**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 QUẦN ÁO**

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

**Sinh viên thực hiện:**  **Nguyễn Tiến 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 QUẦN ÁO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1871020119 | Nguyễn Tiến Đại | 21/05/2006 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

(Nếu có)

MỤC LỤC

**(Đánh tự động)**

**MỤC LỤC HÌNH ẢNH**

**(Nếu có)**

**MỤC LỤC BẢNG**

**(Nếu có)**

**BẢNG CÁC TỪ VIẾT TẮT**

**(Nếu có)**

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

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

* 1. **Các khái niệm cơ bản**

- Chắc hẳn rằng chúng ta đã quá quen thuộc với các trang web cá nhân như blog, diễn đàn. Hoặc các trang thương mại điện tử của các công ty kinh doanh online có quy mô từ nhỏ đến lớn. Đó chính là các hình thức phổ biến mà chúng ta dễ nhìn thấy. Nhưng để có được trang Web hoàn chỉnh thì cần phải thực hiện nhiều bước một cách chỉnh chu. Thiết kế Website hoàn thiện cần phải thực hiện hai phần quan trọng là Web tĩnh và Web động.

**1.2. Phân biệt Web tĩnh, Web động**

(1) Website tĩnh là trang Web dùng hoàn toàn thuật ngữ HTML nên người dùng không thể tương tác. Chỉ khi tải trang từ máy chủ xuống thì trình duyệt sẽ dịch thuật mã để hiển thị nội dung trang. Để thực hiện thiết kế này cũng cần trải qua hai giai đoạn là Design và dựng giao diện.

Các Designer sẽ sử dụng phần mềm thiết kế đồ họa như Sketch, Ps, Dn,… để bắt đầu dựng giao diện trang Web dưới dạng hình ảnh.Để dựng giao diện thiết kế Website các lập trình viên sẽ dùng các ngôn ngữ HTML.



*Hình 1.1 : Thiết kế web dùng thuật ngữ HTML*

Sau khi thực hiện xong thì bạn sẽ có được trang Web cơ bản, tuy nhiên sẽ không sử dụng được. Nó chỉ cho phép ghi nhận, click hay chuyển trang mà chưa có khả năng lưu trữ hay xuất dữ liệu.

(2) Web động chính là trang Web đã có hệ thống thông tin để cung cấp dữ liệu cho Website. Lợi thế của trang web này so với web tĩnh chính là khối dữ liệu được lưu trữ tốt. Người dùng có thể nhanh chóng cập nhật và thêm các tiện ích một cách dễ dàng.

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

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

- Hosting là thuật ngữ chung để chỉ về dịch vụ lưu trữ dữ liệu online. Đây là không gian lưu trữ được chia nhỏ từ một máy chủ, giúp bạn đăng tải, lưu trữ dữ liệu. Trong đó phổ biến nhất là lưu trữ website và các ứng dụng trên internet.

- Tên miền là đường tắt đi đến server host website của bạn. Một tên miền (domain name) giống như một địa chỉ nhà vì đó là cách mọi người tìm thấy bạn trên World Wide Web. Đó là lý do tại sao chúng tôi gọi thanh trên đầu trình duyệt web là thanh địa chỉ – đó là nơi bạn nhập tên miền để tìm một trang web.

- Homepage hay còn gọi là “trang chủ”, là trang web đầu tiên và quan trọng nhất của một website. Đây là trang mặc định hiển thị khi người dùng truy cập vào địa chỉ tên miền của trang web mà không kèm các yếu tố khác.

Trang chủ là điểm xuất phát, nơi giới thiệu tổng quan về doanh nghiệp, sản phẩm/dịch vụ và dẫn dắt người dùng đến các phần nội dung khác. Trang chủ đóng vai trò như “bìa sách” của website, tạo ấn tượng ban đầu và định hướng trải nghiệm cho khách truy cập.

- Website đã trở nên rất phổ biến, hầu như trong lĩnh vực nào, ngành nghề nào cũng có website riêng biệt. Có thể kế đến những mẫu website phổ biến như website bán hàng, website giới thiệu, website tin tức, website bất động sản…Tuy nhiên không phải ai cũng am hiểu những kiến thức cơ bản về website

- Web Server là một máy tính lưu trữ các file thành phần tạo nên một Website (ví dụ: HTML, images, CSS, và file javascript...) và truyền chúng tới người dùng cuối. Web Server được kết nối đến Internet và truy cập thông qua một domain giống như mozilla.org.

- URL (Uniform Resource Locator, còn gọi là địa chỉ web ) là mã định danh duy nhất được sử dụng để định vị một tài nguyên trên internet. URL bao gồm nhiều phần -- bao gồm giao thức và tên miền -- cho trình duyệt web biết cách và nơi để truy xuất tài nguyên.

- Browser hay còn gọi là trình duyệt web, là một phần mềm cho phép bạn truy cập vào các trang web trên Internet. Browser có thể hiển thị các nội dung đa phương tiện như văn bản, hình ảnh, video, âm thanh hoặc các ứng dụng web. Browser cũng có thể tương tác với các máy chủ web để gửi và nhận dữ liệu.

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

- Visual Studio Code là một trong những trình soạn thảo mã nguồn rất phổ biến được các lập trình viên sử dụng. Với các ưu điểm nổi bật là sự nhanh chóng, nhẹ, hỗ trợ đa nền tảng cùng nhiều tính năng và là mã nguồn mở chính. Visual Studio Code ngày càng được ưa chuộng sử dụng, là lựa chọn hàng đầu của các lập trình viên. Bài viết sau đây sẽ cung cấp các thông tin để giúp bạn hiểu rõ hơn Visual Studio Code là gì cũng như các tính năng nổi bật của Visual Studio Code.

- Adobe Dreamweaver là một công cụ mạnh mẽ giúp người dùng thiết kế, quản lý trang web và ứng dụng web một cách hiệu quả. Nền tảng sở hữu hàng loạt tính năng ưu việt nhằm hoàn thành nhiệm vụ thiết kế theo yêu cầu của người dùng. Bài viết hôm nay giới thiệu đến bạn đọc những tiện ích đặc trưng của Adobe Dreamweaver

**1.5. Giới thiệu về emmet**

- Emmet (trước đây là Zen Coding  ) là một bộ [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) cho [trình soạn thảo văn bản](https://en.wikipedia.org/wiki/Text_editors) cho phép mã hóa và chỉnh sửa tốc độ cao ở [định dạng HTML](https://en.wikipedia.org/wiki/HTML) , [XML](https://en.wikipedia.org/wiki/XML) , [XSLT](https://en.wikipedia.org/wiki/XSLT) và các định dạng mã có cấu trúc khác thông qua [hỗ trợ nội dung](https://en.wikipedia.org/wiki/Content_assist) . Dự án được Vadim Makeev khởi xướng vào năm 2008 và tiếp tục được Sergey Chikuyonok và người dùng Emmet tích cực phát triển. Từ năm 2015, Mikael Geletsyan  chịu trách nhiệm về UX tại Emmet. Các công cụ đã được tích hợp vào một số [trình soạn thảo văn bản](https://en.wikipedia.org/wiki/Text_editor) phổ biến cũng như một số plug-in do nhóm Emmet phát triển và một số plug-in khác được triển khai độc lập. Tuy nhiên, Emmet chủ yếu độc lập với bất kỳ trình soạn thảo văn bản nào vì công cụ này hoạt động trực tiếp với văn bản chứ không phải với bất kỳ phần mềm cụ thể nào.

**KẾT LUẬN CHƯƠNG 1**

Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực cốt lõi trong ngành công nghệ thông tin, đóng vai trò quan trọng trong việc xây dựng và vận hành các ứng dụng và dịch vụ số hiện đại. Quá trình thiết kế web không chỉ đòi hỏi sự am hiểu về giao diện người dùng (UI) và trải nghiệm người dùng (UX), mà còn cần tích hợp chặt chẽ với các công nghệ lập trình, quản lý cơ sở dữ liệu, và bảo mật để tạo ra các sản phẩm trực quan, thân thiện và hiệu quả.Trong khi đó, triển khai hệ thống phần mềm là bước tiến quan trọng từ thiết kế đến thực tế, đảm bảo phần mềm hoạt động ổn định, đáp ứng đúng yêu cầu nghiệp vụ và khả năng mở rộng trong tương lai. Quá trình này yêu cầu sự phối hợp giữa các chuyên gia lập trình, quản lý dự án và kỹ sư hệ thống để hoàn thiện sản phẩm theo lộ trình tối ưu.

Tóm lại, việc kết hợp hài hòa giữa thiết kế web sáng tạo và triển khai phần mềm hiệu quả chính là nền tảng để xây dựng các giải pháp công nghệ toàn diện, đáp ứng nhu cầu ngày càng cao của xã hội số hóa.

**CHƯƠNG II: HTML & HTML5**

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

**Cách dùng các thẻ bằng emmet trong HTML**

**2.1. Tổng quan về HTML**

- HTML (Hypertext Markup Language) hay còn được gọi là ngôn ngữ đánh dấu siêu văn bản được thiết kế để tạo nên các trang web trên World Wide Web (www). HTML có chức năng giúp người dùng có thể định dạng, thiết kế cấu trúc các thành phần của một trang web hay ứng dụng, bao gồm tiêu đề (headings), liên kết (links), bảng (table), hình ảnh, đoạn văn bản và nhiều yếu tố khác.

- Một tài liệu HTML được hình thành từ các phần tử HTML (HTML Elements) và được quy định bằng các cặp thẻ (tags và attribute). Các cặp thẻ này sẽ được nhận biết bởi một dấu ngoặc nhọn và bao gồm thẻ mở <p> và thẻ đóng </p>

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

Mỗi trang web đều có cách thể hiện cấu trúc khác nhau, có trang 1 cột, có trang 2 và cũng có trang chứa nhiều cột, bên dưới đây chúng ta tham khảo một trang đơn giản sử dụng 2 cột để layout.

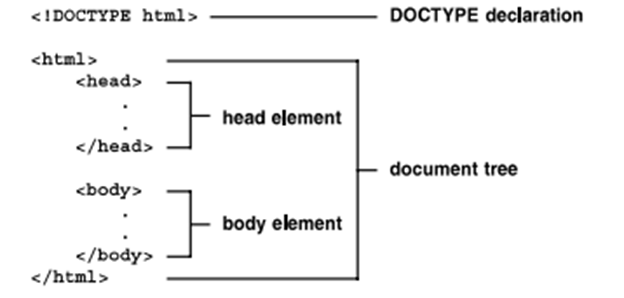
- Phần đầu: header, có thể chứa logo, câu slogan, các liên kết, các banner liên kết, các button, đoạn flash, hoặc các form ngắn như form tìm kiếm,...

- Phần liên kết toàn cục: global navigation, dùng để chứa các liên kết đến những trang quan trọng trong toàn bộ trang, trong phần này có thể chứa thêm các liên kết con (sub navigation).

- Phần thân của trang: page body, phần này chứa phần nội dung chính (content) và phần nội dung phụ (sidebar). Phần nội dung chính: content, phần này chứa nội dung chính cần thể hiện cho người dùng xem.

- Phần nội dung phụ: sidebar, phần này có thể chứa liên kết phụ của từng trang (local navigation), hoặc các banner chứa liên kết liên quan, hoặc có thể dùng để chứa các liên kết quảng cáo,...

- Phần cuối trang web: footer, phần này thường chứa phần liên hệ như: tên công ty, địa chỉ, số điện thoại, mail liên hệ,... và đặc biệt là copyright, hoặc có thể chứa các liên kết toàn trang, các banner liên kết,...



*Hình 2.1 : cấu trúc tổng quát*

**2.3. Các thẻ HTML thông dụng**

- Các thẻ xử lý văn bản : Tiêu đề (Headline) và đoạn văn bản (Paragraph) là hai thành phần mà bạn cần phân biệt để sau này sử dụng cho đúng. Bạn hãy hiểu rằng, tiêu đề văn bản nó giống như tên của từng chương trong một quyển sách vậy, nó sẽ được hiển thị to hơn nhằm nổi bật hơn.Trong HTML, các thẻ tiêu đề sẽ được định nghĩa bằng cặp thẻ <hn>, trong đó n là số tự nhiên từ 1 đến 6 tương ứng với từng cấp độ, số càng nhỏ thì cấp độ càng lớn. Để dễ hiểu hơn, hãy thử soạn thảo một tài liệu HTML như dưới đây.

- Các thẻ tạo bảng : Thẻ <table> dùng để xác định một cái bảng.

+ Thẻ <tr> dùng để xác định một hàng bên trong bảng.

+ Thẻ <th> dùng để xác định một ô (tiêu đề) bên trong hàng.

+ Thẻ <td> dùng để xác định một ô (bình thường) bên trong hàng.

- Thẻ liên kết : Thiết kế, xây dựng website là một công việc đòi hỏi sự tỉ mỉ và cẩn thận. Một website có thể bao gồm rất nhiều webpage khác nhau. Vì vậy việc kết nối các các webpage lại với nhau cũng là một khâu quan trọng. Để kết nối webpage đó lại với nhau thông thường chúng ta sử dụng các thẻ liên kết links.

- Các thẻ đa phương tiện : Trong HTML, các thẻ đa phương tiện (multimedia tags) được sử dụng để hiển thị và tương tác với các loại nội dung như hình ảnh, video, âm thanh, và các ứng dụng nhúng khác. Đây là các thành phần quan trọng giúp website trở nên sinh động và hấp dẫn hơn, thẻ hình ảnh, thẻ âm thanh, thẻ video, thẻ nhúng, thẻ đối tượng.

- Thẻ tạo khung : Thẻ tạo khung trong HTML thường được sử dụng để hiển thị nội dung từ các tài nguyên khác (như một trang web hoặc một tệp tài liệu) trong một khu vực được định sẵn trên trang web hiện tại. Trước đây, HTML sử dụng các thẻ khung như <frameset> và <frame>, nhưng chúng đã lỗi thời. Hiện nay, thẻ **<**iframe**>** là công cụ chính để tạo khung trong HTML hiện đại.

2.4. Các thẻ tạo biểu mẫu (form)

Biểu mẫu (form) trong HTML được sử dụng để thu thập và gửi dữ liệu từ người dùng đến máy chủ. Biểu mẫu thường bao gồm các trường nhập liệu, nút bấm, danh sách lựa chọn, và nhiều thành phần khác, được quản lý bởi thẻ chính là <form>. Các thẻ con trong biểu mẫu là:

+ Trường nhập liệu (<input>)

+ Nhãn cho trường nhập liệu(<label>)

+ Khu vực nhập liệu lớn (<textarea>)

+ Danh sách lựa chọn (<select> và <option>)

+ Nút bấm (<button> và <input type="button">)

+ Nút lựa chọn (<input type="radio"> và <input type="checkbox">)

+ Nút ẩn (<input type="hidden">)

2.5. Một số thẻ HTML đặc biệt

Trong HTML, ngoài các thẻ phổ biến như tiêu đề, đoạn văn, và thẻ tạo biểu mẫu, còn có một số thẻ đặc biệt cung cấp các chức năng cụ thể hoặc mang tính chất nâng cao. Những thẻ này thường được sử dụng để định nghĩa nội dung đặc biệt, cấu trúc nâng cao, hoặc hỗ trợ tính năng của trang web :

+ Thẻ <!DOCTYPE>

Không phải là một thẻ HTML thông thường, nhưng nó là khai báo cần thiết ở đầu tài liệu HTML.

Xác định phiên bản HTML mà trình duyệt nên sử dụng.

+ Thẻ <meta>

Cung cấp siêu dữ liệu về tài liệu HTML, chẳng hạn như mã hóa ký tự, tác giả, hoặc hướng dẫn cho trình duyệt.

+ Thẻ <link>

Dùng để liên kết tài liệu HTML với các tài nguyên bên ngoài, như tệp CSS, biểu tượng website.

+ Thẻ <script>

Chứa hoặc liên kết mã JavaScript, cho phép thêm các tính năng tương tác vào trang web.

+ Thẻ <noscript>

Cung cấp nội dung thay thế khi trình duyệt không hỗ trợ hoặc đã tắt JavaScript.

+ Thẻ <style>

Chứa mã CSS để định kiểu trực tiếp trong tài liệu HTML.

+ Thẻ <base>

Xác định URL cơ bản cho tất cả các liên kết tương đối trong tài liệu HTML.

+ Thẻ <iframe>

Cho phép nhúng nội dung từ một trang web khác vào trang hiện tại.

+ Thẻ <canvas>

Cung cấp một vùng vẽ đồ họa bằng JavaScript, thường dùng để tạo hình động hoặc game.

+ **Thẻ** <svg>

Dùng để vẽ đồ họa vector (Scalable Vector Graphics), thường cho biểu đồ, hình minh họa.

+ Thẻ <map> và <area>

Sử dụng để tạo bản đồ hình ảnh, giúp chia hình ảnh thành nhiều khu vực có thể nhấp.

+ Thẻ <picture>

Cung cấp hình ảnh đáp ứng, hiển thị phiên bản hình ảnh khác nhau tùy vào kích thước màn hình.

+ Thẻ <progress> và <meter>

**<**progress**>**: Hiển thị tiến trình một nhiệm vụ.

**<**meter**>**: Hiển thị giá trị trong một phạm vi nhất định.

+ Thẻ <details> và <summary>

**<**details**>**: Tạo vùng nội dung có thể thu gọn/mở rộng.

**<**summary**>**: Tóm tắt cho vùng nội dung đó.

+ Thẻ <template>

Định nghĩa mẫu nội dung không được hiển thị trực tiếp trong trình duyệt mà có thể sử dụng qua JavaScript.

+ Thẻ <slot> (dùng trong Web Components)

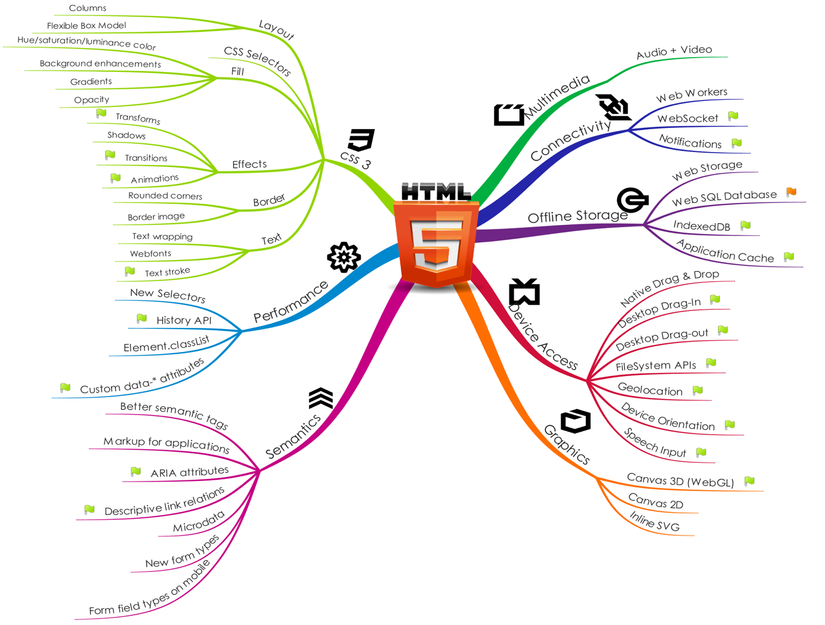
Xác định vùng nội dung động sẽ được nhúng vào một thành phần tùy chỉnh.

**2.6. HTML5**

* HTML5 là gì?

HTML5 là một ngôn ngữ cấu trúc và trình bày nội dung cho World Wide Web. Đây là phiên bản thứ 5 của ngôn ngữ HTML, được giới thiệu bởi World Wide Web Consortium (W3C). HTML5 vẫn sẽ giữ lại những đặc điểm cơ bản của HTML4 và bổ sung thêm các đặc tả nổi trội của XHTML, DOM cấp 2, đặc biệt là JavaScript.

Với sự ra đời của HTML5 cùng với các ứng dụng cần những hiệu ứng đồ họa và chuyển động đặc biệt, lập trình viên có thể sử dụng Canvas với kiểu bitmap hoặc SVG với kiểu vector. Không chỉ áp dụng cho việc thiết kế các trang web trực quan, HTML5 còn được áp dụng để tạo ra các thư viện đồ họa giúp tạo ra các ứng dụng đồ thị, game trong cả môi trường 2D và 3D như những ứng dụng trên desktop.



*Hình 2.2 : HTML 5*

* Cú pháp HTML5

Ngôn ngữ HTML5 có một cú pháp "custom" mà tương thích với tài liệu HTML4 và XHTML1 đã công bố trên Web, nhưng nó không tương thích với các tính năng riêng SGML của HTML4.HTML5 không có quy tắc cú pháp chung như XHTML, như chúng ta cần các tên thẻ là chữ thường, trích dẫn các thuộc tính của nó, một thuộc tính phải có một giá trị và để đóng tất cả các phần tử trống.Nhưng HTML5 có nhiều sự linh động hơn và sẽ hỗ trợ những điều sau:

+ Các tên thẻ chữ hoa

+ Trích dẫn là tùy ý cho các thuộc tính

+ Giá trị của thuộc tính là tùy ý

+ Đóng các phần tử trống là tùy ý

* Khai báo DOCTYPE trong HTML5

Khai báo DOCTYPE là một dòng khai báo đặc biệt ở đầu tài liệu HTML, dùng để xác định phiên bản của HTML mà trình duyệt cần sử dụng để hiển thị trang web. Trong HTML5, khai báo DOCTYPE đơn giản hơn nhiều so với các phiên bản trước đó.

Không phân biệt chữ hoa chữ thường: Dòng khai báo có thể viết là <!doctype html> hoặc <!DOCTYPE html>, cả hai đều hợp lệ.Đơn giản hóa cú pháp: So với các phiên bản HTML cũ (HTML 4.01 hoặc XHTML), cú pháp trong HTML5 ngắn gọn và không yêu cầu liên kết đến DTD (Document Type Definition).Mục đích chính:

+ Thông báo cho trình duyệt biết tài liệu này được viết bằng HTML5.

+ Giúp trình duyệt hiển thị nội dung trong chế độ tiêu chuẩn (Standards Mode), thay vì chế độ "quirks mode" (chế độ lỗi thời có thể hiển thị không chính xác).

* Thẻ ngữ nghĩa

Một phần tử ngữ nghĩa mô tả rõ ràng ý nghĩa của nó cho cả trình duyệt và lập trình viên. Ví dụ về phần tử không mang ngữ nghĩa: <div> và <span> - Không nói gì về nội dung của nó. Ví dụ về phần tử ngữ nghĩa: <form>, <table>, và <article> - Xác định nội dung rõ ràng nội dung của nó. Các phần tử ngữ nghĩa trong HTML5 được hỗ trợ bởi tất cả các trình duyệt hiện đại. Thêm vào đó, bạn có thể “dạy” các trình duyệt cũ cách để xử lý các “phần tử lạ”.

* Webform

Webform là công cụ trực tuyến cho phép người dùng tạo các biểu mẫu trên web một cách dễ dàng và nhanh chóng. Công cụ này giúp người dùng thu thập thông tin từ khách hàng hoặc người dùng mà họ muốn tương tác trên trang web của mình. Webform cung cấp các tính năng linh hoạt và đa dạng để tạo ra các biểu mẫu theo nhu cầu cụ thể của người dùng.

**KẾT LUẬN CHƯƠNG 2**

HTML (HyperText Markup Language) là nền tảng của tất cả các trang web, cung cấp cấu trúc và nội dung cơ bản. Qua nhiều năm phát triển, HTML đã tiến hóa thành HTML5, mang lại những cải tiến đáng kể về tính năng, hiệu suất và khả năng hỗ trợ cho các công nghệ web hiện đại.

**CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

**Áp dụng cách dùng các thẻ bằng emmet trong CSS**

**3.1. CSS là gì?**

**CSS** là viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

**3.2. Cú pháp CSS**

CSS được tạo thành từ các Style Rule. Các rule này được thông dịch bởi trình duyệt và sau đó được áp dụng tới các phần tử tương ứng trong tài liệu của bạn. Style Rule bao gồm ba phần:

+ Selector: (hiểu nôm na là bộ chọn phần tử)Một selector là một thẻ HTML mà tại đó một bạn áp dụng một style cho nó. Đó có thể là bất kỳ một thẻ HTML nào, như thẻ div, thẻ span, thẻ p, …

+ Property: (thuộc tính)Là property của một kiểu thuộc tính của thẻ HTML. Nói một cách đơn giản thì tất cả các thuộc tính trong HTML được chuyển đổi thành các CSS property. Đó có thể là *color*, *border*....

+ Value: Là các giá trị được gán cho các property. Ví dụ, *color* có thể có các giá trị là *red* *#F1F1F1* etc.

**3.3. Áp dụng CSS vào trang HTML**

Để áp dụng CSS vào trang HTML, có ba cách chính mà bạn có thể sử dụng: **inline CSS**, **internal CSS** và **external CSS**.

**3.4. Selectors**

CSS Selectors hay còn gọi là bộ chọn CSS là phần đầu tiên của Quy tắc CSS. Đó là một mẫu gồm các phần tử và các thuật ngữ khác cho trình duyệt, việc nên chọn phần tử HTML nào để áp dụng các giá trị thuộc tính CSS bên trong quy tắc cho chúng. Phần tử hoặc các phần tử được chọn bởi bộ chọn được gọi là chủ đề của bộ chọn. Các phần tử được bộ chọn CSS nhắm mục tiêu được gọi là "chủ đề của bộ chọn". Một chủ đề có thể được chọn dựa trên loại thành phần, lớp, tên ID, thuộc tính đã cho hoặc pseudo-state. Với rất nhiều loại CSS khác nhau có sẵn, chúng ta không chỉ có thể tùy chỉnh trang web của mình nhanh hơn mà còn có thể duy trì chi tiết kiểm soát code của bạn

3.4.1. Universal selector

Universal selector có thể được hiển thị bằng dấu hoa thị (\*), đây là một CSS bộ chọn đặc biệt cho tất cả các phần tử trên trang web. Nó áp dụng một kiểu cho mọi phần tử trong phạm vi được xác định rõ ràng, khiến nó trở thành một bộ mạnh nhưng có phạm vi rộng.

3.4.2. Type selector

Cú pháp cơ bản của bộ chọn kiểu rất dễ hiểu. Sử dụng tên thẻ làm bộ chọn, sau đó là các kiểu bạn muốn sử dụng để nhắm mục tiêu đến các thành phần của một thẻ nhất định.

3.4.3. Identity selector

Bộ chọn id sử dụng thuộc tính id của một phần tử HTML để chọn một phần tử cụ thể . Id của một phần tử là duy nhất trong một trang, do đó bộ chọn id được sử dụng để chọn một phần tử duy nhất! Để chọn một phần tử có id cụ thể, hãy viết ký tự thăng (#), theo sau là id của phần tử.

3.4.4. Class selector

Trong CSS, bộ chọn lớp được định dạng là ký tự dấu chấm (.) theo sau là tên của lớp . Nó chọn tất cả các phần tử có thuộc tính lớp đó để có thể áp dụng các khai báo CSS duy nhất cho các phần tử cụ thể đó mà không ảnh hưởng đến các phần tử khác trên trang.

3.4.5. Descendant selector

CSS [Selector](https://www.geeksforgeeks.org/css-syntax-and-selectors/) được sử dụng để chọn các phần tử HTML dựa trên tên phần tử, id, thuộc tính, v.v. của chúng, giúp dễ dàng chọn nhiều phần tử cùng lúc. [Descendant Selector](https://www.geeksforgeeks.org/css-child-vs-descendant-selectors/) là một trong những loại [Combinator](https://www.geeksforgeeks.org/css-combinators/) trong CSS, trong đó combinator kết hợp 2 selector theo cách mà nếu một phần tử tổ tiên khớp với selector đầu tiên thì các phần tử được khớp với selector thứ hai sẽ được chọn và các selector này sử dụng descendant Combinator là *Descendant Selector* . Nói một cách đơn giản, Descendant Selector có thể là bất kỳ selector nào có khoảng trắng giữa các phần tử mà không cần sử dụng bất kỳ combinator nào. Descendant là một cách để lồng nhau ở bất kỳ đâu trong cây [DOM](https://www.geeksforgeeks.org/dom-document-object-model/) . Selector này được sử dụng để chọn tất cả các phần tử con của thẻ đã chỉ định.

3.4.6. Child selector

Trong CSS, các bộ chọn con và con cháu được sử dụng để nhắm mục tiêu các phần tử dựa trên mối quan hệ phân cấp của chúng trong cấu trúc HTML . Các bộ chọn này cho phép bạn áp dụng các kiểu một cách chính xác và có kiểm soát.

3.4.7. Adjacent sibling selector

Adjacent sibling selector CSS không phải là một bộ chọn riêng lẻ, mà là một cách kết hợp hai bộ chọn . Ví dụ: p + p { margin: 0; } Dấu cộng (+) là bộ kết hợp anh chị em liền kề, giữa hai bộ chọn thẻ đoạn văn (phần tử).

3.4.8. Attribute selector

Attribute selector được sử dụng để chọn và định dạng các phần tử HTML có các thuộc tính và giá trị được chỉ định . Bộ chọn [thuộc tính] chọn tất cả các phần tử có thuộc tính được chỉ định. Bộ chọn [thuộc tính="giá trị"] chọn tất cả các phần tử có thuộc tính và giá trị được chỉ định.

3.4.9. Pseudo class selector

Pseudo class selector được sử dụng để thêm các kiểu vào bộ chọn, nhưng chỉ khi các bộ chọn đó đáp ứng các điều kiện nhất định . Một pseudo-class được thể hiện bằng cách thêm dấu hai chấm (:) sau một bộ chọn trong CSS, theo sau là một pseudo-class như "hover", "focus" hoặc "active", như thế này: a:hover { /\* your style here \*/ }

3.4.10. Group selector

Group selector được sử dụng để chọn nhiều phần tử và định dạng chúng cùng nhau . Điều này làm giảm mã và công sức khai báo các kiểu chung cho từng phần tử. Để nhóm các bộ chọn, mỗi bộ chọn được phân tách bằng một khoảng trắng.

**3.5. Đơn vị đo lường CSS**

Đơn vị đo tương đối là đơn vị đo thường được phụ thuộc vào 1 thành phần nào đó thì mới xác định được giá trị của nó. Trong các giá trị đo dưới đây thì các bạn cần chú trọng đến các đơn vị đo là em, rem, vh, vw và %, còn lại các đơn vị đo khác thì ít sử dụng hơn.

+ em: 1em = với kích cỡ của font hiện tại (font-size). Tương tự ta có 2em= 2 lần font,...

+ ex: 1ex = với chiều cao của 1 chữ x (in thường) của font hiện tại

+ ch: 1ch = chiều rộng của số 0

+ rem: giá trị tương đối với font của thành phần gốc

+ vw: 1vw = 1% của chiều rộng cửa sổ khung hình

+ vh: 1vh = 1% của chiều cao cửa sổ khung hình

+ vmin: 1vmin = 1% của khung nhìn nhỏ nhất

+ vmax: 1vmax = 1% của khung nhìn nhỏ nhất

+ %: Xác định một đơn vị đo theo phần trăm liên quan đến một giá trị khác, thường là một phần tử bao quanh.

**3.6. Kế thừa thuộc tính**

Thử nghĩ chúng ta có các lớp Class mayAsus, Class mayAcer, Class mayLenovo đều có các thuộc tính (attribute): chieudai, chieurong và phương thức (method):  upRam(). Khi đó, nếu chúng ta tạo các lớp này thì chúng ta phải viết trong mỗi lớp đều có 3 phương thức trên. Hơn thế nữa, nếu bạn muốn sửa lại code trong một phương thức nào đó thì bạn phải sửa chúng cả ở 3 lớp sẽ rất tốn thời gian, và có thể dễ sai sót. Vì thế tính kế thừa sẽ được sử dụng trong trường hợp này để giải quyết vấn đề trên.

Tính kế thừa cho phép xây dựng một lớp mới dựa trên các định nghĩa của lớp đã có. Có nghĩa là lớp cha có thể chia sẽ dữ liệu và phương thức cho các lớp con. Các lớp con khỏi phải định nghĩa lại, ngoài ra có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới. Tái sử dụng mã nguồn 1 cách tối ưu, tận dụng được mã nguồn.

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

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

Font chữ :

### + Font-family

Xác định loại font chữ cho văn bản.

Danh sách các font được cung cấp theo thứ tự ưu tiên. Nếu font đầu tiên không khả dụng, trình duyệt sẽ dùng font tiếp theo.

+ Font-size

Định kích thước font chữ.

Đơn vị thường dùng: px, %, em, rem, vw.

+ Font-weight

Quy định độ đậm của chữ.

Giá trị: normal, bold, lighter, hoặc số từ 100 đến 900.

+ Font-style

Thiết lập kiểu chữ: nghiêng hoặc không nghiêng.

Giá trị: normal, italic, oblique.

### + Font-variant

Biến thể chữ viết, ví dụ như viết chữ nhỏ hơn (small-caps).

Giá trị: normal, small-caps.

### + Line-height

Quy định khoảng cách giữa các dòng trong một đoạn văn.

Đơn vị: px, %, hoặc số thập phân (tỷ lệ so với kích thước font).

### + Font ****(**rút gọn)**

Rút gọn tất cả các thuộc tính font trong một dòng: font-style, font-variant, font-weight, font-size, line-height, font-family.

Văn bản :

+ Text-align

Căn chỉnh văn bản.

Giá trị: left, right, center, justify.

+ Text-transform

Thay đổi cách viết chữ: in hoa, thường.

Giá trị:

Uppercase: Tất cả in hoa.

Lowercase: Tất cả in thường.

Capitalize: Chữ cái đầu viết hoa.

### + Text-decoration

Thêm hiệu ứng trang trí văn bản.

Giá trị:

None: Không trang trí.underline: Gạch chân.

Overline: Gạch phía trên.line-through:Gạch ngang.

Blink: Nhấp nháy.

### + Letter-spacing

Quy định khoảng cách giữa các ký tự.

### + Word-spacing

Quy định khoảng cách giữa các từ.

### + Text-indent

Thụt lề dòng đầu tiên của đoạn văn.

Đơn vị: px, %.

### + White-space

Quy định cách xử lý khoảng trắng.

Giá trị:

Normal: Xóa khoảng trắng thừa.

Nowrap: Không xuống dòng.

Pre: Giữ nguyên khoảng trắng và xuống dòng.

Pre-line: Xóa khoảng trắng thừa, giữ xuống dòng.

### + Text-shadow

Thêm bóng cho văn bản.

Cú pháp: text-shadow, offset-x offset-y blur-radius color.

- Thuộc tính định dạng nền (background)

**+** Background-color (thiết lập màu nền)

**+** Background-image (thiết lập hình nền)

**+** Background-repeat (thiết lập sự lặp lại của hình nền)

**+** Background-attachment (thiết lập vị trí cố định cho hình nền)

**+** Background-position (thiết lập vị trí của hình nền)

- Mô hình hộp (box model)

**+** Content - nội dung trong hộp, nơi hiển thị văn bản và hình ảnh

**+** Padding - phần khoảng trắng xung quanh nội dung, trong suốt

**+** Border - đường viền xung quanh nội dung và padding

**+** Margin - phần lề bên ngoài đường viền, trong suốt

- Các thuộc tính định margin, padding, border

**+** Margin : quy định khoảng cách bên ngoài của một phần tử đến các phần tử hoặc cạnh bao quanh.

Margin-top: Khoảng cách trên.

Margin-right: Khoảng cách bên phải.

Margin-left: Khoảng cách bên trái.

Margin-bottom: Khoảng cách dưới.

**+** Padding xác định khoảng cách bên trong, giữa nội dung của phần tử và viền của nó.

Padding-top: Khoảng cách bên trên nội dung.

Padding-right: Khoảng cách bên phải nội dung.

Padding-bottom: Khoảng cách bên dưới nội dung.

Padding-left: Khoảng cách bên trái nội dung.

**+** Border định nghĩa viền bao quanh phần tử.

Border-top: Viền trên.

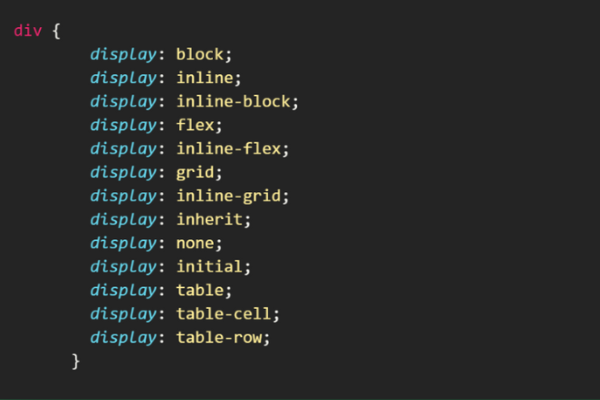
Border-right: Viền bên phải.

Border-bottom: Viền dưới.

Border-left: Viền bên trái.

- Thuộc tính định dạng cách hiển thị (display)

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



*Hình 3.1 : Các Dạng display*

- Thuộc tính xác định vị trí (position)

Static: Đây được xem là giá trị hiển thị Position trong css một cách mặc định (default), các thành phần sẽ nằm theo thứ tự của văn bản**.**

Relative:Định vị trí tuyệt đối cho các thành phần, không gây ảnh hưởng tới vị trí ban đầu hay các thành phần khác.

Absolute:Giá trị này sẽ giúp định vị trí tuyệt đối cho thành phần theo thành phần bao ngoài, hoặc ít nhất là theo cửa sổ trình duyệt. Fixed:Định vị trí tương đối cho thành phần theo cửa sổ trình duyệt.

Fixed:Định vị và giúp cho phần tử luôn cố định một chỗ, ví dụ như khi bạn scroll trình duyệt chẳng hạn, phần tử sẽ không thay đổi.

Inherit:Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài).

**3.8. Float & Clear**

**-** Float :

+ left: Đẩy phần tử sang bên trái, các phần tử khác bao quanh bên phải.

+ right: Đẩy phần tử sang bên phải, các phần tử khác bao quanh bên trái.

+ none (mặc định): Không áp dụng float, phần tử hiển thị như bình thường.

+ inherit: Kế thừa giá trị float từ phần tử cha.

**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

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

**4.1. Ý tưởng của Website**

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

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

**Thiết kế phần Header:**

+ HTML:

+ CSS:

+ Kết quả:

**Thiết kế phần Benner:**

+ HTML:

+ CSS:

+ Kết quả:

**Thiết kế phần Menu:**

+ HTML:

+ CSS:

+ Kết quả:

**Thiết kế phần Main:**

+ HTML:

+ CSS:

+ Kết quả:

**Thiết kế phần Footer:**

+ HTML:

+ CSS:

+ Kết quả:

**4.5. Kết quả đạt được**

- Chụp màn hình một số giao diện chính hoặc chức năng của trang Web **(Có ghi chú hình)**

- Hướng dần người dùng sử dụng giao diện hoặc chức năng trong trang Web

Kết luận chương 4

**KẾT LUẬN**

***(Trình bày thành 3 đoạn văn nêu Ưu điểm, nhược điểm và hướng phát triển chủ đề)***