao gồm bộ chọn, thuộc tính và giá trị được viết trong ngoặc vuông và được ngăn cách bởi dấu hai chấm.





## 3.3. Áp dụng CSS vào trang HTML

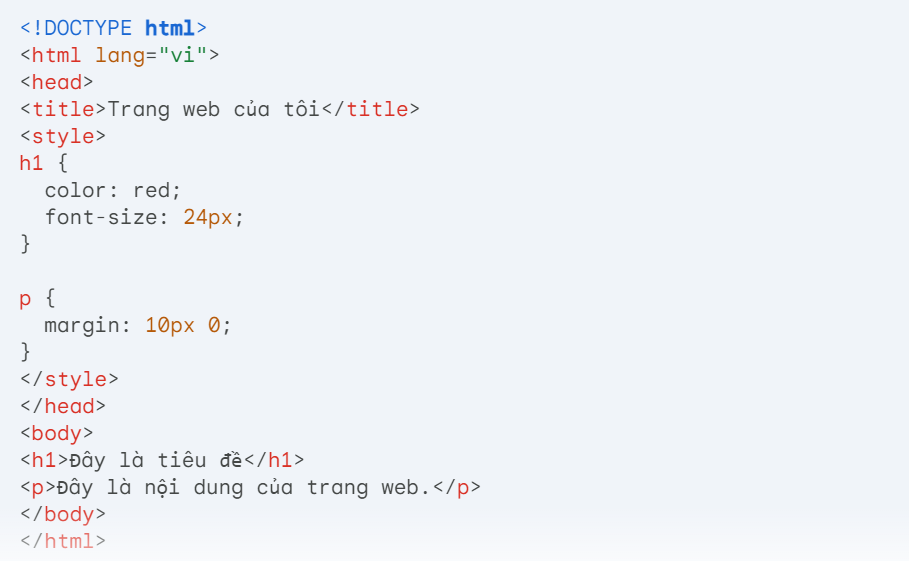
**1. CSS nội tuyến (Inline CSS)**

Với cách này, bạn thêm mã CSS trực tiếp vào thẻ HTML. Đây là cách đơn giản nhất để áp dụng CSS, nhưng nó không phù hợp cho việc sử dụng nhiều lần hoặc cho các trang web lớn.

\

**2. CSS nhúng (Internal CSS)**

Với cách này, bạn thêm mã CSS vào phần <head> của trang HTML. Đây là cách tốt để áp dụng CSS cho một trang HTML duy nhất.



**3. CSS liên kết (External CSS)**

Với cách này, bạn lưu trữ mã CSS trong một tệp riêng biệt với phần mở rộng .css và sau đó liên kết tệp đó với trang HTML. Đây là cách tốt nhất để áp dụng CSS cho nhiều trang web.





**Lựa chọn phương pháp phù hợp:**

Lựa chọn phương pháp áp dụng CSS phù hợp phụ thuộc vào nhu cầu cụ thể của bạn.

**CSS nội tuyến:** Sử dụng cho các thay đổi nhỏ và đơn giản.

**CSS nhúng:** Sử dụng cho các thay đổi áp dụng cho một trang HTML duy nhất.

**CSS liên kết:** Sử dụng cho các thay đổi áp dụng cho nhiều trang HTML.

**Lời khuyên:**

Nên sử dụng CSS liên kết để tách biệt nội dung HTML khỏi định dạng CSS.

Nên đặt tên tệp CSS theo cách mô tả nội dung của nó.

Nên sử dụng các công cụ dành cho nhà phát triển web để giúp bạn viết và kiểm tra mã CSS của bạn.

## 3.4. Selectors

**Bộ chọn CSS** là mẫu để chọn phần tử HTML mà bạn muốn định dạng. Có nhiều loại bộ chọn khác nhau, bao gồm:

**1. Bộ chọn thẻ (Type Selectors)**

Chọn các phần tử HTML dựa trên tên thẻ của chúng. Ví dụ:

h1: Chọn tất cả các thẻ <h1>.

p: Chọn tất cả các thẻ <p>.

div: Chọn tất cả các thẻ <div>.

**2. Bộ chọn lớp (Class Selectors)**

Chọn các phần tử HTML dựa trên lớp được gán cho chúng. Ví dụ:

.header: Chọn tất cả các phần tử HTML có lớp header.

.content: Chọn tất cả các phần tử HTML có lớp content.

.footer: Chọn tất cả các phần tử HTML có lớp footer.

**3. Bộ chọn ID (ID Selectors)**

Chọn các phần tử HTML dựa trên ID được gán cho chúng. Ví dụ:

#main: Chọn phần tử HTML có ID main.

#sidebar: Chọn phần tử HTML có ID sidebar.

#footer: Chọn phần tử HTML có ID footer.

**4. Bộ chọn con (Descendant Selectors)**

Chọn các phần tử HTML con của một phần tử khác. Ví dụ:

div p: Chọn tất cả các thẻ <p> là con của thẻ <div>.

ul li: Chọn tất cả các thẻ <li> là con của thẻ <ul>.

table td: Chọn tất cả các thẻ <td> là con của thẻ <table>.

**5. Bộ chọn chung (Universal Selector)**

Chọn tất cả các phần tử HTML trên trang. Ví dụ:

\*: Chọn tất cả các phần tử HTML.

**6. Bộ chọn kết hợp (Combinators)**

Kết hợp các bộ chọn để chọn các phần tử HTML cụ thể hơn. Ví dụ:

h1.title: Chọn tất cả các thẻ <h1> có lớp title.

div p.content: Chọn tất cả các thẻ <p> có lớp content là con của thẻ <div>.

ul li:first-child: Chọn thẻ <li> đầu tiên là con của thẻ <ul>.

**7. Bộ chọn thuộc tính (Attribute Selectors)**

Chọn các phần tử HTML dựa trên thuộc tính của chúng. Ví dụ:

img[alt]: Chọn tất cả các thẻ <img> có thuộc tính alt.

a[href]: Chọn tất cả các thẻ <a> có thuộc tính href.

input[type="radio"]: Chọn tất cả các thẻ <input> có thuộc tính type bằng radio.

**8. Bộ chọn giả (Pseudo-classes)**

Chọn các phần tử HTML dựa trên trạng thái của chúng. Ví dụ:

a:hover: Chọn thẻ <a> khi con trỏ chuột di chuột qua.

li:active: Chọn thẻ <li> đang được kích hoạt.

input:focus: Chọn thẻ <input> đang được chọn.

**9. Bộ chọn giả phần tử (Pseudo-elements)**

Chọn các phần tử giả của một phần tử HTML. Ví dụ:

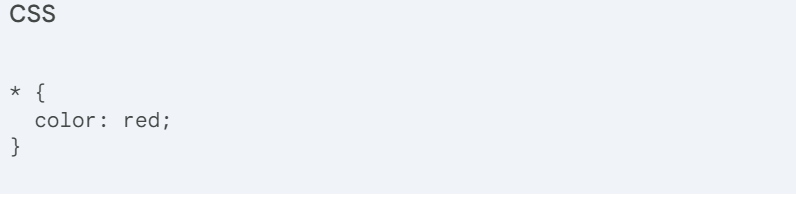
::first-letter: Chọn chữ cái đầu tiên của một phần tử.

::before: Chọn nội dung trước một phần tử.

::after: Chọn nội dung sau một phần tử.

### 3.4.1. Universal selector

**Universal selector**, hay còn gọi là **bộ chọn chung**, là một bộ chọn CSS dùng để chọn tất cả các phần tử HTML trên trang web. Nó được biểu thị bằng ký hiệu \*.

****

**Lưu ý:**

Universal selector nên được sử dụng một cách cẩn thận vì nó có thể ảnh hưởng đến tất cả các phần tử HTML trên trang web.

Nên sử dụng các bộ chọn cụ thể hơn để tránh ảnh hưởng đến các phần tử HTML không mong muốn.

**Một số trường hợp sử dụng Universal selector:**

**Đặt lại kiểu dáng:** Sử dụng universal selector để đặt lại tất cả các kiểu dáng được áp dụng cho các phần tử HTML.

**Tạo hiệu ứng toàn trang:** Sử dụng universal selector để tạo hiệu ứng ảnh hưởng đến tất cả các phần tử HTML trên trang web.

**Chọn các phần tử không có lớp hoặc ID:** Sử dụng universal selector để chọn các phần tử HTML không có lớp hoặc ID.

### 3.4.2. Type selector

Trong CSS, **Type selector (bộ chọn theo loại)** là một cách chọn các phần tử HTML dựa trên **tên thẻ** của chúng.

Đây là bộ chọn cơ bản và dễ sử dụng nhất, giúp bạn định dạng kiểu dáng cho nhiều phần tử cùng loại trên trang web một cách nhanh chóng.

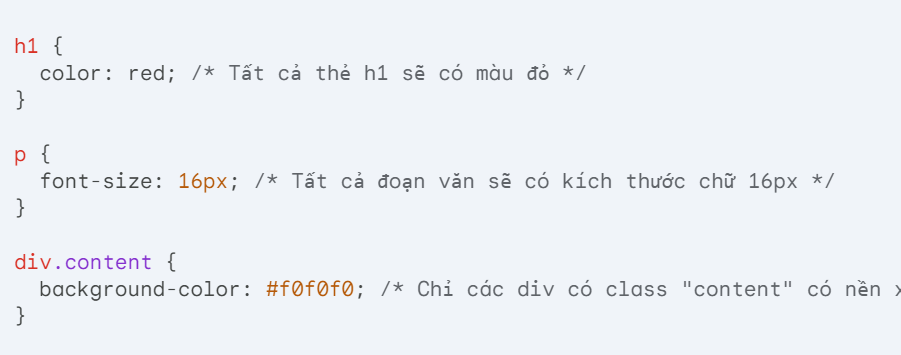
**Cách sử dụng:**

Chỉ cần viết tên thẻ bạn muốn chọn, không cần thêm bất kỳ ký hiệu đặc biệt nào. Ví dụ:

h1: Chọn tất cả các thẻ tiêu đề chính <h1>.

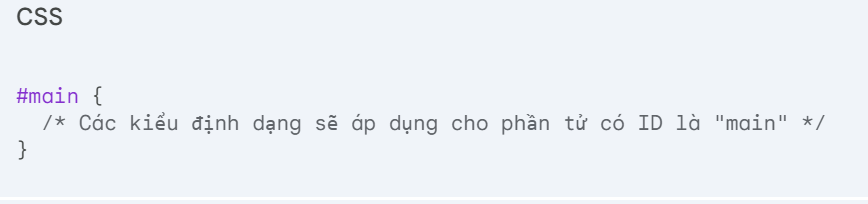
p: Chọn tất cả các đoạn văn <p>.

div: Chọn tất cả các vùng nội dung <div>.

****

### 3.4.3. Identity selector

Trong CSS, **Identity selector (bộ chọn ID)** là cách chọn một phần tử HTML duy nhất trên trang web dựa trên **ID** (định danh) được gán cho nó. ID phải là duy nhất trong toàn bộ tài liệu HTML và thường dùng để chọn các thành phần chính của trang**.**

****

**Ưu điểm:**

Chính xác cao, chỉ chọn được một phần tử duy nhất.

Dễ dàng định dạng các thành phần quan trọng của trang.

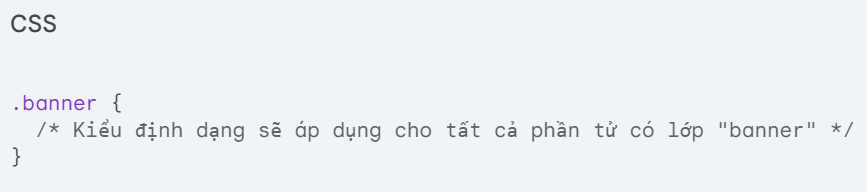
Thường dùng trong JavaScript để tương tác với các phần tử trên trang.

### 3.4.4. Class selector

Trong CSS, **Class selector (bộ chọn lớp)** là một cách chọn các phần tử HTML dựa trên **lớp (class)** được gán cho chúng. Lớp là một thuộc tính tùy chọn có thể gán cho nhiều phần tử khác nhau, giúp bạn định dạng kiểu dáng theo nhóm một cách linh hoạt.

**Cách sử dụng:**

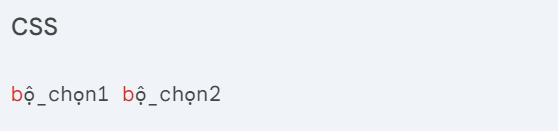
Để chọn các phần tử dựa trên lớp, hãy sử dụng dấu chấm (.) trước tên lớp.

****

### 3.4.5. Descendant selector

Bộ chọn con cháu, được biểu thị bằng một khoảng trắng đơn, là một trong những bộ kết hợp cơ bản trong CSS. Nó cho phép bạn chọn các phần tử được lồng nhúng bất kỳ nơi nào bên trong một phần tử khác, bất kể độ sâu của chúng trong cây DOM.

**Cú pháp:**



bộ\_chọn1: Có thể là bất kỳ bộ chọn CSS hợp lệ nào (ví dụ: tên thẻ, tên lớp, ID).

bộ\_chọn2: Chọn bất kỳ phần tử nào được lồng nhúng bên trong bộ\_chọn1.

### 3.4.6. Child selector

**Chọn con** là một khái niệm quan trọng trong CSS, cho phép bạn chọn các phần tử dựa trên mối quan hệ của chúng với các phần tử khác trong DOM. Có hai khía cạnh chính cần nắm rõ:

**1. Trình kết hợp con (>)**:

Ký hiệu này được sử dụng để chọn các phần tử là con trực tiếp của một phần tử khác. Ví dụ: div > p sẽ chọn tất cả các phần tử <p> là con trực tiếp của các phần tử <div>.

**2. Tả bán giả :nth-child():**

Tả bán giả này cho phép bạn chọn các phần tử dựa trên vị trí của chúng trong danh sách con của phần tử cha. Bạn có thể chỉ định vị trí chính xác (ví dụ: :nth-child(3) cho con thứ ba), sử dụng từ khóa như odd hoặc even, hoặc tạo công thức như an + b để phù hợp với các mẫu phức tạp hơn.

**Lưu ý**:

Trình kết hợp con khắt khe hơn trình kết hợp hậu duệ (), vốn chọn bất kỳ phần tử nào là hậu duệ của phần tử được chỉ định, bất kể mức độ sâu.

Khi sử dụng :nth-child(), hãy nhớ rằng chỉ mục của con đầu tiên là 1.

Cũng có các biến thể như :nth-of-type() chỉ xem xét các phần tử cùng loại (tên thẻ) trong phần tử cha.

### 3.4.7. Adjacent sibling selector

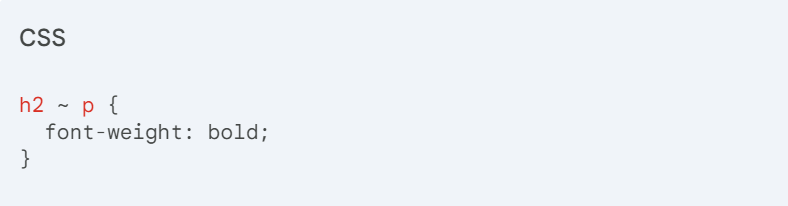
**1. Trình kết hợp anh chị em liền kề (+):**

Ký hiệu + cho phép bạn chọn phần tử anh chị em **liền kề ngay sau** phần tử được chỉ định. Cả hai phần tử phải cùng cấp và có cùng phần tử cha.



**2. Trình chọn anh chị em tổng quát (~):**

Ký hiệu ~ cho phép bạn chọn **tất cả các phần tử anh chị em** của phần tử được chỉ định, **miễn là chúng cùng cấp và có cùng phần tử cha**.



### 3.4.8. Attribute selector

Chọn phần tử dựa trên thuộc tính (Attribute selector) là một công cụ mạnh mẽ trong CSS cho phép bạn chọn các phần tử HTML dựa trên giá trị của các thuộc tính HTML. Dưới đây là một số điểm chính cần nắm rõ:

**Cấu trúc:**



### 3.4.9. Pseudo class selector

Bộ chọn lớp giả là một loại bộ chọn đặc biệt trong CSS cho phép bạn chọn các phần tử dựa trên trạng thái hoặc điều kiện hiện tại của chúng. Chúng không phải là một phần của HTML của phần tử mà được thêm vào bộ chọn phần tử bằng cách sử dụng dấu hai chấm (:).

**Dưới đây là một số ví dụ phổ biến về bộ chọn lớp giả:**

**:hover:** Bộ chọn lớp giả này chọn một phần tử khi người dùng di chuột qua nó. Nó thường được sử dụng để thay đổi giao diện của các nút, liên kết và các phần tử tương tác khác khi di chuột qua.

**:active:** Bộ chọn lớp giả này chọn một phần tử khi người dùng nhấp vào nó và giữ nút chuột trái. Nó thường được sử dụng để định kiểu các nút và các phần tử biểu mẫu khác trong khi chúng được nhấp.

**:visited:** Bộ chọn lớp giả này chọn các liên kết mà người dùng đã truy cập trước đây. Nó có thể được sử dụng để định kiểu các liên kết đã truy cập khác với các liên kết chưa truy cập.

**:first-child:** Bộ chọn lớp giả này chọn phần tử con đầu tiên của phần tử cha của nó. Nó có thể được sử dụng để định kiểu đoạn văn đầu tiên trong một phần khác với các đoạn văn khác, ví dụ.

**:last-child:** Bộ chọn lớp giả này chọn phần tử con cuối cùng của phần tử cha của nó. Nó có thể được sử dụng để định kiểu đoạn văn cuối cùng trong một phần khác với các đoạn văn khác, ví dụ.

Bộ chọn lớp giả có thể là một công cụ mạnh mẽ để thêm tính tương tác và hiệu ứng trực quan vào trang web của bạn. Chúng có thể được sử dụng để tạo các hiệu ứng như menu di chuột, chú thích công cụ, v.v.

### 3.4.10. Group selector

Trong CSS, **trình chọn nhóm (group selector)** là một cách hữu ích để chọn nhiều phần tử khác nhau và áp dụng cùng một style cho chúng.

**Dấu phẩy (,)**:

Bạn có thể liệt kê các bộ chọn khác nhau, ngăn cách nhau bằng dấu phẩy, để chọn một nhóm phần tử. Mọi style được áp dụng sau dấu phẩy sẽ được áp dụng cho tất cả các phần tử được chọn trong nhóm.



## 3.5. Đơn vị đo lường CSS

**CSS** sử dụng nhiều đơn vị đo lường khác nhau để xác định kích thước, vị trí, và các thuộc tính khác của các thành phần trong trang web. Việc lựa chọn đơn vị đo phù hợp sẽ ảnh hưởng đến tính linh hoạt, khả năng đáp ứng và hiệu quả của trang web.

Dưới đây là một số đơn vị đo lường phổ biến trong CSS:

**Đơn vị tuyệt đối:**

**px (pixel):** Đơn vị cơ bản để đo lường trên màn hình, tương ứng với một điểm ảnh.

**pt (point):** Bằng 1/72 inch, thường được sử dụng để định cỡ font chữ.

**in (inch):** Đơn vị đo lường inch.

**cm (centimeter):** Đơn vị đo lường centimet.

**mm (millimeter):** Đơn vị đo lường milimet.

**Đơn vị tương đối:**

**em:** Bằng kích thước font chữ hiện tại của phần tử cha.

**rem:** Bằng kích thước font chữ của phần tử gốc (html).

**% (phần trăm):** Tỷ lệ phần trăm so với kích thước của phần tử cha.

**Đơn vị viewport:**

**vw (viewport width):** Tỷ lệ phần trăm so với chiều rộng của viewport.

**vh (viewport height):** Tỷ lệ phần trăm so với chiều cao của viewport.

## 3.6. Kế thừa thuộc tính

**Kế thừa thuộc tính CSS** là tính năng cho phép các thuộc tính CSS được truyền từ phần tử cha sang phần tử con trong cấu trúc HTML. Nhờ vậy, bạn có thể định dạng một nhóm phần tử HTML cùng lúc mà không cần phải lặp lại các thuộc tính cho từng phần tử.

**Cách thức hoạt động:**

Khi một phần tử con không được định nghĩa một thuộc tính CSS nào, nó sẽ thừa hưởng giá trị của thuộc tính đó từ phần tử cha.

Nếu phần tử con có định nghĩa riêng cho thuộc tính CSS, giá trị được định nghĩa sẽ được ưu tiên hơn giá trị được kế thừa từ phần tử cha.

Việc kế thừa thuộc tính CSS có thể xảy ra qua nhiều cấp trong cấu trúc HTML.

**Lợi ích:**

Giúp giảm thiểu lượng mã CSS cần viết.

Giúp tạo ra sự đồng nhất trong định dạng trang web.

Giúp dễ dàng thay đổi định dạng cho một nhóm phần tử HTML cùng lúc.



## 3.7. Các nhóm thuộc tính trong CSS

**1. Thuộc tính định dạng font chữ, văn bản (font, text):**

font-family: Kiểu chữ

font-size: Kích thước chữ

font-weight: Độ dày chữ

font-style: Kiểu chữ (in nghiêng, đậm,…)

text-align: Căn chỉnh văn bản

text-decoration: Trang trí văn bản (gạch chân, gạch ngang,…)

**2. Thuộc tính định dạng nền (background):**

background-color: Màu nền

background-image: Hình ảnh nền

background-position: Vị trí hình ảnh nền

background-repeat: Lặp lại hình ảnh nền

**3. Mô hình hộp (box model):**

margin: Khoảng cách giữa phần tử và các phần tử xung quanh

padding: Khoảng cách giữa nội dung và viền của phần tử

border: Viền của phần tử

width: Chiều rộng của phần tử

height: Chiều cao của phần tử

**4. Các thuộc tính định margin, padding, border:**

margin-top: Khoảng cách trên

margin-right: Khoảng cách bên phải

margin-bottom: Khoảng cách dưới

margin-left: Khoảng cách bên trái

padding-top: Khoảng cách trên

padding-right: Khoảng cách bên phải

padding-bottom: Khoảng cách dưới

padding-left: Khoảng cách bên trái

border-top: Viền trên

border-right: Viền bên phải

border-bottom: Viền dưới

border-left: Viền bên trái

**5. Thuộc tính định dạng cách hiển thị (display):**

display: block: Hiển thị phần tử như một khối

display: inline: Hiển thị phần tử như một phần tử nội dòng

display: none: Ẩn phần tử

**6. Thuộc tính xác định vị trí (position):**

position: static: Vị trí mặc định của phần tử

position: relative: Vị trí tương đối so với vị trí ban đầu

position: absolute: Vị trí tuyệt đối so với phần tử cha

position: fixed: Vị trí cố định so với khung nhìn

3.8. Float & Clear

**Float** là một thuộc tính CSS cho phép bạn đặt một phần tử HTML bên cạnh trái hoặc phải của dòng văn bản xung quanh nó. Điều này tạo ra bố cục linh hoạt hơn so với bố cục theo khối mặc định. Tuy nhiên, việc sử dụng float có thể gây ra một số vấn đề về bố cục, đặc biệt là khi các phần tử khác không "chạy" bên dưới các phần tử float.

**Clear** là một thuộc tính CSS được sử dụng để giải quyết vấn đề này. Nó cho biết các phần tử xung quanh một phần tử float nên "chạy" bên dưới nó như thế nào.

**Dưới đây là chi tiết hơn về từng thuộc tính:**

**Float:**

Các giá trị thường dùng của float:

left: Đặt phần tử sang trái.

right: Đặt phần tử sang phải.

none: Loại bỏ float (trở về trạng thái mặc định).

Các phần tử float sẽ được tách ra khỏi dòng văn bản và nổi bên cạnh dòng văn bản.

Nội dung sau phần tử float sẽ không "chạy" bên dưới nó.

**Clear:**

Các giá trị thường dùng của clear:

none: Không xóa float.

left: Xóa float bên trái.

right: Xóa float bên phải.

both: Xóa float cả bên trái và bên phải.

Sử dụng clear để cho biết các phần tử xung quanh một phần tử float nên "chạy" bên dưới nó như thế nào.

Ví dụ, clear: both sẽ đảm bảo nội dung sau phần tử float sẽ "chạy" bên dưới cả hai bên trái và phải của nó.

**Một số lưu ý:**

Sử dụng float và clear có thể làm cho mã CSS phức tạp hơn.

Nên sử dụng các phương pháp bố cục hiện đại hơn như flexbox và grid thay vì float và clear whenever possible.

Nếu bạn phải sử dụng float và clear, hãy sử dụng chúng một cách thận trọng và tuân theo các quy tắc tốt nhất để tránh các vấn đề về bố cục.

## 3.9. Flex

Flexbox, viết tắt của Flexible Box Layout Module, là một mô hình bố cục trong CSS cung cấp cách thức linh hoạt và mạnh mẽ để sắp xếp các phần tử HTML theo chiều ngang hoặc chiều dọc. Nó được hỗ trợ rộng rãi bởi các trình duyệt web hiện đại và đã trở thành một công cụ phổ biến để tạo ra bố cục web đáp ứng.

**Những lợi ích chính của Flexbox:**

**Bố cục linh hoạt:** Flexbox cho phép bạn dễ dàng phân phối không gian giữa các phần tử và thay đổi kích thước của chúng dựa trên kích thước của khung nhìn hoặc nội dung của chúng.

**Hỗ trợ chiều ngang và chiều dọc:** Bạn có thể sử dụng Flexbox để sắp xếp các phần tử theo chiều ngang (ví dụ: thanh điều hướng) hoặc chiều dọc (ví dụ: nội dung của một thẻ section).

**Căn chỉnh dễ dàng:** Flexbox cung cấp các tùy chọn căn chỉnh linh hoạt để sắp xếp các phần tử trong một container theo ý muốn.

**Giảm thiểu mã CSS:** Flexbox có thể giúp bạn giảm thiểu lượng mã CSS cần viết để tạo ra bố cục phức tạp.

**Các thành phần chính của Flexbox:**

**Flex container:** Phần tử cha chứa các phần tử con được sắp xếp bằng Flexbox.

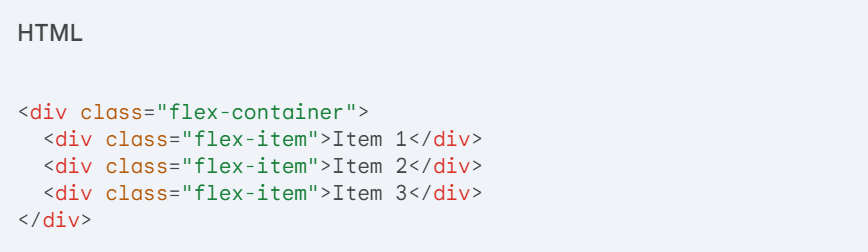
**Flex items:** Các phần tử con bên trong Flex container.

**Flex direction:** Xác định hướng chính của việc sắp xếp các phần tử (row, column, row-reverse, column-reverse).

**Justify-content:** Xác định cách phân phối không gian dọc theo trục chính của Flex container.

**Align-items:** Xác định cách căn chỉnh các phần tử theo trục ngang của Flex container.

**Flex-grow, flex-shrink, flex-basis:** Xác định cách phân phối không gian còn lại và điều chỉnh kích thước của các phần tử Flex item.





## 

## 3.10. Grid

**Grid Layout** là một mô hình bố cục mạnh mẽ trong CSS cho phép bạn sắp xếp các phần tử HTML thành một lưới hai chiều linh hoạt. Nó cung cấp khả năng kiểm soát chính xác vị trí và kích thước của các phần tử, giúp bạn tạo ra các bố cục phức tạp và đáp ứng một cách dễ dàng.

**Những lợi ích chính của Grid Layout:**

**Bố cục hai chiều:** Cho phép bạn sắp xếp các phần tử theo cả chiều ngang và chiều dọc, linh hoạt hơn so với các phương pháp bố cục truyền thống.

**Kiểm soát chính xác:** Cung cấp các tùy chọn để xác định chính xác vị trí, kích thước và khoảng cách giữa các phần tử.

**Đáp ứng tốt:** Giúp bạn tạo ra bố cục phù hợp với mọi kích thước màn hình và thiết bị.

**Giảm thiểu mã CSS:** Có thể giúp bạn giảm thiểu lượng mã CSS cần viết để tạo ra bố cục phức tạp.

**Các thành phần chính của Grid Layout:**

**Grid container:** Phần tử cha chứa các phần tử con được sắp xếp bằng Grid Layout.

**Grid items:** Các phần tử con bên trong Grid container.

**Grid lines:** Các đường ngang và dọc chia Grid container thành các ô (cell).

**Grid tracks:** Các đường ngang (grid rows) và dọc (grid columns) xác định kích thước và số lượng ô trong Grid container.

**Grid placement:** Xác định vị trí của các Grid item trong Grid container bằng cách tham chiếu các grid lines hoặc grid tracks

## 3.11. CSS3

CSS3 là phiên bản thứ ba và cũng là mới nhất của Cascading Style Sheets (CSS), ngôn ngữ được sử dụng để định dạng các trang web. So với CSS2, CSS3 bổ sung thêm nhiều tính năng mới mạnh mẽ và hữu ích, giúp bạn tạo các trang web đẹp, hiện đại và tương tác hơn.

## Tính năng nổi bật của CSS3:

**Selectors:** Cung cấp nhiều bộ chọn mới để nhắm mục tiêu chính xác các phần tử HTML, bao gồm:

:not(): Chọn các phần tử không khớp với bộ chọn con trong ngoặc.

:nth-child(): Chọn các phần tử con dựa trên vị trí (ví dụ: phần tử con thứ hai).

:pseudo-class nâng cao như :hover (khi di chuột qua), :focus (khi có tiêu điểm).

**Rounded Corners:** Cho phép tạo các góc bo tròn cho các phần tử mà không cần sử dụng hình ảnh.

**Border Images:** Sử dụng hình ảnh để tạo đường viền cho các phần tử.

**Backgrounds:** Cung cấp nhiều thuộc tính mới để tạo ảnh nền đa dạng hơn, bao gồm:

background-size: Xác định kích thước của ảnh nền.

background-position: Xác định vị trí của ảnh nền.

background-repeat: Xác định cách lặp lại ảnh nền.

**Gradients:** Tạo hiệu ứng chuyển màu mượt mà giữa hai hoặc nhiều màu sắc.

**Shadows:** Thêm bóng đổ cho các phần tử để tạo chiều sâu và hiệu ứng 3D.

**Fonts:** Cung cấp các thuộc tính mới để tùy chỉnh font chữ, bao gồm:

font-face: Cho phép tải và sử dụng các font chữ tùy chỉnh.

text-shadow: Thêm bóng đổ cho văn bản.

**2D Transforms:** Xoay, dịch chuyển và thu phóng các phần tử.

**3D Transforms:** Tạo hiệu ứng 3D thực sự cho các phần tử.

**Transitions:** Thay đổi các thuộc tính CSS theo thời gian mượt mà.

**Animations:** Tạo các hoạt ảnh phức tạp cho các phần tử.

**Multiple Columns:** Hiển thị nội dung thành nhiều cột.

## Lưu ý:

Không phải tất cả các trình duyệt đều hỗ trợ đầy đủ các tính năng của CSS3. Kiểm tra tính tương thích trước khi sử dụng các tính năng mới.

Sử dụng hợp lý các tính năng của CSS3 để tránh website quá nặng và chậm.

## 3.12. SCSS

SCSS là một ngôn ngữ tiền xử lý CSS (CSS preprocessor) được viết trên cú pháp Ruby. Nó cho phép bạn viết mã CSS theo cách ngắn gọn, dễ đọc và dễ bảo trì hơn so với CSS thông thường.

**Lợi ích của SCSS:**

**Biến:** Giúp bạn lưu trữ và sử dụng lại các giá trị CSS, ví dụ như màu sắc, font chữ, kích thước.

**Mixin:** Giúp bạn tạo các nhóm thuộc tính CSS có thể sử dụng lại cho nhiều phần tử.

**Nesting:** Giúp bạn lồng các phần tử CSS vào nhau, tạo code dễ đọc và dễ quản lý hơn.

**Functions:** Giúp bạn thực hiện các phép toán và logic trong CSS, ví dụ như tính toán kích thước, tạo hiệu ứng.

**Operators:** Giúp bạn sử dụng các toán tử toán học như cộng, trừ, nhân, chia trong CSS.

**Inheritance:** Giúp bạn kế thừa các thuộc tính CSS từ phần tử cha sang phần tử con.

## 3.13. SASS

SASS, viết tắt của **Syntactically Awesome Style Sheets**, là một preprocessor cho CSS, giúp bạn viết CSS dễ dàng hơn, duy trì và mở rộng hơn. Nó cung cấp cho bạn các tính năng như biến, mixin, nested rules (quy tắc lồng nhau), hàm, các phép toán và nhiều hơn nữa, giúp bạn viết CSS ngắn gọn, tổ chức và dễ đọc hơn.

**Dưới đây là một số lợi ích chính của việc sử dụng SASS:**

**Học CSS dễ dàng hơn:** SASS sử dụng cú pháp giống với cú pháp của nhiều ngôn ngữ lập trình khác, giúp người mới dễ dàng học CSS.

**Mã CSS ngắn gọn và dễ đọc:** Các tính năng như biến, mixin và nested rules giúp bạn viết mã CSS ngắn gọn và dễ hiểu hơn.

**Tổ chức và duy trì mã tốt hơn:** SASS giúp bạn tổ chức mã CSS thành các phần riêng biệt, dễ dàng duy trì và mở rộng.

**Tính năng mạnh mẽ:** SASS cung cấp các tính năng như hàm và các phép toán, giúp bạn viết CSS linh hoạt và mạnh mẽ hơn.

**Kiểm soát phiên bản tốt hơn:** Bạn có thể sử dụng các công cụ như Sass compilers để kiểm soát phiên bản của mã SASS và compile nó thành CSS tương thích với trình duyệt.

**Tuy nhiên, SASS cũng có một số nhược điểm:**

**Phụ thuộc vào công cụ compile:** Bạn cần sử dụng một công cụ compile để chuyển đổi SASS thành CSS trước khi hiển thị trên trình duyệt.

**Có thể làm chậm quá trình phát triển:** Quá trình compile SASS có thể làm chậm quá trình phát triển, đặc biệt là trên các dự án lớn.

**Không phải trình duyệt nào cũng hỗ trợ trực tiếp:** Trình duyệt không hiểu SASS, bạn cần compile nó thành CSS trước khi trình duyệt có thể hiển thị nội dung trang web.

**Tổng kết:**

SASS là một công cụ mạnh mẽ có thể giúp bạn viết CSS dễ dàng hơn, duy trì và mở rộng hơn. Nếu bạn đang tìm cách cải thiện khả năng viết CSS của mình, SASS là một lựa chọn tốt. Tuy nhiên, hãy cân nhắc những nhược điểm của nó trước khi sử dụng.

## Kết luận chương 3

CSS và CSS3 đóng vai trò quan trọng trong việc tạo ra các trang web đẹp mắt, thu hút và dễ sử dụng. CSS cung cấp các công cụ cơ bản để định dạng giao diện trang web, bao gồm màu sắc, phông chữ, bố cục, và hiệu ứng. CSS3 là phiên bản nâng cao của CSS, bổ sung nhiều tính năng mới giúp bạn tạo ra các trang web với giao diện hiện đại, đáp ứng và tương tác tốt hơn..

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

## 4.1. Ý tưởng của Website

**1. Xác định thị trường mục tiêu:**

**Bạn muốn bán đồng hồ cho ai?** Nam, nữ, trẻ em, hay cả ba?

**Mức độ thu nhập của họ như thế nào?**

**Họ có sở thích và nhu cầu gì?**

**2. Lựa chọn sản phẩm:**

**Bạn sẽ bán loại đồng hồ nào?**  Đồng hồ điện tử,đồng hồ mạ vàng , bạc hay những chiếc đồng hồ xa xỉ được giới nhà giàu tiêu dùng?

**Bạn sẽ bán thương hiệu nào?** Trong nước hay quốc tế?

**Mức giá sản phẩm như thế nào?** Phổ thông hay cao cấp?

**3. Xây dựng website:**

**Chọn giao diện đẹp mắt và dễ sử dụng.** Giao diện cần phù hợp với thị hiếu của khách hàng mục tiêu và thể hiện được phong cách của thương hiệu bạn.

**Cung cấp thông tin chi tiết về sản phẩm.** Bao gồm mô tả, hình ảnh, video , và hướng dẫn sử dụng.

**Có hình ảnh và video chất lượng cao.** Hình ảnh và video cần rõ ràng, sắc nét và thể hiện được sản phẩm một cách chân thực nhất.

**Cung cấp nhiều phương thức thanh toán và vận chuyển.** Đáp ứng nhu cầu đa dạng của khách hàng.

**Tối ưu hóa website cho SEO.** Giúp website dễ dàng được tìm kiếm trên Google và các công cụ tìm kiếm khác.

**4. Marketing và quảng cáo:**

**SEO website để tăng thứ hạng trên công cụ tìm kiếm.**

**Chạy quảng cáo trên mạng xã hội và Google Ads.**

**Tham gia các chương trình khuyến mãi.**

**Hợp tác với các influencer.**

**Tạo nội dung blog và video về đồng hồ.**

**5. Chăm sóc khách hàng:**

**Cung cấp dịch vụ khách hàng tốt.** Giải đáp thắc mắc, tư vấn sản phẩm, và xử lý khiếu nại của khách hàng một cách nhanh chóng và hiệu quả.

**Duy trì mối quan hệ tốt với khách hàng.** Gửi email cảm ơn, chương trình tri ân khách hàng, và thông tin về các sản phẩm mới.

## 4.2. Xây dựng bố cục của trang Web

**1. Xác định thị trường mục tiêu:**

Bạn muốn bán đồng hồ cho ai? Nam, nữ, trẻ em?

Mức độ thu nhập của họ như thế nào?

Họ có sở thích và nhu cầu gì?

**2. Lựa chọn sản phẩm:**

Bạn sẽ bán loại đồng hồ nào? Đồng hồ điện tử,đồng hồ rolex,đồng hồ mạ?

Bạn sẽ bán thương hiệu nào?

Mức giá sản phẩm như thế nào?

**3. Xây dựng website:**

Chọn giao diện đẹp mắt và dễ sử dụng.

Cung cấp thông tin chi tiết về sản phẩm.

Có hình ảnh và video chất lượng cao.

Cung cấp nhiều phương thức thanh toán và vận chuyển.

**4. Marketing và quảng cáo:**

SEO website để tăng thứ hạng trên công cụ tìm kiếm.

Chạy quảng cáo trên mạng xã hội và Google Ads.

Tham gia các chương trình khuyến mãi.

**5. Chăm sóc khách hàng:**

Cung cấp dịch vụ khách hàng tốt.

Giải quyết khiếu nại nhanh chóng.

Duy trì mối quan hệ tốt với khách hàng.

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

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## 4.5. Kết quả đạt được

A screenshot of a website

Description automatically generatedA group of watches with black dials

Description automatically generatedA white background with red text

Description automatically generatedA white background with black text

Description automatically generated

## Kết luận chương 4

Web bán đồng hồ là kênh bán hàng hiệu quả, mang lại lợi ích cho cả người bán và người mua. Với sự phát triển của công nghệ, web bán đồng hồ ngày càng trở nên phổ biến và là lựa chọn hàng đầu cho việc mua sắm đồng hồ.Tăng doanh thu dễ dàng giới thiệu sản phẩm, chương trình khuyến mãi, và thu hút khách hàng mua hàng.Cải thiện hiệu quả kinh doanh theo dõi dữ liệu bán hàng, hành vi khách hàng, và tối ưu hóa chiến lược kinh doanh.

**KẾT LUẬN**

Web bán đồng hồ có nhiều hướng phát triển khác nhau, mỗi hướng đều có ưu điểm và nhược điểm riêng. Doanh nghiệp cần lựa chọn hướng phát triển phù hợp với mục tiêu kinh doanh và khả năng của mình.Cung cấp đa dạng sản phẩm và lựa chọn cho khách hàng,tăng khả năng cạnh tranh trên thị trường,tiết kiệm chi phí marketing và quảng cáo.Nhược điểm khó kiểm soát chất lượng sản phẩm và dịch vụ mức độ cạnh tranh cao giữa các nhà bán lẻ,khó tạo dựng thương hiệu riêng.Hướng phát triển phù hợp cho web bán đồng hồ phụ thuộc vào nhiều yếu tố, bao gồm.Loại sản phẩm,thị trường mục tiêu,nguồn lực tài chính,khả năng cạnh tranh

**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.