**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 website bán gấu bông)**

**Giảng viên hướng dẫn: ThS. Phạm Hà Thanh**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSV** | **Họ và Tên** | **Lớp** |
| **1** | **1771020525** | **Đinh Thị Quỳnh Như** | **CNTT 17-13** |
| **2** | **1771020297** | **Nguyễn Tử Minh Hoàng** | **CNTT 17-13** |

**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 website bán tay cầm chơi game)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020525 | Đinh Thị Quỳnh Như | 01/04/2005 |  |  |
| 2 | 1771020297 | Nguyễn Tử Minh Hoàng | 24/08/2005 |  |  |

### 

|  |  |
| --- | --- |
| CÁN BỘ CHẤM THI 1 | CÁN BỘ CHẤM THI 2 |
|
|  |  |

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.Lời nói đầu tóm tắt về HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.窗体底端

HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra và định dạng các trang web. HTML được sử dụng để xác định cấu trúc của một trang web bằng cách sử dụng các thẻ và các phần tử. Mỗi thẻ HTML định nghĩa một phần cụ thể của nội dung trang web, như tiêu đề, đoạn văn bản, hình ảnh, liên kết, bảng, và các phần tử khác. Các trình duyệt web sử dụng HTML để hiển thị nội dung của một trang web cho người dùng. HTML thường kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra trải nghiệm web toàn diện và tương tác.

**MỤC LỤC**

**CHƯƠNG 1: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

[1.1.Các khái niệm cơ bản 8](#_Toc16825)

[1.2. Phân biệt Web tĩnh, Web động 11](#_Toc25147)

[1.3. Một số thuật ngữ 12](#_Toc3897)

[1.4. Một số công cụ dùng thiết kế Web 15](#_Toc21151)

**CHƯƠNG 2: HTML & HTML5**

[2.1. Tổng quan về HTML 18](#_Toc1913)

[2.2. Cấu trúc tổng quát trang HTML 19](#_Toc4412)

[2.3. Các thẻ HTML thông dụng 20](#_Toc19122)

[2.4. Các thẻ tạo biểu mẫu (form) 23](#_Toc27394)

[2.5. Một số thẻ HTML đặc biệt 23](#_Toc28971)

2.6.HTML5.............................................................................................................................26

**CHƯƠNG 3: CSS và CSS3**

[3.1. CSS là gì? 29](#_Toc463)

[3.2. Cú pháp CSS 29](#_Toc7480)

[3.3. Áp dụng CSS vào trang HTML 31](#_Toc32491)

3.4.Selcector............................................................................................................................32

[3.5. Đơn vị đo lường CSS 35](#_Toc10809)

[3.6. Kế thừa thuộc tính 36](#_Toc12935)

[3.7. Các nhóm thuộc tính trong CSS 38](#_Toc18599)

[3.8. Float & Clear 39](#_Toc20509)

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

[4.1. Tại sao cần lập Website? 41](#_Toc5994)

[4.2. Xây dựng bố cục của trang Web 41](#_Toc14049)

[4.3.Kết quả của quá trình tạo web 45](#_Toc3349)

**MỤC LỤC HÌNH ẢNH**

**CHƯƠNG 1: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

Hình 1.1: Thiết kế web

Hình 1.2: Website

Hình 1.3: Máy chủ web

Hình 1.4: Hosting

Hình 1.5: Giao diện người dùng (UI)

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

Hình 4.1: Trang chính của web

Hình 4.2: submenu

Hình 4.3: Phần login

Hình 4.4: phần main

Hình 4.5: Phần chăm sóc khách hàng

Hình 4.6: Shop Gấu bông web

**MỤC LỤC BẢNG**

**(Nếu có)**

Bảng 1.1: So sánh Web tĩnh và Web động............................................................................11

**BẢNG CÁC TỪ VIẾT TẮT**

**(Nếu có)**

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

**CHƯƠNG 1: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

**1.1.Các khái niệm cơ bản**

* **Thiết kế web:** Là quá trình tạo ra giao diện và cấu trúc của trang web, bao gồm bố cục, hình ảnh, màu sắc, typography, và chức năng.



***Hình 1.1: Thiết kế web***

* **Triển khai hệ thống phần mềm:** Là quá trình cài đặt, cấu hình và đưa hệ thống phần mềm vào hoạt động.

Triển khai hệ thống phần mềm

* **Website:** Là tập hợp các trang web được liên kết với nhau, có chung tên miền và được lưu trữ trên một hoặc nhiều máy chủ web.



***Hình 1.2: Website***

* **Trang web:** Là một tài liệu được lưu trữ trên máy chủ web, có thể truy cập được qua mạng internet bằng trình duyệt web.
* **Máy chủ web:** Là máy tính lưu trữ các trang web và cung cấp chúng cho người dùng khi có yêu cầu.

******

***Hình 1.3: Máy chủ web***

* **Tên miền:** Là tên dễ nhớ được sử dụng để truy cập website thay cho địa chỉ IP của máy chủ web.
* **Hosting:** Là dịch vụ cung cấp không gian lưu trữ cho website trên máy chủ web

******

***Hình 1.4: Hosting***

* **Giao diện người dùng (UI):** Là phần mà người dùng trực tiếp tương tác với hệ thống, bao gồm các yếu tố như bố cục, hình ảnh, màu sắc, typography, và chức năng.

******

***Hình 1.5: Giao diện người dùng (UI)***

* **Trải nghiệm người dùng (UX):** Là cảm nhận của người dùng khi sử dụng hệ thống, bao gồm sự dễ sử dụng, hiệu quả và mức độ hài lòng.

**Một số thuật ngữ khác:**

* **HTML:** Ngôn ngữ lập trình được sử dụng để tạo ra cấu trúc của trang web.
* **CSS:** Ngôn ngữ lập trình được sử dụng để định kiểu cho trang web, bao gồm màu sắc, phông chữ, và bố cục.
* **JavaScript:** Ngôn ngữ lập trình được sử dụng để thêm chức năng cho trang web, chẳng hạn như tương tác với người dùng và xử lý dữ liệu.
* **PHP:** Ngôn ngữ lập trình được sử dụng để tạo ra các trang web động.
* **ASP.NET:** Ngôn ngữ lập trình được sử dụng để tạo ra các trang web động trên nền tảng Microsoft.
* **Ruby on Rails:** Khung ứng dụng web được sử dụng để tạo ra các trang web động.

**\*Kết luận**

- Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực quan trọng trong công nghệ thông tin. Việc hiểu rõ các khái niệm cơ bản về hai lĩnh vực này sẽ giúp bạn có thể bắt đầu học tập và làm việc trong lĩnh vực này.

**1.2. Phân biệt Web tĩnh, Web động**

**Web tĩnh:**

* Là website có nội dung được lưu trữ dưới dạng các tập tin HTML, CSS và JavaScript tĩnh.
* Khi người dùng truy cập website, máy chủ web sẽ gửi các tập tin này đến trình duyệt web của họ.
* Nội dung của website không thay đổi thường xuyên.
* Ví dụ: website giới thiệu công ty, website portfolio cá nhân.

**Web động:**

* Là website có nội dung được tạo ra động khi người dùng truy cập.
* Máy chủ web sẽ sử dụng các ngôn ngữ lập trình như PHP, ASP.NET, hoặc Ruby on Rails để tạo ra nội dung dựa trên yêu cầu của người dùng.
* Nội dung của website có thể thay đổi thường xuyên.
* Ví dụ: website bán hàng trực tuyến, website tin tức, website mạng xã hội.

**Bảng 1.1: So sánh Web tĩnh và Web động:**

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Web tĩnh** | **Web động** |
| **Nội dung** | Tĩnh, được lưu trữ dưới dạng tập tin | Động, được tạo ra khi có yêu cầu |
| **Thay đổi nội dung** | Khó khăn, cần chỉnh sửa tập tin | Dễ dàng, thông qua hệ thống quản trị nội dung |
| **Ví dụ** | Website giới thiệu công ty, portfolio cá nhân | Website bán hàng trực tuyến, website tin tức |
| **Ưu điểm** | Chi phí thấp, dễ triển khai, bảo mật cao | Tính tương tác cao, dễ dàng cập nhật nội dung |
| **Nhược điểm** | Khó thay đổi nội dung, không có tính tương tác | Chi phí cao hơn, bảo mật thấp hơn |

**Lựa chọn Web tĩnh hay Web động:**

Lựa chọn Web tĩnh hay Web động phụ thuộc vào nhu cầu sử dụng của bạn:

* **Web tĩnh:** Phù hợp cho website có nội dung ít thay đổi, chi phí thấp và yêu cầu bảo mật cao.
* **Web động:** Phù hợp cho website có nội dung thay đổi thường xuyên, cần tính tương tác cao và dễ dàng cập nhật nội dung.

**Kết luận:**

Web tĩnh và Web động là hai loại website khác nhau với những ưu điểm và nhược điểm riêng.

**1.3. Một số thuật ngữ**

***1. Hosting:***

* Dịch vụ cung cấp không gian lưu trữ cho website trên máy chủ web.
* Giống như thuê một mảnh đất trên internet để chứa các tập tin của website của bạn.
* Các nhà cung cấp dịch vụ hosting thường cung cấp nhiều gói hosting khác nhau với các mức giá và dung lượng lưu trữ khác nhau.

***2. Tên miền (Domain Name):***

* Tên dễ nhớ được sử dụng để truy cập website thay cho địa chỉ IP của máy chủ web.
* Giống như địa chỉ nhà của bạn trên internet, giúp người dùng dễ dàng tìm thấy website của bạn.
* Tên miền cần được đăng ký với một nhà cung cấp dịch vụ tên miền để có thể sử dụng.

***3. Web Server Name:***

* Tên của máy chủ web lưu trữ website.
* Thường trùng với tên miền của website, nhưng không phải lúc nào cũng vậy.
* Ít được sử dụng thông thường bởi người dùng cuối vì họ thường truy cập website thông qua tên miền.

***4. HomePage:***

* Trang chủ của website, thường là trang đầu tiên mà người dùng nhìn thấy khi truy cập website.
* Giống như trang chào đón của website, thường chứa các thông tin chính, menu điều hướng và liên kết đến các trang khác trong website.

***5. Website:***

* Tập hợp các trang web được liên kết với nhau, có chung tên miền và được lưu trữ trên một hoặc nhiều máy chủ web.
* Giống như một bộ sưu tập các tài liệu điện tử được tổ chức theo một cấu trúc nhất định.
* Website có thể được sử dụng cho nhiều mục đích khác nhau, như giới thiệu thông tin, bán hàng trực tuyến, cung cấp dịch vụ hoặc giao tiếp.

***6. WebServer:***

* Phần mềm chạy trên máy chủ web để xử lý các yêu cầu truy cập website.
* Giống như một người trung gian giữa người dùng và website, nhận yêu cầu từ người dùng và gửi lại nội dung của website.
* Các web server phổ biến bao gồm Apache, Nginx, IIS.

***7. URL (Uniform Resource Locator):***

* Địa chỉ của trang web trên mạng internet.
* Giống như địa chỉ đường dẫn đến một trang cụ thể trên website.
* URL bao gồm các thành phần như giao thức (http, https), tên miền, đường dẫn thư mục và tên file.

***8. Browser (Trình duyệt web):***

* Phần mềm được sử dụng để truy cập website.
* Giống như một công cụ để hiển thị nội dung của website trên màn hình máy tính của bạn.
* Các trình duyệt web phổ biến bao gồm Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

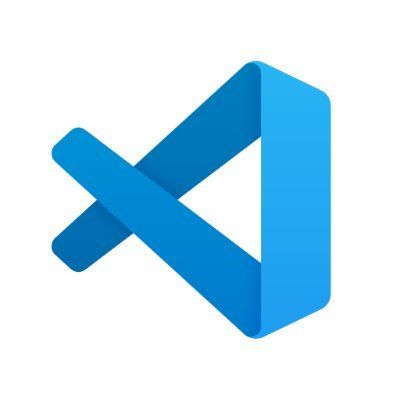
**\*Ngoài ra, còn một số thuật ngữ khác thường gặp:**

* **HTML:** Ngôn ngữ lập trình được sử dụng để tạo ra cấu trúc của trang web.
* **CSS:** Ngôn ngữ lập trình được sử dụng để định kiểu cho trang web, bao gồm màu sắc, phông chữ, và bố cục.
* **JavaScript:** Ngôn ngữ lập trình được sử dụng để thêm chức năng cho trang web, chẳng hạn như tương tác với người dùng và xử lý dữ liệu.
* **Database:** Cơ sở dữ liệu lưu trữ thông tin của website, thường được sử dụng trong các website động.

**1.4. Một số công cụ dùng thiết kế Web**

- Có rất nhiều công cụ khác nhau được sử dụng để thiết kế web, mỗi công cụ có ưu điểm và nhược điểm riêng. Dưới đây là một số công cụ phổ biến:

1. ***Visual Studio Code:***



* Trình soạn thảo mã miễn phí và mã nguồn mở được phát triển bởi Microsoft.
* Hỗ trợ nhiều ngôn ngữ lập trình, bao gồm HTML, CSS, JavaScript, PHP, Python, Java, C++, C#,...
* Có nhiều tính năng hữu ích cho việc thiết kế web, như tô sáng cú pháp, tự động hoàn thành mã, gỡ lỗi,...
* Phù hợp cho cả người mới bắt đầu và lập trình viên chuyên nghiệp.

1. ***Adobe Dreamweaver:***



* Phần mềm thiết kế web chuyên nghiệp được phát triển bởi Adobe.
* Cung cấp giao diện trực quan WYSIWYG (What You See Is What You Get) giúp người dùng dễ dàng thiết kế web mà không cần biết nhiều về mã.
* Hỗ trợ nhiều tính năng nâng cao, như quản lý nội dung, tích hợp với các hệ thống quản trị nội dung (CMS),...
* Phù hợp cho người dùng muốn thiết kế web một cách chuyên nghiệp và hiệu quả.

**3*. Sublime Text:***

* Trình soạn thảo mã trả phí với nhiều tính năng nâng cao.
* Hỗ trợ nhiều ngôn ngữ lập trình, bao gồm HTML, CSS, JavaScript, PHP, Python, Java, C++, C#,...
* Giao diện đơn giản và dễ sử dụng.
* Tốc độ xử lý nhanh chóng.
* Phù hợp cho lập trình viên có kinh nghiệm.

***4.Notepad++:***



* Trình soạn thảo mã miễn phí và nhẹ dành cho hệ điều hành Windows.
* Hỗ trợ nhiều ngôn ngữ lập trình, bao gồm HTML, CSS, JavaScript, PHP, Python, Java, C++, C#,...
* Giao diện đơn giản và dễ sử dụng.
* Phù hợp cho người mới bắt đầu học lập trình web.

**\*Ngoài ra, còn có một số công cụ khác như:**

* **WYSIWYG editors:** Các trình soạn thảo web trực quan cho phép người dùng thiết kế web mà không cần biết mã.
* **CMS (Content Management System):** Hệ thống quản trị nội dung giúp người dùng dễ dàng tạo và quản lý website.
* **Web frameworks:** Khung ứng dụng web giúp người dùng phát triển website một cách nhanh chóng và hiệu quả.

**\*Kết luận chương 1:**

* Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực quan trọng trong công nghệ thông tin.
* Việc hiểu rõ các khái niệm cơ bản về hai lĩnh vực này sẽ giúp bạn có thể bắt đầu học tập và làm việc trong lĩnh vực này.
* Lựa chọn loại Web và công cụ thiết kế Web phù hợp phụ thuộc vào nhu cầu và kỹ năng và ứng dụng nó tốt hơn

**CHƯƠNG 2: HTML & HTML5**

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

**2.1. Tổng quan về HTML**

**HTML** là viết tắt của **Hyper Text Markup Language**, là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo ra cấu trúc và định dạng cho trang web. HTML sử dụng các thẻ để xác định các phần tử khác nhau của trang web, như tiêu đề, đoạn văn, hình ảnh, bảng, liên kết,.

**2.1.1.Chức năng chính của HTML:**

* Xác định cấu trúc của trang web.
* Định dạng nội dung trang web.
* Tạo liên kết giữa các trang web.
* Chèn hình ảnh, video, âm thanh vào trang web.
* Tạo biểu mẫu để thu thập dữ liệu từ người dùng.

**2.1.2.Ưu điểm của HTML:**

* Dễ học và sử dụng.
* Miễn phí và mã nguồn mở.
* Hỗ trợ bởi tất cả các trình duyệt web.
* Có thể kết hợp với các ngôn ngữ lập trình khác để tạo ra các trang web phức tạp.

**2.1.3.Nhược điểm của HTML:**

* Không có khả năng xử lý dữ liệu động.
* Khó tạo ra các trang web có giao diện phức tạp.

**Để học HTML, bạn có thể tham khảo các tài liệu sau:**

* Sách giáo khoa, tài liệu hướng dẫn.
* Các khóa học trực tuyến.
* Các cộng đồng online về thiết kế web và lập trình.

**\*Kết luận:**

HTML là ngôn ngữ cơ bản để tạo trang web. Việc học HTML là bước đầu tiên để bạn bước vào lĩnh vực thiết kế web và lập trình web.

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

Cấu trúc tổng quát của trang HTML bao gồm:

***1. Dòng khai báo DOCTYPE:***

Khai báo phiên bản HTML được sử dụng

***2. Thẻ*** *<html>****:***

Khai báo bắt đầu trang HTML.

***3. Thẻ*** *<head>****:***

Chứa thông tin về trang web, như tiêu đề, metadata.

***4. Thẻ*** *<title>****:***

Chứa tiêu đề của trang web.

***5. Thẻ*** *<body>****:***

Chứa nội dung chính của trang web.

***6. Thẻ*** *</html>****:***

Khai báo kết thúc trang HTML.

Dưới đây là ví dụ về cấu trúc tổng quát của trang HTML:

**Giải thích các phần:**

* **Dòng khai báo DOCTYPE:** Khai báo phiên bản HTML5 được sử dụng.
* **Thẻ**<html>**:** Khai báo bắt đầu trang HTML.
* **Thẻ**<head>**:** Chứa thông tin về trang web, như tiêu đề.
* **Thẻ**<title>**:** Chứa tiêu đề của trang web.
* **Thẻ**<body>**:** Chứa nội dung chính của trang web, bao gồm:
  + **Thẻ**<h1>**:** Tạo tiêu đề cấp 1.
  + **Thẻ**<p>**:** Tạo đoạn văn.
* **Thẻ**</html>**:** Khai báo kết thúc trang HTML.

**Lưu ý:**

* Thẻ <html> và </html> là thẻ bắt đầu và kết thúc trang HTML.
* Thẻ <head> và </head> là thẻ bắt đầu và kết thúc phần đầu trang HTML.
* Thẻ <body> và </body> là thẻ bắt đầu và kết thúc phần thân trang HTML.
* Thẻ <title> là thẻ khai báo tiêu đề của trang web.
* Nội dung trang web được đặt trong thẻ <body>.

**2.3. Các thẻ HTML thông dụng**

- Các thẻ xử lý văn bản

<h1> đến <h6>: Tạo tiêu đề với các cấp độ khác nhau, <h1> là cấp độ lớn nhất, <h6> là cấp độ nhỏ nhất.

<p>: Tạo đoạn văn.

<strong>: Làm đậm văn bản.

<em>: In nghiêng văn bản.

<br>: Dòng xuống.

<code>: Định dạng văn bản như mã code.

<pre>: Định dạng văn bản với định dạng cố định.

<kbd>: Định dạng văn bản như phím bấm.

- Các thẻ tạo bảng

***1. Thẻ*** *<table>****:***

Khai báo bắt đầu bảng.

***2. Thẻ*** *<tr>****:***

Khai báo bắt đầu hàng trong bảng.

***3. Thẻ*** *<td>****:***

Khai báo bắt đầu ô trong bảng.

***4. Thẻ*** *</th>****:***

Khai báo bắt đầu ô tiêu đề trong bảng (thường dùng cho hàng đầu tiên của bảng).

***5. Thẻ*** *</table>****:***

Khai báo kết thúc bảng.

- Thẻ liên kết

1. ***Thẻ*** *<a>*

được sử dụng để tạo liên kết trong HTML

- Các thẻ đa phương tiện

* **Chèn hình ảnh:**
* **Thẻ**<img>**:**
* src: Thuộc tính bắt buộc, chỉ ra đường dẫn đến hình ảnh.
* alt: Thuộc tính mô tả nội dung của hình ảnh (hiển thị khi không tải được hình ảnh hoặc người dùng di chuột qua hình ảnh).

**Chèn video:**

* **Thẻ**<video>**:**
* controls: Thuộc tính hiển thị thanh điều khiển video.
* source: Thuộc tính chỉ ra đường dẫn đến video.
* type: Thuộc tính chỉ ra định dạng video

**Chèn âm thanh:**

* **Thẻ**<audio>**:**
* controls: Thuộc tính hiển thị thanh điều khiển âm thanh.
* source: Thuộc tính chỉ ra đường dẫn đến âm thanh.
* type: Thuộc tính chỉ ra định dạng âm thanh.
* **Kết luận:**
* Các thẻ đa phương tiện giúp chèn hình ảnh, video, âm thanh vào trang web, tạo ra nội dung phong phú và thu hút người dùng. Việc sử dụng các thẻ đa phương tiện một cách hợp lý sẽ giúp bạn tạo ra các trang web đẹp mắt và hiệu quả.

- Thẻ tạo khung

***1. Thẻ*** *<frameset>****:***

Khai báo bắt đầu bộ khung.

***2. Thẻ*** *<frame>****:***

Khai báo mỗi khung trong bộ khung.

* rows: Thuộc tính chỉ ra kích thước các hàng trong bộ khung.
* cols: Thuộc tính chỉ ra kích thước các cột trong bộ khung.
* src: Thuộc tính chỉ ra trang web được hiển thị trong mỗi khung.

**2.4. Các thẻ tạo biểu mẫu (form)**

***1. Thẻ*** *<form>****:***

Khai báo bắt đầu biểu mẫu.

***2. Thẻ*** *<input>****:***

Tạo các trường nhập liệu trong biểu mẫu.

***3. Thẻ*** *<label>****:***

Nhãn cho các trường nhập liệu.

***4. Thẻ*** *<button>****:***

Tạo nút submit hoặc reset cho biểu mẫu.

**Ngoài ra, bạn có thể sử dụng các thẻ khác để tạo biểu mẫu:**

* textarea: Tạo trường nhập liệu đa dòng.
* select: Tạo danh sách lựa chọn.
* option: Tạo các lựa chọn trong danh sách.
* fieldset: Nhóm các trường nhập liệu liên quan.
* legend: Tiêu đề cho nhóm trường nhập liệu.

**2.5. Một số thẻ HTML đặc biệt**

## Một số thẻ HTML đặc biệt

Ngoài các thẻ HTML cơ bản được sử dụng phổ biến, còn có một số thẻ đặc biệt có chức năng riêng biệt và hữu ích trong việc thiết kế và phát triển trang web. Dưới đây là một số ví dụ:

***1. Thẻ*** *DOCTYPE****:***

* Khai báo phiên bản HTML của trang web.
* Giúp trình duyệt web hiểu và hiển thị trang web chính xác.
* Ví dụ: <!DOCTYPE html>

***2. Thẻ*** *html****:***

* Khai báo bắt đầu trang web HTML.
* Chứa tất cả các nội dung của trang web.
* Ví dụ: <html>

***3. Thẻ*** *head****:***

* Chứa thông tin meta về trang web, như tiêu đề, mô tả, charset, favicon.
* Không hiển thị trực tiếp trên trang web.
* Ví dụ: <head>

***4. Thẻ*** *title****:***

* Xác định tiêu đề của trang web.
* Hiển thị trên thanh tiêu đề của trình duyệt web.
* Ví dụ: <title>Trang web của tôi</title>

***5. Thẻ*** *meta****:***

* Cung cấp thông tin meta về trang web, như charset, author, keywords, description.
* Không hiển thị trực tiếp trên trang web.
* Ví dụ: <meta charset="UTF-8">

***6. Thẻ*** *body****:***

* Chứa nội dung chính của trang web, như văn bản, hình ảnh, video, biểu mẫu.
* Hiển thị trực tiếp trên trang web.
* Ví dụ: <body>

***7. Thẻ*** *script****:***

* Chèn mã JavaScript vào trang web.
* Cho phép thêm chức năng tương tác cho trang web.
* Ví dụ: <script src="script.js"></script>

***8. Thẻ*** *link****:***

* Tạo liên kết đến các tài nguyên bên ngoài, như stylesheet, favicon.
* Ví dụ: <link rel="stylesheet" href="style.css">

***9. Thẻ*** *style****:***

* Định dạng nội dung trang web bằng CSS.
* Ví dụ: <style>body {color: red;}</style>

***10. Thẻ*** *img****:***

* Chèn hình ảnh vào trang web.
* Ví dụ: <img src="image.jpg" alt="Đây là một bức ảnh">

***11. Thẻ*** *video****:***

* Chèn video vào trang web.
* Ví dụ: <video controls><source src="video.mp4" type="video/mp4"></video>

***12. Thẻ*** *audio****:***

* Chèn âm thanh vào trang web.
* Ví dụ: <audio controls><source src="audio.mp3" type="audio/mp3"></audio>

***13. Thẻ*** *iframe****:***

* Chèn một trang web khác vào trang web hiện tại.
* Ví dụ: <iframe src="https://www.google.com"></iframe>

**2.6. HTML5**

* HTML5 là gì?

HTML5 là phiên bản thứ năm của ngôn ngữ lập trình HTML (HyperText Markup Language) được sử dụng để tạo cấu trúc và nội dung cho các trang web. HTML5 được phát triển bởi World Wide Web Consortium (W3C) với mục tiêu tạo ra một ngôn ngữ web mạnh mẽ, linh hoạt và dễ sử dụng hơn so với các phiên bản trước đây.

**Một số tính năng chính của HTML5 bao gồm:**

* **Cấu trúc ngữ nghĩa:** HTML5 cung cấp các thẻ mới để xác định rõ ràng ý nghĩa của nội dung trên trang web, ví dụ như header, article, section, aside và footer.
* **Đa phương tiện:** HTML5 hỗ trợ tích hợp các nội dung đa phương tiện như video và âm thanh trực tiếp vào trang web mà không cần đến plugin hay phần mềm bổ sung.
* **Vẽ đồ họa:** HTML5 cung cấp API Canvas cho phép vẽ đồ họa và hình ảnh trực tiếp trên trang web.
* **Lưu trữ dữ liệu cục bộ:** HTML5 cung cấp API Web Storage cho phép lưu trữ dữ liệu cục bộ trên trình duyệt web.
* **Kết nối WebSocket:** HTML5 cho phép tạo kết nối hai chiều giữa trình duyệt web và máy chủ, giúp tạo ra các ứng dụng web thời gian thực.

**HTML5 mang lại nhiều lợi ích cho việc phát triển web:**

* **Dễ sử dụng:** HTML5 có cú pháp đơn giản và dễ học hơn so với các phiên bản trước đây.
* **Hiệu quả:** HTML5 giúp tối ưu hóa hiệu suất của trang web.
* **Khả năng tương thích:** HTML5 được hỗ trợ bởi hầu hết các trình duyệt web hiện đại.
* **Tính linh hoạt:** HTML5 có thể được sử dụng để phát triển nhiều loại ứng dụng web khác nhau.

**\*HTML5 là một công nghệ quan trọng trong việc phát triển web hiện đại. Việc sử dụng HTML5 giúp bạn tạo ra các trang web đẹp mắt, hiệu quả và dễ sử dụng.**

* Cú pháp HTML5

Cú pháp HTML5 dựa trên cú pháp XML, với một số điểm khác biệt để đơn giản hóa việc sử dụng. Dưới đây là một số điểm chính về cú pháp HTML5:

***1. Cấu trúc cơ bản:***

* Một trang HTML5 bắt đầu vớiDOCTYPE <!DOCTYPE html>.
* Tiếp theo là thẻ <html> bao gồm toàn bộ nội dung của trang web.
* Bên trong thẻ <html> có thẻ <head> chứa thông tin meta về trang web, như tiêu đề, mô tả, charset, favicon.
* Sau thẻ <head> là thẻ <body> chứa nội dung chính của trang web, như văn bản, hình ảnh, video, biểu mẫu.

***2. Thẻ HTML:***

* Thẻ HTML được viết dưới dạng cặp thẻ mở và thẻ đóng.
* Ví dụ: <p> là thẻ mở cho đoạn văn bản, </p> là thẻ đóng cho đoạn văn bản.
* Một số thẻ HTML có thể được sử dụng tự đóng, ví dụ: <img> là thẻ tự đóng cho hình ảnh.

***3. Thuộc tính:***

* Thẻ HTML có thể có các thuộc tính để định dạng hoặc cung cấp thêm thông tin về nội dung.
* Ví dụ: thẻ <img> có thể có thuộc tính src để chỉ ra đường dẫn đến hình ảnh.

***4. Giá trị thuộc tính:***

* Giá trị img thuộc tính được đặt trong ngoặc kép.
* Ví dụ: < src="image.jpg">

***5. Nhận xét:***

* Nhận xét được sử dụng để ghi chú trong mã HTML.
* Nhận xét bắt đầu với ``.

***6. Dấu cách:***

* Dấu cách được sử dụng để định dạng mã HTML cho dễ đọc.
* Tuy nhiên, HTML5 không yêu cầu bắt buộc sử dụng dấu cách.

***7. Chữ hoa và chữ thường:***

* Cú pháp HTML5 không phân biệt chữ hoa và chữ thường.
* Khai báo DOCTYPE trong HTML5
* Thẻ ngữ nghĩa
* Webform

**\*Kết luận chương 2**

* giúp ta xác định được HTML có lịch sử và lợi ích như nào .cho ta thấy cách nó hoạt động các cú pháp có thể tạo ra 1 trang wed hoản trình (từ âm thanh đến hình ảnh ) và có thể tạo ra đa phương tiện HTML5 là ngôn ngữ web hiện đại với nhiều tính năng mạnh mẽ và linh hoạt.
* Cú pháp HTML5 đơn giản và dễ học.
* Việc sử dụng HTML5 giúp bạn tạo ra các trang web đẹp mắt, hiệu quả và dễ sử dụng.

**CHƯƠNG 3: CSS và CSS3**

(Cascading Style Sheets)

**3.1. CSS là gì?**

**CSS** là viết tắt của **Cascading Style Sheets**, là ngôn ngữ lập trình dùng để định dạng và tạo kiểu cho các trang web. CSS giúp kiểm soát cách hiển thị của các phần tử HTML như màu sắc, phông chữ, bố cục, v.v.

**Ví dụ:**

* Thay đổi màu sắc của tiêu đề trang web.
* Định dạng phông chữ cho văn bản.
* Tạo bố cục hai cột cho trang web.
* Thêm viền cho hình ảnh.
* Hiển thị ảnh động.

**Lợi ích của CSS:**

* Giúp trang web đẹp mắt và thu hút hơn.
* Giúp người dùng dễ đọc và hiểu nội dung trang web.
* Giúp kiểm soát bố cục trang web.
* Giúp trang web tải nhanh hơn.

**Có ba cách để áp dụng CSS vào trang HTML:**

* **Nhúng trực tiếp:** Sử dụng thuộc tính style trong thẻ HTML.
* **CSS nội tuyến:** Sử dụng thẻ <style> trong phần <head> của trang HTML.
* **Tập tin CSS riêng:** Tạo tập tin .css riêng và liên kết nó với trang HTML sử dụng thẻ <link>.

**3.2. Cú pháp CSS**

Cú pháp CSS bao gồm:

***1. Chọn phần tử:***

Sử dụng **selector** để xác định phần tử HTML muốn định dạng. Có nhiều loại selector khác nhau như:

* **Universal selector:** Chọn tất cả các phần tử (\*).
* **Type selector:** Chọn phần tử dựa trên tên thẻ (h1, p, v.v.).
* **Identity selector:** Chọn phần tử dựa trên ID (#id).
* **Class selector:** Chọn phần tử dựa trên class (class).
* **Descendant selector:** Chọn phần tử con (div p).
* **Child selector:** Chọn phần tử con trực tiếp (div > p).
* **Adjacent sibling selector:** Chọn phần tử liền kề (h1 + p).
* **Attribute selector:** Chọn phần tử dựa trên thuộc tính ([href]).
* **Pseudo class selector:** Chọn phần tử dựa trên trạng thái (a:hover).
* **Group selector:** Chọn một nhóm phần tử (h1, h2, h3).

***2. Khai báo thuộc tính:***

Sử dụng tên thuộc tính và giá trị để định dạng phần tử.

***3. Khối quy tắc:***

Bao gồm selector và các khai báo thuộc tính được bao quanh bởi dấu ngoặc nhọn { }.

**Ví dụ:**

CSS

h1 {

color: red;

font-size: 24px;

* Thứ tự của các quy tắc CSS rất quan trọng. Quy tắc được khai báo sau sẽ ghi đè lên quy tắc được khai báo trước.
* Có thể sử dụng nhiều selector cho một quy tắc CSS.
* Có thể sử dụng dấu phẩy để phân cách các selector trong cùng một quy tắc CSS.

**3.3. Áp dụng CSS vào trang HTML**

Có ba cách để áp dụng CSS vào trang HTML:

***1. Nhúng trực tiếp:***

Sử dụng thuộc tính style trong thẻ HTML để khai báo các quy tắc CSS trực tiếp cho phần tử đó.

***2. CSS nội tuyến:***

Sử dụng thẻ <style> trong phần <head> của trang HTML để khai báo các quy tắc CSS cho toàn bộ trang

***3. Tập tin CSS riêng:***

Tạo tập tin .css riêng để khai báo các quy tắc CSS và liên kết nó với trang HTML sử dụng thẻ <link> trong phần <head> của trang HTML.

**Lợi ích của việc sử dụng tập tin CSS riêng:**

* Giúp code HTML gọn gàng và dễ đọc hơn.
* Giúp dễ dàng quản lý và cập nhật các quy tắc CSS.
* Giúp tăng hiệu suất tải trang web.

Chú ý

* Thứ tự áp dụng CSS: Nhúng trực tiếp > CSS nội tuyến > Tập tin CSS riêng.
* Khi sử dụng tập tin CSS riêng, cần lưu ý đến đường dẫn tương đối của tập tin

**3.4. Selectors**

**Selectors trong CSS** là những công cụ giúp bạn **xác định các phần tử HTML** mà bạn muốn định dạng. Chúng hoạt động như những con trỏ để xác định các phần tử mà bạn muốn áp dụng quy tắc CSS.

**Có nhiều loại selectors khác nhau trong CSS:**

***1. Universal Selector (\*):*** Selector này chọn **tất cả các phần tử** trên trang. Nó ít được sử dụng trong thực tế vì việc quản lý các styles sẽ khó khăn nếu áp dụng cho mọi thứ.

***2. Type Selector (ví dụ: h1, p, div):*** Selector này chọn các phần tử dựa trên **tên thẻ HTML** của chúng. Ví dụ: h1 chọn tất cả các heading level 1, và p chọn tất cả các paragraph.

***3. ID Selector (#id):*** Selector này chọn một phần tử có **ID attribute** duy nhất. ID giống như tên được gán cho các phần tử cụ thể, và selector sử dụng ký hiệu # theo sau là tên ID.

**Ví dụ:** <h1 id="main-title">Đây là tiêu đề chính</h1>

***4. Class Selector (.class):*** Selector này chọn các phần tử có **class attribute** cụ thể. Class có thể được gán cho nhiều phần tử, cho phép bạn định dạng chúng tương tự nhau. Selector sử dụng ký hiệu . theo sau là tên class.

**Ví dụ:** <p class="important">Đây là văn bản quan trọng.</p>

***5. Descendant Selector (ancestor element > descendant element):*** Selector này chọn các phần tử là **con cháu** của một phần tử khác. Selector sử dụng ký hiệu > để phân biệt phần tử tổ tiên và phần tử con cháu.

**Ví dụ:** div > p chọn tất cả các paragraph là con trực tiếp của div.

***6. Child Selector (ancestor element > child element):*** Selector này tương tự như descendant selector, nhưng nó chỉ chọn **con trực tiếp** của phần tử tổ tiên được chỉ định.

**Ví dụ:** #content > p chọn tất cả các paragraph là con trực tiếp của element có ID "content".

***7. Adjacent Sibling Selector (element1 + element2):*** Selector này chọn một phần tử **ngay sau** một phần tử khác. Selector sử dụng ký hiệu + để phân biệt hai loại phần tử.

**Ví dụ:** h1 + p chọn tất cả các paragraph ngay sau một h1.

***8. Attribute Selector ([attribute])*:** Selector này chọn các phần tử dựa trên **sự hiện diện hoặc giá trị** của một attribute cụ thể. Selector sử dụng ngoặc vuông [] để bao quanh tên attribute và tùy chọn là một toán tử so sánh và giá trị.

**Ví dụ:** [href] chọn tất cả các phần tử có attribute href (như liên kết), và [href="#top"] chọn tất cả các phần tử có attribute href liên kết đến element có ID "top".

***9. Pseudo-Class Selector (element:pseudo-class):*** Selector này chọn các phần tử dựa trên **trạng thái hoặc hành vi** của chúng. Chúng sử dụng dấu hai chấm : theo sau là tên pseudo-class. Các ví dụ phổ biến bao gồm:

* :hover: Áp dụng styles khi người dùng di chuột qua một phần tử.
* :active: Áp dụng styles khi người dùng nhấp vào một phần tử.
* :focus: Áp dụng styles khi một phần tử được chọn (ví dụ: khi một trường form được chọn).

***10. Group Selector (selector1, selector2, ...):*** Selector này cho phép bạn chọn **nhiều phần tử** cùng một lúc, được phân cách bằng dấu phẩy.

**Ví dụ:** h1, h2, h3 chọn tất cả các heading level 1, 2 và 3.

Bằng cách hiểu và sử dụng hiệu quả các loại selectors khác nhau. có thể chọn chính xác các phần tử mà bạn muốn định dạng trong trang web của mình, tạo ra trải nghiệm trực quan và thân thiện với người dùng.

1. Universal selector

**Universal selector (bộ chọn vạn năng)** trong CSS là ký tự \* dùng để **chọn tất cả các phần tử HTML** trên trang web, bất kể tên thẻ, ID hay class của chúng

**Trường hợp sử dụng:**

* **Khôi phục về kiểu mặc định:** Universal selector có thể được sử dụng kết hợp với từ khóa initial để đưa tất cả các kiểu trên các phần tử về kiểu mặc định của trình duyệt. Điều này hữu ích để thiết lập nền tảng cho việc tạo kiểu cho tất cả các phần tử.
* **Kiểu tạm thời:** Để kiểm tra nhanh hoặc demo, bạn có thể sử dụng universal selector để áp dụng các kiểu tạm thời cho tất cả các phần tử trên trang.

**Tuy nhiên, việc sử dụng quá nhiều universal selector trong code thực tế thường không được khuyến khích.** Lý do:

* **Tính đặc thù (specificity):** Universal selector có tính đặc thù rất thấp, nghĩa là nó dễ dàng bị ghi đè bởi các selector có tính đặc thù cao hơn như type selector, ID selector và class selector. Điều này có thể gây khó khăn trong việc quản lý và kiểm soát các kiểu.
* **Tính bảo trì:** Sử dụng universal selector có thể dẫn đến code khó bảo trì hơn, vì khó theo dõi các kiểu được áp dụng ở đâu, đặc biệt trong các dự án lớn.
* **Hiệu suất:** Mặc dù tác động đến hiệu suất thường không đáng kể, nhưng việc sử dụng universal selector quá nhiều có thể làm chậm quá trình render của trình duyệt do cần phải kiểm tra nó với tất cả các phần tử.

**Các lựa chọn thay thế cho universal selector:**

* **Group selector:** Sử dụng dấu phẩy để chọn một nhóm các type selector cụ thể, ví dụ h1, p, span.
* **Descendant selector:** Kết hợp type selector với descendant selector để chọn các phần tử cụ thể trong cấu trúc, ví dụ div p.
* **Khôi phục về kiểu mặc định:** Sử dụng các framework hoặc thư viện CSS cung cấp các kiểu reset được định sẵn để thiết lập nền tảng cho tất cả các phần tử.

Mặc dù universal selector có thể là một công cụ hữu ích trong các tình huống cụ thể, nhưng nên sử dụng nó một cách tiết kiệm và cân nhắc các phương pháp thay thế để có khả năng bảo trì và kiểm soát tốt hơn các kiểu CSS

b) Type selector

c) Identity selector

d) Class selector

e) Descendant selector

f) Child selector

g)Adjacent sibling selector

h) Attribute selector

i) Pseudo class selector

k) Group selector

**3.5. Đơn vị đo lường CSS**

**Đơn vị đo lường** là yếu tố quan trọng trong việc xác định kích thước, vị trí và các thuộc tính khác của các phần tử HTML trong CSS. Có nhiều loại đơn vị đo lường khác nhau trong CSS, mỗi loại có mục đích sử dụng riêng:

***1. Đơn vị tuyệt đối:***

* **px (pixel):** Đơn vị phổ biến nhất, dựa trên mật độ pixel của màn hình.
* **cm (centimet):** Đơn vị dựa trên kích thước thực tế, không phụ thuộc vào màn hình.
* **mm (milimet):** Tương tự như cm, nhưng nhỏ hơn.
* **in (inch):** Đơn vị đo lường inch.

***2. Đơn vị tương đối:***

* **em:** Kích thước dựa trên font-size của phần tử cha.
* **rem:** Kích thước dựa trên font-size của element gốc (html).
* **vh (viewport height):** % của chiều cao viewport (phần hiển thị của trang web).
* **vw (viewport width):** % của chiều rộng viewport.

***3. Đơn vị khác:***

* **% (percent):** Tỷ lệ phần trăm của một giá trị khác.
* **deg (degree):** Đơn vị đo góc.
* **s (second):** Đơn vị đo thời gian.

**Lựa chọn đơn vị đo lường:**

Việc lựa chọn đơn vị đo lường phù hợp phụ thuộc vào mục đích sử dụng:

* **Kích thước cố định:** Sử dụng px cho các element có kích thước cố định, không thay đổi theo kích thước màn hình.
* **Kích thước tương đối:** Sử dụng em hoặc rem cho các element có kích thước thay đổi theo font-size.
* **Responsive design:** Sử dụng vh và vw cho các layout responsive, co giãn theo kích thước màn hình.

**\*Lưu ý:**

* Nên sử dụng đơn vị đo lường phù hợp để đảm bảo tính nhất quán và dễ dàng bảo trì code.
* Tránh sử dụng quá nhiều đơn vị đo lường khác nhau trong cùng một trang web.
* Có thể sử dụng các công cụ chuyển đổi đơn vị để chuyển đổi giữa các đơn vị đo lường khác nhau.

**3.6. Kế thừa thuộc tính**

**Kế thừa thuộc tính** là một tính năng quan trọng trong CSS cho phép các thuộc tính được **truyền từ phần tử cha sang phần tử con**. Nhờ vậy, bạn có thể định dạng một loạt các phần tử một cách dễ dàng và hiệu quả, mà không cần phải định dạng từng phần tử riêng lẻ.

**Có hai loại kế thừa thuộc tính:**

***1. Kế thừa trực tiếp:***

* Thuộc tính được truyền từ phần tử cha **trực tiếp** sang phần tử con.
* Ví dụ: nếu bạn đặt font-size: 16px cho element div, thì tất cả các element con của div cũng sẽ có font-size là 16px, trừ khi được định dạng riêng.

***2. Kế thừa gián tiếp:***

* Thuộc tính được truyền từ phần tử cha **gián tiếp** qua nhiều cấp.
* Ví dụ: nếu bạn đặt font-size: 16px cho element html, thì tất cả các element con của html (bao gồm cả các element con của các element con) đều sẽ có font-size là 16px, trừ khi được định dạng riêng.

**\*Lưu ý:**

* Không phải tất cả các thuộc tính CSS đều được kế thừa.
* Thuộc tính được kế thừa có thể bị ghi đè bởi thuộc tính được định nghĩa riêng cho phần tử con.
* Thứ tự ưu tiên: Thuộc tính được định nghĩa cho phần tử con > Thuộc tính được kế thừa từ phần tử cha.

**Kế thừa thuộc tính giúp:**

* Giảm thiểu lượng code cần viết.
* Giúp code dễ đọc và dễ bảo trì hơn.
* Tạo sự nhất quán trong việc định dạng trang web.

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

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

* font-family: Chọn font chữ cho element.
* font-size: Kích thước font chữ.
* font-weight: Độ dày font chữ.
* font-style: Kiểu font chữ (normal, italic, oblique).
* text-align: Căn chỉnh văn bản (left, center, right, justify).
* text-decoration: Hiệu ứng trang trí văn bản (underline, overline, line-through).
* text-transform: Chuyển đổi chữ hoa/chữ thường (uppercase, lowercase, capitalize).
* letter-spacing: Khoảng cách giữa các chữ cái.
* word-spacing: Khoảng cách giữa các từ.
* line-height: Chiều cao dòng văn bản.

- Thuộc tính định dạng nền (background)

* background-color: Màu nền.
* background-image: Hình ảnh nền.
* background-repeat: Lặp lại hình ảnh nền (repeat, repeat-x, repeat-y, no-repeat).
* background-position: Vị trí hình ảnh nền.
* background-size: Kích thước hình ảnh nền.
* background-attachment: Cố định hình ảnh nền (scroll, fixed).

- Mô hình hộp (box model)

* width: Chiều rộng hộp.
* height: Chiều cao hộp.
* margin: Khoảng cách giữa hộp và các element xung quanh.
* padding: Khoảng cách giữa nội dung hộp và viền hộp.
* border: Viền hộp.

- Các thuộc tính định margin, padding, border

* margin-top: Khoảng cách trên của margin.
* margin-right: Khoảng cách phải của margin.
* margin-bottom: Khoảng cách dưới của margin.
* margin-left: Khoảng cách trái của margin.
* padding-top: Khoảng cách trên của padding.
* padding-right: Khoảng cách phải của padding.
* padding-bottom: Khoảng cách dưới của padding.
* padding-left: Khoảng cách trái của padding.
* border-width: Độ dày của viền.
* border-style: Kiểu viền (solid, dotted, dashed, double).
* border-color: Màu viền.

- Thuộc tính định dạng cách hiển thị (display)

* display: Kiểu hiển thị element (block, inline, inline-block, none).
* visibility: Hiển thị hoặc ẩn element (visible, hidden).

- Thuộc tính xác định vị trí (position)

* position: Vị trí element (static, relative, absolute, fixed).
* top: Vị trí trên của element.
* right: Vị trí phải của element.
* bottom: Vị trí dưới của element.
* left: Vị trí trái của element.

**3.8. Float & Clear**

**3.9. Flex**

**3.10. Grid**

**3.11. CSS3**

- CSS3 là gì?

- CSS3 Selectors

- CSS3 Rounded Corners

- CSS3 Border Images

- CSS3 Backgrounds

- CSS Gradients

- CSS3 Shadows

- CSS3 Fonts

- CSS3 2D Transforms

- CSS3 3D Transforms

- CSS3 Transitions

- CSS3 Multiple Columns

- CSS3 Animations

**3.12. SCSS**

**3.13. SASS**

**\*Kết luận chương 3**

CSS là ngôn ngữ lập trình quan trọng để tạo trang web đẹp mắt và thu hút. CSS giúp kiểm soát cách hiển thị của các phần tử HTML như màu sắc, phông chữ, bố cục, v.v.

CSS3 là phiên bản mới nhất của CSS với nhiều tính năng mới giúp tạo trang web hiện đại và phức tạp hơn.

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

**(Shop bán gấu bông)**

**4.1. Tại sao cần lập Website?**

- Dựa trên thực tiễn thị trường hiện tại và nhu cầu của nhiều bạn trẻ hiện tại muốn mua bán và trao đổi thêm về gầu bông cùng những món quà dành cho trẻ nhỏ của các bậc phụ huynh đii cùng đó là sự tiện dụng và giá cả hợp lý

- Chính vì những điều kiện cần và đủ đó đã thúc đẩy ... việc tạo ra một website chuyên dụng để tư vấn và giúp cho khách hàng có thêm kiến thức và sự lựa chọn khi mua săms online.

- Việc tạo ra web bán gấu bông thúc đẩy nhu cầu mua bán và trao đổi với khách hàng và tiênj dụng hơn trong việc mua bán sản phẩm và rút ngắn thời gian cho khách hàng

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

*Chủ đề: shop bán gấu bông*

*Sử dụng html,css, ngoài ra còn có java để đóng mở tag.*

**4.2.1. Phần mở đầu trang web**

****

***Hình 4.1: Trang chính của web***

* Phần mở đầu cho khách hàng biết tổng sơ bộ về shop:

+ thẻ <head> để tạo thanh trên cùng, trong đó gồm:

- thẻ <li> để tạo các màn hình như: Trang chủ, Sản Phẩm, Đặt hàng.

- thẻ <img class=”logo” để tạo avatar cho web shop bên trên cùng góc trái màn hình.

- thẻ <div> để tạo tên shop nằm ở góc phải màn hình.

-thẻ <img class=”banner” gắn liên kết ảnh để làm ảnh bìa cho shop thu hút khách hàng

- thẻ <input> để tạo thanh tìm kiếm và placeholder để tạo khoảng trống cho khách hàng có thể gõ và tìm kiếm sản phẩm cần mua.

* Mục **Sản phẩm**:

+thẻ <div><li><a> để làm menu hàng loạt tên sản phẩm hiện ra cho khách hàng biết được sơ bộ về mặt hàng có trong shop.

****

***Hình 4.2: submenu***

* Mục **Đặt Hàng:**

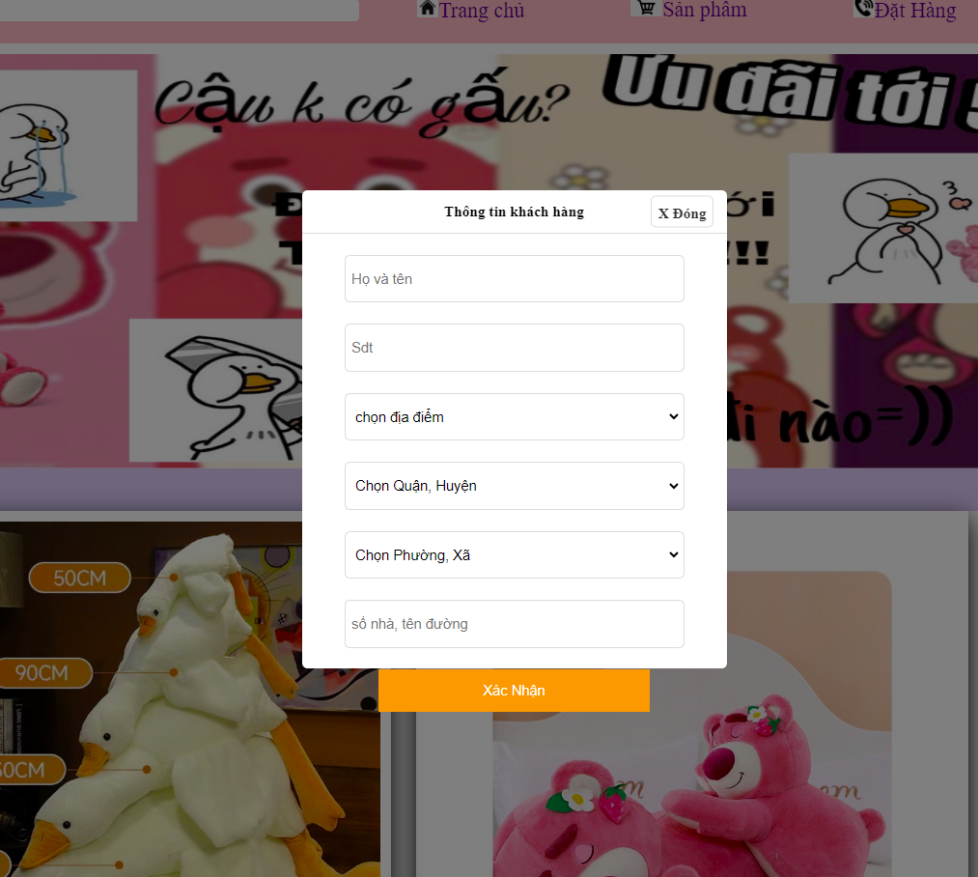
+thẻ <div> , thẻ <h1> <span> để thông báo đây là nơi khách hàng muốn mua hàng điền thông tin và login, logout.

+thẻ <input> kèm placeholder để khách hàng nhấp vào có thể điền được tên, sdt và địa chỉ cụ thể để nhận hàng.

+thẻ<select><option> dùng để cho khách hàng chọn địa điểm nơi nhận hàng mà có sẵn mà không cần nhập.

+thẻ <button> để tạo nút xác nhận điền thông tin khi làm xong để thoát ra khỏi màn hình chính.

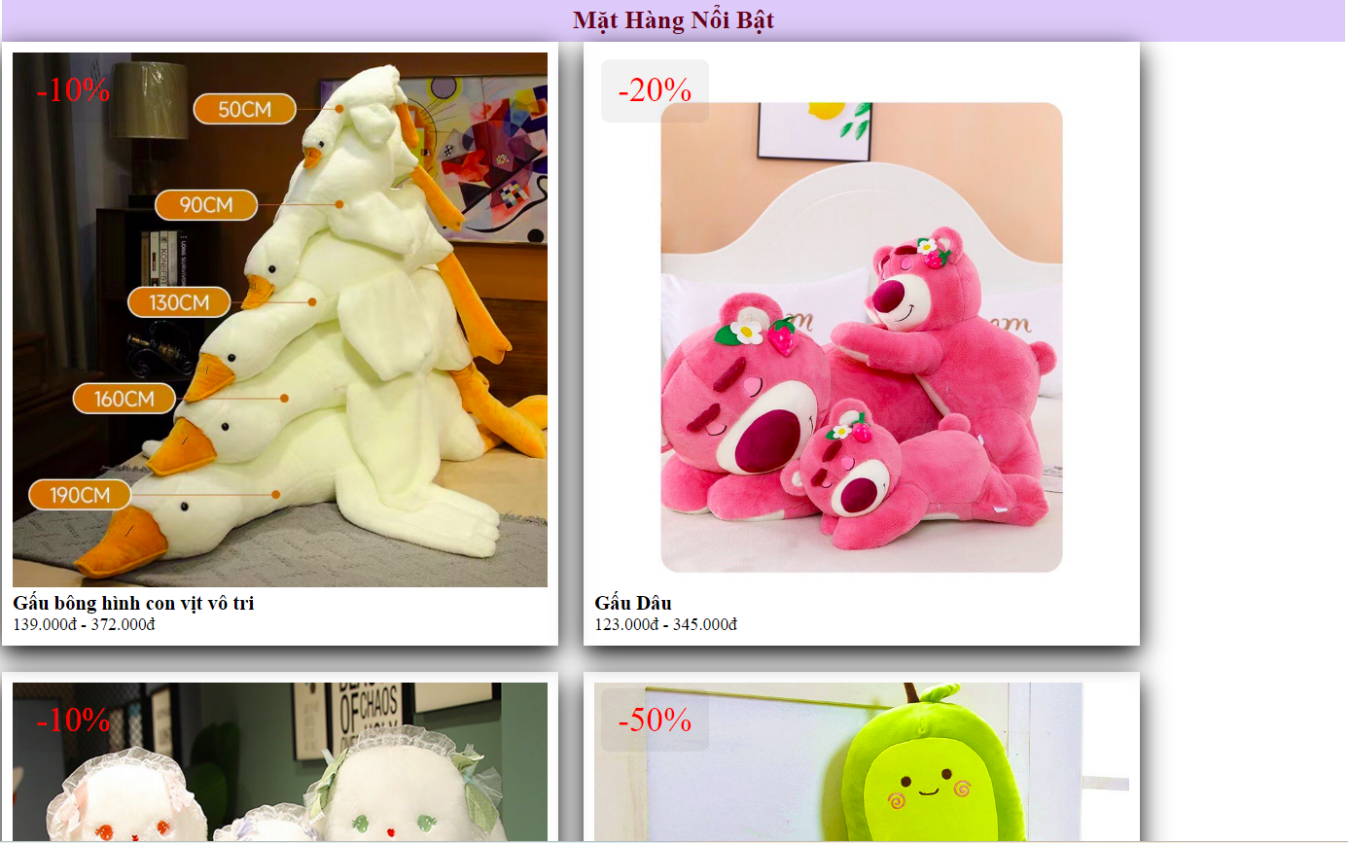
-Để có thể khiến khách hàng khi nhấp vào mục Đặt Hàng hiện ra mục điền thông tin thì ta phải dùng thêm java(.js)

******

***Hình 4.3: Phần login***

**4.2.2. phần thân của trang web**

* Phần tiếp theo là phần main của trang web để khách hàng biết được đến sản phẩm và giá thành:



***Hình 4.4: phần main***

-phần này nằm trong thẻ <section>,trong đó:

+thẻ <h2> làm phần heading để mở đầu cho phần main giúp cho khách hàng xác định được vị trí sản phẩm cần mua cũng như là mặt hàng bestseller của shop.

+thẻ <div> để tách biệt từng sản phẩm, cho biết tên sản phẩm, giá thành.

+thẻ <img> gắn liên kết từng ảnh để đại diện mặt hàng cần bán của shop.

+thẻ <span> để cho biết shop giảm giá từng sản phẩm.

**4.2.3. Phần cuối trang web**

****

***Hình 4.5: Phần chăm sóc khách hàng***

+dùng màu trong css đã liên kết để làm khung

+thẻ <footer> để làm phần kết thúc cho web

+có các thẻ <div> và <img> để làm các dòng hotline và đánh giá sản phẩm kèm icon đại diện

**4.3.Kết quả của quá trình tạo web**



***Hình 4.6: Shop Gấu bông web***

Chúng ta sẽ được một trang web shop gấu bông như trên (f5) chạy trên gg.

Tạo web bằng visual studio code sử dụng .html .css và .js

**KẾT LUẬN**

Tóm lại, HTML là một công nghệ quan trọng và phổ biến trong việc phát triển các ứng dụng web, cung cấp tính linh hoạt, tiêu chuẩn và tiềm năng mở rộng để tạo ra các trang web và ứng dụng web đa dạng và mạnh mẽ.Tạo ra các trang web chủ yếu nhằm phục vụ nhu cầu maketing của người bán và tìm kiếm của người mua hàng, phục vụ công việc thiết yếu của con người trong thời đại công nghệ hóa 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.