

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Yêu cầu có các tài liệu tham khảo cho sinh viên của khoa Công nghệ Thông tin - Trường Cao đẳng Nghề ngày càng trở nên cấp thiết. Việc biên soạn tài liệu này nằm trong kế hoạch xây dựng hệ thống giáo trình các môn học của Khoa.

Đề cương của giáo trình đã được thông qua Hội đồng Khoa học của Khoa và Trường. Mục tiêu của giáo trình nhằm cung cấp cho sinh viên một tài liệu tham khảo chính về môn học Nhập môn lập trình Website, trong đó giới thiệu những khái niệm căn bản nhất về ngôn ngữ lập trình website tĩnh, đồng thời trang bị những kiến thức và một số kỹ năng chủ yếu cho việc thiết kế và xây dựng các giao diện website. Đây có thể coi là những kiến thức ban đầu và nền tảng cho các lập trình viên về thiết kế và quản trị website.

Mặc dù đã có những cố gắng để hoàn thành giáo trình theo kế hoạch, nhưng do hạn chế về thời gian và kinh nghiệm soạn thảo giáo trình, nên tài liệu chắc chắn còn những khiếm khuyết. Rất mong nhận được sự đóng góp ý kiến của các thầy cô trong Khoa cũng như các bạn sinh viên và những ai sử dụng tài liệu này. Các góp ý xin gửi về Khoa Công nghệ thông tin - Trường Cao đẳng nghề Cần Thơ. Xin chân thành cảm ơn.

Cần Thơ, ngày tháng năm 2021

Tham gia biên soạn

1. Chủ biên Nguyễn Phát Minh

MỤC LỤC

TRANG

LỜI GIỚI THIỆU	2
MỤC LỤC	3
GIÁO TRÌNH MÔN HỌC/MÔ ĐUN	4
BÀI 1: TỔNG QUAN	6
Mã bài: MĐ 11 - 01	6
1. Lịch sử World Wide Web (www)	6
2. Nhập môn ngôn ngữ HTML (Hyper Text Markup Language)	8
3. CSS (Cascading Style Sheets)	10
4. Môi trường soạn thảo:	10
BÀI 2: CÁC THẺ HTML THÔNG DỤNG	12
Mã bài: MĐ 11 - 02	12
1. Thẻ trình bày văn bản	12
2. Thẻ khối và thẻ trong dòng văn bản (nội tuyến).....	13
3. Thẻ hình ảnh và thẻ link	14
4. Thẻ Table - tạo bảng.....	16
5. Thẻ danh sách	22
6. Thẻ HTML 5.....	25
7. HTML Form	28
BÀI 3: CSS (Cascading Style Sheets)	39
Mã bài: MĐ 11 - 03	39
1. Cách sử dụng:	39
2. Các thuộc tính thường dùng:	41
Ví dụ: <code>body { background-image: url("paper.gif"); }</code>	42
Trong đó url chính là đường dẫn đến hình như khi dùng thẻ <code>img</code>	42
Ví dụ:	42
BÀI 4: Javascript	51
Mã bài: MĐ 11 - 04	51
1. Cách sử dụng:	51
2. Kiến thức cơ bản về ngôn ngữ javascript:	52
3. Kết hợp javascript với HTML:	52
TÀI LIỆU THAM KHẢO	55

GIÁO TRÌNH MÔN HỌC/MÔ ĐUN

Tên môn học/mô đun: NHẬP MÔN LẬP TRÌNH WEBSITE

Mã môn học/mô đun: MĐ 11

Vị trí, tính chất, ý nghĩa và vai trò của môn học/mô đun:

- Vị trí: Môn học được bố trí sau khi sinh viên học xong các môn học chung, các môn học cơ sở chuyên ngành đào tạo chuyên môn nghề.
- Tính chất: Là môn học cơ sở chuyên ngành bắt buộc.
- Ý nghĩa và vai trò của môn học/mô đun: Nhập môn lập trình website là môn học cơ bản để sinh viên tìm hiểu về các khái niệm về lập trình web để làm nền tảng học các môn chuyên sau này như Thiết kế Website

Mục tiêu của môn học/mô đun:

- Về kiến thức:
 - Sinh viên hiểu rõ mô hình, cấu trúc và nguyên lý hoạt động của các website;
 - Hiểu rõ cấu trúc một trang HTML và tính năng, cú pháp của các thẻ HTML;
- Về kỹ năng:
 - Xây dựng được các website có thẩm mỹ
 - Chuyển đổi được giao diện website trên tập tin psd thành trang web hoạt động được
- Về năng lực tự chủ và trách nhiệm:
 - Nghiêm túc, tỉ mỉ trong việc tiếp nhận kiến thức. Chủ động, tích cực trong thực hành và tìm kiếm nguồn bài tập liên quan

Nội dung của môn học/mô đun:

Số TT	Tên chương, mục	Thời gian			
		Tổng số	Lý thuyết	Thực hành Bài tập	Kiểm tra* (LT hoặc TH)
I	Tổng quan Lịch sử World Wide Web (www) Nhập môn ngôn ngữ HTML (Hyper Text Markup Language) CSS (Cascading Style Sheets) Môi trường soạn thảo	2	1	1	
II	Các thẻ HTML thông dụng Thẻ trình bày văn bản	24	13	10	1

	Thẻ khối và thẻ trong dòng văn bản				
	Thẻ hình ảnh và thẻ link				
	Thẻ Table - tạo bảng				
	Thẻ danh sách				
	Thẻ HTML 5				
	HTML Form				
III	CSS (Cascading Style Sheets)	24	14	9	1
	Cách sử dụng				
	Các thuộc tính thường dùng				
IV	Javascript	10	2	7	1
	Cách sử dụng				
	Kiến thức cơ bản về ngôn ngữ javascript				
	Kết hợp javascript với HTML				
	Cộng	60	30	27	3

BÀI 1: TỔNG QUAN

Mã bài: MĐ 11 - 01

Giới thiệu:

Bài học này nhằm giới thiệu sơ lược về lịch sử của World Wide Web (www), URL, về giao thức HTTP và ngôn ngữ phổ biến được dùng bởi World Wide Web là HTML (Hyper Text Markup Language) và CSS (Cascading Style Sheets)

Mục tiêu của bài:

- Trình bày được lịch sử của WWW.
- Trình bày được cấu trúc của một trang HTML.
- Trình bày được khái niệm CSS.
- Sử dụng được một trong các trình soạn thảo được giới thiệu.

Nội dung chính:

1. Lịch sử World Wide Web (www)

Mục tiêu: Trình bày được lịch sử của WWW.

1.1. Giới thiệu về World Wide Web (www)

Ngày nay người ta dùng máy tính như một công cụ rất hữu ích để truy cập Internet, chủ yếu là tìm kiếm thông tin. Thông tin này có thể là văn bản, hình ảnh, âm thanh hay thông tin đa phương tiện...

Nhu cầu tìm kiếm thông tin ngày càng nhiều đã đưa ra vấn đề: làm thế nào để dùng sử dụng máy tính truy cập Internet như một công cụ phục vụ đặc lực cho việc tra cứu, tìm kiếm thông tin trên mạng thông tin rộng lớn nhất toàn cục.

Vấn đề trên trở nên dễ dàng hơn bởi ý tưởng siêu văn bản (Hypertext) – văn bản thông minh nhà tin học Ted Nelson đề xuất vào năm 1965. Đến 1989, dự án chính thức được thực hiện bởi một kỹ sư trẻ người Anh tên là Tim Berners – Lee.

World Wide Web (www) là một mạng các tài nguyên thông tin. WWW dựa trên 3 cơ chế để các tài nguyên trở nên sẵn dùng cho người xem càng rộng rãi nhất càng tốt, đó là:

- Cơ chế đặt tên cùng dạng đối với việc định dạng các tài nguyên trên WWW (như các URL).
- Các giao thức, để truy cập tới các tài nguyên qua WWW (như HTTP).
- Siêu văn bản, để dễ dàng chuyển đổi giữa các tài nguyên (như HTML).

1.2. Giới thiệu về URL:

Mọi tài nguyên sẵn dùng trên WWW – tài liệu HTML, ảnh, video clip, chương trình v..v.. - có một địa chỉ mà có thể được mã hóa bởi một URL.

URL được xem là một con trỏ dùng với mục đích đơn giản là xác định vị trí tài nguyên của môi trường Internet. Thông qua các URL mà Web Browser có thể tham chiếu đến một Web Server hoặc các dịch vụ khác trên Internet và ngược lại.

Các URL thường gồm 3 phần:

- Việc đặt tên của các cơ chế dùng để truy cập tài nguyên.
- Tên của máy tính lưu trữ (tổ chức) tài nguyên.
- Tên của bản thân tài nguyên, như một đường dẫn.

Ví dụ: URL xác định trang W3C Technical Reports là **http://www.w3.org/TR**

URL này có thể được đọc như sau: Có một tài liệu sẵn dùng theo giao thức HTTP, đang lưu trong máy **www.w3.org**, có thể truy cập theo đường dẫn **"/TR"**. Các cơ chế khác ta có thể thấy trong các tài liệu HTML bao gồm **"mailto"** đối với thư điện tử và **"ftp"** đối với FTP.

Ví dụ sau đây chỉ tới hộp thư (mailbox) của người dùng:

Mọi góp ý, xin gửi thư tới

Joe Cool

Các định danh đoạn (fragment identifiers): Một số URL chỉ tới việc định vị một tài nguyên. Kiểu này của URL kết thúc với **"#"** theo sau bởi một dấu hiệu kết nối (gọi là các định danh đoạn). Ví dụ, đây là một URL đánh dấu một móc tên là **section_2**: http://somesite.com/html/top.html#section_2

Các URL tương đối: không theo cơ chế đặt tên. Đường dẫn của nó thường tham chiếu tới một tài nguyên trên cùng một máy chứa tài liệu hiện tại. Các URL tương đối có thể gồm các thành phần đường dẫn tương đối (như **".."** nghĩa là một mức trên trong cấu trúc được định nghĩa bởi đường dẫn), và có thể bao gồm các dấu hiệu đoạn.

Ví dụ của giải pháp URL tương đối, giả sử chúng ta có URL gốc:

"http://www.acme.com/support/intro.html".

URL tương đối trong đánh dấu dưới đây cho một liên kết siêu văn bản:

Suppliers

sẽ mở rộng thành URL đầy đủ

"http://www.acme.com/support/suppliers.html"

trong khi URL tương đối trong việc đánh dấu cho một ảnh dưới đây

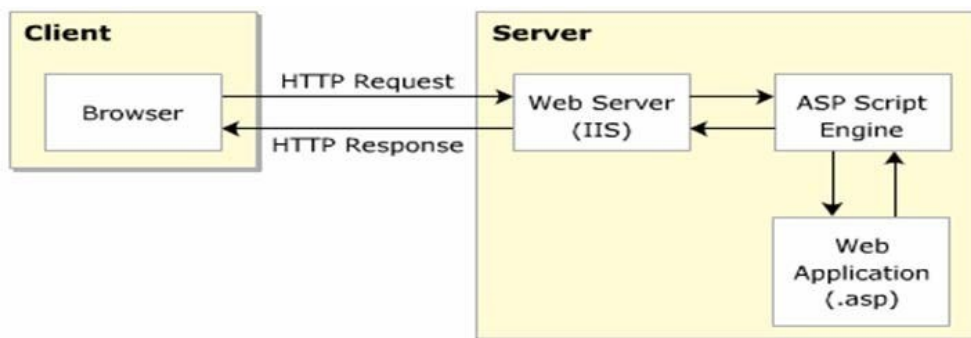
sẽ mở rộng thành URL đầy đủ **"http://www.acme.com/icons/logo.gif"**

Các URL được dùng để:

- Liên kết tới tài liệu hoặc tài nguyên khác.
- Liên kết tới kiểu dạng bên ngoài hoặc kịch bản (script).

1.3. Giới thiệu về HTTP

Web Browser và Web Server giao tiếp với nhau thông qua một giao thức được gọi là HTTP. Sự kết nối HTTP qua 4 giai đoạn:



Hình 1.1: Sự kết nối HTTP

- Tạo kết nối: Web Browser giao tiếp với Web Server nhờ địa chỉ URL.
- Internet và số cổng (ngầm định là 80) được đặc tả trong URL.
- Thực hiện yêu cầu: Web Browser gửi thông tin tới Web Server để yêu cầu phục vụ. Việc gửi thông tin ở đây là gửi phương thức GET dùng cho việc lấy một đối tượng từ Server, hay POST dùng cho việc gửi dữ liệu tới một đối tượng trên Server.
- Phản hồi: Web Server gửi một phản hồi về Web Browser nhằm đáp ứng yêu cầu của Web Browser.
- Kết thúc kết nối: Khi kết thúc quá trình trao đổi giữa Web Browser và Web Server thì sự kết nối chấm dứt. Và như vậy mối liên hệ giữa Client và Server chỉ được tồn tại trong quá trình trao đổi với nhau, điều này có lợi điểm rất lớn là giảm được lưu thông trên mạng.

2. Nhập môn ngôn ngữ HTML (Hyper Text Markup Language)

Mục tiêu: Trình bày được cấu trúc của một trang HTML và các thẻ HTML cơ bản.

Ngôn ngữ phổ biến dùng bởi World Wide Web là HTML (Hyper Text Markup Language). Nó được dùng cho các mục đích sau:

- Phổ biến các tài liệu trực tuyến với các heading, văn bản, bảng, danh sách, ảnh,..v.v...
- Truy tìm thông tin trực tuyến theo các liên kết siêu văn bản bằng việc kích vào một nút.
- Thiết kế các định dạng cho việc kiểm soát các giao dịch (transaction) với các thiết bị từ xa, đối với người dùng trong việc tìm kiếm thông tin, tạo các sản phẩm, đặt hàng,..v.v...
- Bao gồm spread-sheets, video clips, sound clips, và các ứng dụng trực tiếp khác trong các tài liệu của họ.

HTML đánh dấu văn bản dưới dạng các thẻ (Tag). Cấu trúc thẻ HTML có dạng như sau:

- Thẻ đóng: **<Tag>** văn bản chịu tác động **</Tag>**

Trong đó:

- + **<Tag>**: bắt đầu hiệu ứng thẻ.

+ </Tag>: kết thúc hiệu ứng thẻ.

Ví dụ: văn bản này được in đậm

sẽ cho kết quả ở trình duyệt là:

văn bản này được in đậm

- Thẻ mở: <Tag> văn bản chịu tác động

Ví dụ: Đoạn 1<p> Đoạn 2

sẽ cho kết quả là:

Đoạn 1

Đoạn 2

- Về quy tắc các thẻ có thể lồng lẫn nhau nhưng vẫn phải đảm bảo đúng cú pháp của thẻ đó.

Khung sườn chính của trang web:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Page Title</title>
```

```
</head>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

Trong đó:

<!DOCTYPE html>: là để khai báo trang web có sử dụng thẻ HTML 5

<head></head>: để khai báo các thành phần dùng cho trang web

Chú ý:

Muốn hiển thị tiếng Việt trên trang web chúng ta cần khai báo 1 thẻ meta bên trong thẻ head. Cú pháp như sau:

```
<meta charset="UTF-8">
```

<body> toàn bộ nội dung của trang web sẽ khai báo ở đây

Để viết ghi chú trong HTML chúng ta dùng như sau: <!-- Ghi chú -->

3. CSS (Cascading Style Sheets)

CSS là các Style dùng định nghĩa cách trình duyệt hiển thị các đối tượng HTML. Các Style này được lưu trong Style. Nhiều định nghĩa Style cho cùng một loại đối tượng sẽ được sử dụng theo lớp.

Cú pháp CSS

Cú pháp của CSS gồm 3 phần: đối tượng, thuộc tính và giá trị:

Đối tượng {thuộc tính: giá trị}

Trong đó:

+ Đối tượng thường là các tag HTML cần định nghĩa cách hiển thị, có thể là 1 khai báo ID hoặc CLASS

+ Thuộc tính là thuộc tính hiển thị của đối tượng đó.

+ Giá trị là cách mà bạn muốn một thuộc tính hiển thị như thế nào.

+ Các cặp thuộc tính: giá trị sẽ được phân cách nhau bởi dấu “;”

Ví dụ:

Để định nghĩa Style cho thẻ p

```
p { text-align: center; color: black; font-family: arial }
```

4. Môi trường soạn thảo:

Để có thể soạn thảo một trang web, chúng ta có thể dùng nhiều phần mềm để biên soạn, trong giáo trình này giới thiệu đến các học sinh phần mềm có sẵn trên hệ điều hành của window là NOTEPAD, WORDPAD, chỉ cần sau khi soạn xong lưu lại với phần mở rộng là HTML. Ngoài ra có rất nhiều phần mềm hỗ trợ chuyên dụng như ATOM, PHPDESIGN, ...

Bài tập:

Sử dụng phần mềm NOTEPAD soạn thảo một trang web gồm có các thông tin sau:

- Tiêu đề trang là: Hello World
- Có 1 đoạn văn bản hiển thị thông tin là tên của mình.

Bài tập nâng cao:

Sử dụng phần mềm ATOM, tạo 1 thư mục có tên của mình, sau đó tiến hành soạn thảo một trang web trong ATOM có các thông tin sau:

- Tiêu đề là: Hello ATOM
- Có 2 đoạn văn bản hiển thị thông tin là tên của mình, và email.

Những trọng tâm cần chú ý trong bài:

- Trình bày được cấu trúc của một trang HTML.
- Sử dụng được một trong các trình soạn thảo được giới thiệu

Yêu cầu về đánh giá kết quả học tập:**Nội dung:**

- + Về kiến thức: Trình bày được cấu trúc của một trang HTML
- + Về kỹ năng: Sử dụng được một trong các trình soạn thảo được giới thiệu.
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

- + Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp
- + Về kỹ năng: Đánh giá kỹ sử dụng phần mềm thiết kế và giao diện trang web.
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 2: CÁC THẺ HTML THÔNG DỤNG

Mã bài: MĐ 11 - 02

Giới thiệu:

Các thẻ dùng thiết kế Website có rất nhiều, sau đây sẽ trình bày một số thẻ thường dùng nhất để tạo thành một trang Web hoàn chỉnh.

Mục tiêu của bài:

- Sử dụng được các thẻ HTML thông dụng.
- Sử dụng được các thẻ HTML 5.
- Xây dựng được giao diện trang web hoàn chỉnh dùng các thẻ HTML 5.

Nội dung chính:

1. Thẻ trình bày văn bản

1.1. Thẻ tiêu đề:

Mở đầu các trang văn bản thường là các tiêu đề cần làm nổi bật từng phần của văn bản như Chương, Mục,... cũng cần có đề mục rõ ràng khác với phần thân để người đọc theo dõi cho thuận tiện.

Có 6 mức tiêu đề trong HTML: h1, h2, h3, h4, h5, h6. Sử dụng như các thẻ html thông thường gồm có thẻ mở và thẻ đóng.

Ví dụ:

`<H1>Tiêu đề 1</H1>` cho kết quả **Tiêu đề 1**

`<H6>Tiêu đề 6</H6>` cho kết quả **Tiêu đề 2**

1.2. Thẻ đoạn văn bản:

Thẻ này dùng để xác định một đoạn văn bản. Thẻ `<p>` (Paragraph) có thể dùng kèm thuộc tính để ấn định cách trình bày đoạn văn bản.

Ví dụ:

`<p>Đoạn văn bản 1</p>`

`<p>Đoạn văn bản 2</p>`

Để căn lề văn bản, chúng ta có thể dùng thuộc tính chỉnh lề nằm trong thẻ p như sau: `<p align= "left" | "right" | "center">Đoạn văn bản</p>`

1.3. Thẻ xuống dòng:

Khi dùng thẻ p, các đoạn văn bản sẽ tự xuống dòng để phân thành 1 đoạn. Nếu chúng ta cần xuống dòng tại 1 vị trí bất kỳ, thì có thể dùng thẻ `
`, khi dùng thẻ `
` thì toàn bộ văn bản phía sau sẽ nhảy xuống 1 dòng, dùng nhiều `
` sẽ nhảy xuống nhiều dòng.

Ví dụ:

<p>Đoạn
văn

bản 1<p> → có kết quả là:

Đoạn

văn

bản 1

1.4. Đường kẻ ngang:

Thẻ này sẽ cho chúng ta 1 đường kẻ ngang màn hình, đường kẻ này có thể chỉnh chiều dài bằng cách thay đổi thuộc tính chiều rộng của thẻ

Ví dụ:

<hr> → 1 đường kẻ ngang màn hình

<hr width="50%"> → 1 đường kẻ ngang màn hình có độ rộng là 50% màn hình

<hr width="50px"> → 1 đường kẻ ngang màn hình có độ rộng là 50px

1.5. Thẻ định dạng văn bản thông dụng:

Để định dạng văn bản như in đậm, in nghiêng, gạch dưới chúng ta có thể sử dụng các thẻ lần lượt là , <i>, <u>

Ví dụ:

<i><u><p>Đoạn văn bản</p></u></i> → ***Đoạn văn bản***

<p>Đoạn <i>văn</i> <u>bản</u></p> → **Đoạn** *văn* bản

1.6. Thẻ giữ nguyên định dạng văn bản:

Thông thường khi dùng thẻ p để tạo đoạn văn bản, nếu chúng ta xuống nhiều dòng trong đoạn đó sẽ không được tính, và khi hiển thị sẽ chỉ hiển thị 1 đoạn. Để hiển thị đúng với những gì mình thực hiện khi soạn thảo, chúng ta có thể dùng thẻ <pre> thay cho p. Khi dùng thẻ pre, văn bản sẽ chuyển thành font chữ Courier.

2. Thẻ khối và thẻ trong dòng văn bản (nội tuyến)

Mỗi thẻ HTML đều có giá trị hiển thị mặc định, tùy thuộc vào loại thẻ đó. Có hai giá trị hiển thị: khối và nội tuyến.

2.1. Thẻ cấp khối:

Một thẻ cấp khối luôn bắt đầu trên một dòng mới.

Một thẻ cấp khối luôn chiếm toàn bộ chiều rộng có sẵn (trái dài sang trái và phải hết mức có thể). Thẻ cấp khối có lề trên và lề dưới, trong khi thẻ nội tuyến thì không. Thẻ <div>, <p> là thẻ cấp khối.

Một số thẻ cấp khối:

<address><article><aside><blockquote><canvas><dd>

<div><dl><dt><fieldset><figcaption><figure><footer><form>

<h1>-<h6><header><hr><main><nav>

<noscript><p><pre><section><table><tfoot><video>

2.2. Thẻ nội tuyến:

Một thẻ nội tuyến không bắt đầu trên một dòng mới. Một thẻ nội tuyến chỉ chiếm nhiều chiều rộng khi cần thiết. Thẻ ``, ``, `<i>`, `<u>` là thẻ nội tuyến.

Một số thẻ nội tuyến:

```
<a><abbr><acronym><b><bdo><big><br><button>
<cite><code><dfn><em><i><img><input><kbd><label>
<map><object><output><q><samp><script><select>
<small><span><strong><sub><sup><textarea><time><tt><var>
```

2.3. Thẻ DIV

Thẻ `<div>` thường được sử dụng làm vùng chứa cho các phần tử HTML khác. Thẻ `<div>` không có thuộc tính bắt buộc, nhưng `style`, `class` và `id` thường được dùng.

Khi được sử dụng cùng với CSS, `<div>` phần tử có thể được sử dụng để tạo kiểu cho các khối nội dung, xây dựng bố cục trang web.

Ví dụ:

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>DIV</h2>
```

```
  <p>
```

Thẻ `div` thường được sử dụng làm vùng chứa cho các phần tử HTML khác.

```
  </p>
```

```
</div>
```

2.4. Thẻ SPAN

Thẻ `` là một vùng chứa nội tuyến được sử dụng để đánh dấu một phần của văn bản hoặc một phần của tài liệu. Thẻ `` không có thuộc tính bắt buộc, nhưng `style`, `class` và `id` thường được dùng.

Khi được sử dụng cùng với CSS, `` phần tử có thể được sử dụng để tạo kiểu cho các phần của văn bản.

Ví dụ:

```
<p>Thẻ <span style="color:blue;font-weight:bold">span</span> là một vùng chứa
<span style="color:darkolivegreen;font-weight:bold">nội tuyến</span> được sử dụng
để đánh dấu một phần của văn bản hoặc một phần của tài liệu</p>
```

3. Thẻ hình ảnh và thẻ link

3.1. Thẻ Link:

Các liên kết được tìm thấy trong hầu hết các trang web. Liên kết cho phép người dùng nhấp theo cách của họ từ trang này sang trang khác.

Các liên kết HTML là các siêu liên kết. Có thể nhấp vào một liên kết và chuyển đến một tài liệu khác. Khi di chuyển chuột qua một liên kết, mũi tên chuột sẽ biến thành một bàn tay nhỏ.

Lưu ý: Một liên kết không nhất thiết phải là văn bản. Một liên kết có thể là một hình ảnh hoặc bất kỳ phần tử HTML nào khác!

<a>Thẻ HTML xác định một siêu liên kết.

Cú pháp:

Liên kết

Thuộc tính quan trọng nhất của <a> là href, chỉ ra điểm đến của liên kết. Các văn bản liên kết là phần mà sẽ được hiển thị cho người đọc. Nhấp vào văn bản liên kết, sẽ đưa người đọc đến địa chỉ URL được chỉ định.

Ví dụ:

Cùng học HTML

Theo mặc định, trang được liên kết sẽ được hiển thị trong cửa sổ trình duyệt hiện tại. Để thay đổi điều này, bạn phải chỉ định một target - mục tiêu khác cho liên kết. Các target quy định cụ thể thuộc tính nơi để mở tài liệu được liên kết.

Các target có thể có một trong các giá trị sau:

- `_self`: Mở tài liệu trong cùng một cửa sổ / tab khi nó được nhấp vào
- `_blank`: Mở tài liệu trong một cửa sổ hoặc tab mới
- `_parent`: Mở tài liệu trong khung chính
- `_top`: Mở tài liệu trong toàn bộ phần thân của cửa sổ

Ví dụ:

Cùng học HTML

3.2. Thẻ hình ảnh:

Hình ảnh có thể cải thiện thiết kế và giao diện của một trang web

 là thẻ HTML được sử dụng để nhúng một hình ảnh vào một trang web.

Hình ảnh không được chèn vào một trang web về mặt kỹ thuật; hình ảnh được liên kết với các trang web. Các thẻ trống, nó chỉ chứa các thuộc tính, và **không có một thẻ đóng**.

Các thẻ có hai thuộc tính cần thiết:

- `src` - Chỉ định đường dẫn đến hình ảnh
- `alt` - Chỉ định văn bản thay thế cho hình ảnh khi hình ảnh bị lỗi không hiển thị được

Cú pháp:

Ví dụ:

 → hinhnhen.png nằm cùng thư mục với tập tin html

 → <https://ibb.co/qF1NZqY/> là đường dẫn chứa hình ảnh trên internet

Có thể đặt kích thước hiển thị cho hình ảnh theo tỉ lệ % hay px. Đặt kích thước hình ảnh bằng cách dùng thuộc tính width hay height hoặc đặt theo css với thuộc tính style.

Ví dụ:

```

```

```

```

Có thể đặt link cho hình ảnh để tạo ra hình ảnh có dạng liên kết, bằng cách đặt thẻ img bên trong thẻ a

Ví dụ:

```
<a href="https://google.com">
```

```

```

```
</a>
```

Các định dạng hình ảnh chung:

Dưới đây là các loại tệp hình ảnh phổ biến nhất, được hỗ trợ trong tất cả các trình duyệt (Chrome, Edge, Firefox, Safari, Opera):

Loại tập tin	Mô tả	Đuôi của tập tin
APNG	Hình động dùng cho mạng	.apng
GIF	Hình ảnh có thể chuyển động	.gif
ICO	Hình dạng ICON của Microsoft	.ico, .cur
JPEG	Hình ảnh thông thường	.jpg, .jpeg, .jfif, .pjpeg, .jpp
PNG	Hình ảnh chuyên dùng trên mạng	.png
SVG	Hình ảnh dạng vector (dùng trong thiết kế)	.svg

4. Thẻ Table - tạo bảng

Bảng HTML cho phép các nhà phát triển web sắp xếp dữ liệu thành các hàng và cột.

Ví dụ:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany

Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Thẻ dùng để tạo bảng là <table>. Một bảng trong HTML bao gồm các ô bảng bên trong các hàng <tr> và cột <td>. Tiêu đề của bảng có thể có hoặc không, <th> là thẻ để tạo tiêu đề.

Ví dụ: Một bảng thông thường trong HTML

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Theo ví dụ trên ta thấy, các thẻ <td> và <th> phải nằm trong thẻ <tr>, tức là phải tạo hàng trước rồi mới tạo cột hay tiêu đề. <td> ở đây còn gọi là ô, vì chỉ những gì thể hiện trong <td> mới hiển thị lên website

4.1. Đường viền:

Mặc định bảng khi được tạo ra sẽ không có đường viền. Để hiển thị đường viền, chúng ta cần thay đổi thuộc tính **border** của table thành giá trị (thông thường là 1)

Ví dụ: `<table border="1">...`

Ngoài ra, chúng ta có thể dùng thuộc tính `style` để tùy chỉnh hoặc dùng CSS (sẽ học ở bài sau) để làm cho bảng chúng ta sinh động hơn.

4.2. Kích thước bảng:

Bảng HTML có thể có các kích thước khác nhau cho từng cột, hàng hoặc toàn bộ bảng. Sử dụng thuộc tính `style` với `width` hoặc `height` để chỉ định kích thước của bảng, hàng hoặc cột.

Để đặt chiều rộng của bảng, hãy thêm thuộc tính `style` vào `<table>`:

Ví dụ:

```
<table style="width:100%">
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Để đặt kích thước của một cột cụ thể, hãy thêm thuộc tính `style` vào một `<th>` hoặc `<td>` phía trên cùng.


```

<table style="width:100%">
  <tr>
    <th style="width:70%">Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>

```

Để đặt chiều cao của một hàng cụ thể, hãy thêm thuộc tính style vào <tr> trong bảng:


```

<table style="width:100%">
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr style="height:200px">
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>

```

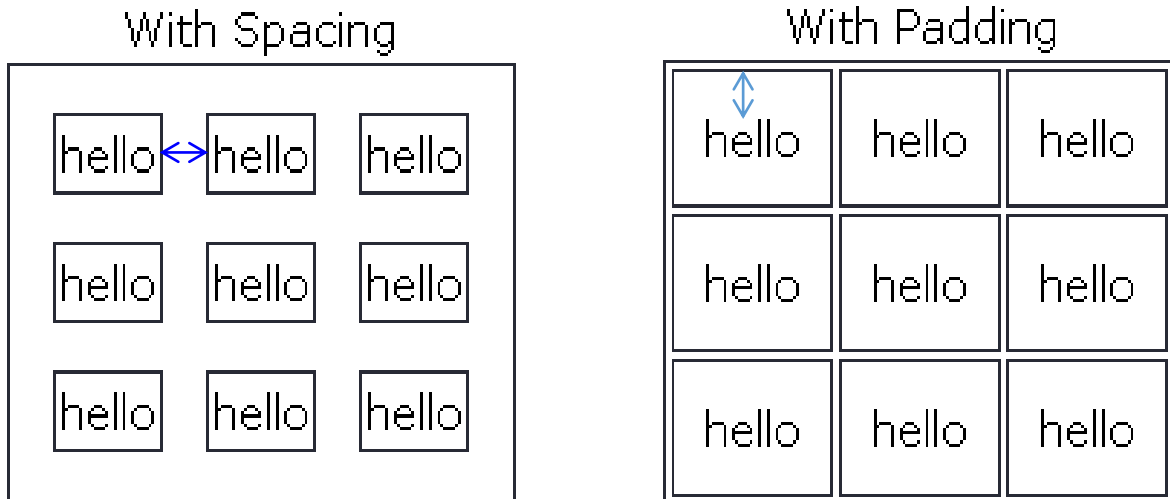
```

</tr>
<tr>
  <td>16</td>
  <td>14</td>
  <td>10</td>
</tr>
</table>

```

4.3. Cellspacing & cellpadding:

Cellspacing và cellpadding là 2 thuộc tính của table có công dụng điều chỉnh khoảng cách giữa các ô và khoảng cách từ biên của ô đến nội dung bên trong ô, ngoài 2 thuộc tính này, chúng ta có thể sử dụng css để điều chỉnh



```

<table style="width:100%" cellspacing="10px"
cellpadding="10px">
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>

```

```

    <td>14</td>
    <td>10</td>
</tr>
</table>

```

4.4. Trộn hàng và trộn cột trong bảng của HTML:

Bảng HTML có thể có các ô kéo dài (trộn lại) trên nhiều hàng hoặc cột.

TÊN		

THÁNG TƯ		

Để tạo một ô mở rộng trên nhiều cột, sử dụng thuộc tính colspan cho 1 ô. Cú pháp colspan= “số ô muốn trộn”

Ví dụ:

```

<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>

```

Name		Age
Jill	Smith	43
Eve	Jackson	57

Để tạo một ô mở rộng trên nhiều hàng, hãy sử dụng thuộc tính rowspan cho 1 ô. Cú pháp: rowspan = “số ô muốn trộn”

Ví dụ:

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
```

Name	Jill
Phone	555-1234
	555-8745

5. Thẻ danh sách

Danh sách HTML cho phép các nhà phát triển web nhóm một tập hợp các mục có liên quan trong danh sách. Có 2 loại

Danh sách HTML không có thứ tự:

- Bài báo
- Bài báo
- Bài báo
- Bài báo

Danh sách HTML có thứ tự:

1. Mục đầu tiên
2. Mục hàng thứ hai
3. Mục hàng thứ ba
4. Mục hàng thứ tư

5.1. Danh sách HTML không có thứ tự

Một danh sách không có thứ tự bắt đầu bằng thẻ. Mỗi mục danh sách bắt đầu bằng thẻ.

Các mục trong danh sách sẽ được đánh dấu bằng dấu đầu dòng (vòng tròn nhỏ màu đen) theo mặc định:

Ví dụ:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

Chọn công cụ đánh dấu mục danh sách cho danh sách không theo thứ tự bằng cách dùng thuộc tính list-style-type, có 4 loại:

Loại đánh dấu	Mô tả
----------------------	--------------

disc	Vòng tròn đen đặc (mặc định của danh sách)
------	--

circle	Vòng tròn rỗng
--------	----------------

square	Hình vuông đặc
--------	----------------

none	Không hiển thị hình đánh dấu
------	------------------------------

Ví dụ:

```
<ul style="list-style-type:circle;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

5.2. Danh sách HTML có thứ tự:

Một danh sách có thứ tự bắt đầu bằng thẻ. Mỗi mục danh sách bắt đầu bằng thẻ.

Các mục trong danh sách sẽ được đánh dấu bằng số theo mặc định:

Ví dụ:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

Các thuộc tính type của thẻ, xác định các loại điểm đánh dấu mục danh sách, có các loại sau:

Loại	Mô tả
-------------	--------------

type="1"	Theo số thứ tự 1, 2, 3 (cái này mặc định)
----------	---

type="A"	Theo chữ cái hoa A, B, C
----------	--------------------------

type="a" Theo chữ cái thường a, b, c

type="I" Theo chữ số la mã hoa I, II, III

type="i" Theo chữ số la mã thường i, ii, iii

Ví dụ:

```
<ol type="A">
  <li>Coffee</li>      A. Coffee
  <li>Tea</li>        B. Tea
  <li>Milk</li>       C. Milk
</ol>
```

5.3. Danh sách mô tả:

Danh sách mô tả là danh sách các thuật ngữ, với mô tả của từng thuật ngữ.

Các <dl>thẻ định nghĩa danh sách mô tả, thẻ định nghĩa thuật ngữ (tên), và các thẻ mô tả từng hạn: <dt> <dd>

Ví dụ:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Coffee
- black hot drink
Milk
- white cold drink

Chú ý các danh sách có thể được lồng vào nhau (danh sách bên trong danh sách):

Ví dụ:

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
 - Black tea
 - Green tea
- Milk

6. Thẻ HTML 5

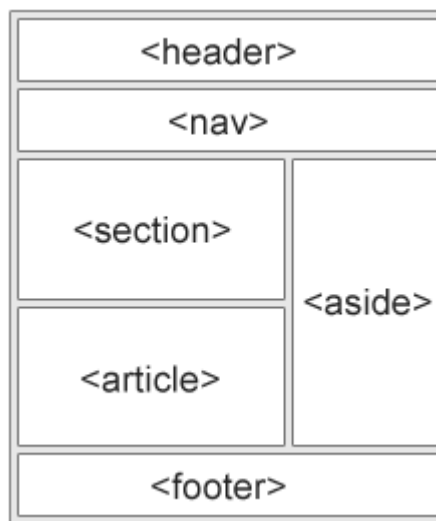
Nhãn	Sự miêu tả
<article>	Xác định một bài báo.
<aside>	Xác định một số nội dung liên quan lỏng lẻo đến nội dung trang.
<audio>	Nhúng âm thanh hoặc luồng âm thanh vào tài liệu HTML.
<bdi>	Trình bày văn bản được biệt lập với xung quanh cho mục đích định dạng văn bản hai chiều.
<canvas>	Xác định một vùng trong tài liệu, vùng này có thể được sử dụng để vẽ đồ họa một cách nhanh chóng thông qua tập lệnh (thường là JavaScript).
<data>	Liên kết một phần nội dung với bản dịch có thể đọc được bằng máy.
<datalist>	Đại diện cho một tập hợp các tùy chọn được xác định trước cho một <input>phần tử.
<details>	Đại diện cho một tiện ích mà từ đó người dùng có thể nhận được thông tin bổ sung hoặc các điều khiển theo yêu cầu.
<dialog>	Xác định hộp thoại hoặc cửa sổ phụ.
<embed>	Nhúng ứng dụng bên ngoài, thường là nội dung đa phương tiện như âm thanh hoặc video vào tài liệu HTML.
<figcaption>	Xác định chú thích hoặc chú thích cho một hình.
<figure>	Đại diện cho một hình được minh họa như một phần của tài liệu.
<footer>	Đại diện cho chân trang của một tài liệu hoặc một phần.
<header>	Đại diện cho tiêu đề của một tài liệu hoặc một phần.

Nhãn	Sự miêu tả
<hgroup>	Xác định một nhóm các tiêu đề.
<keygen>	Đại diện cho một điều khiển để tạo cặp khóa công khai-riêng tư.
<main>	Trình bày nội dung chính hoặc chủ đạo của tài liệu.
<mark>	Trình bày văn bản được đánh dấu cho mục đích tham khảo.
<menuitem>	Xác định danh sách (hoặc menuitem) các lệnh mà người dùng có thể thực hiện.
<meter>	Đại diện cho một phép đo vô hướng trong một phạm vi đã biết.
<nav>	Xác định một phần của các liên kết điều hướng.
<output>	Biểu diễn kết quả của một phép tính.
<picture>	Xác định vùng chứa cho nhiều nguồn hình ảnh.
<progress>	Đại diện cho tiến độ hoàn thành của một nhiệm vụ.
<rp>	Cung cấp dấu ngoặc đơn dự phòng cho các trình duyệt không hỗ trợ chú thích ruby.
<rt>	Xác định cách phát âm của ký tự được trình bày trong chú thích ruby.
<ruby>	Đại diện cho một chú thích bằng ruby.
<section>	Xác định một phần của tài liệu, chẳng hạn như đầu trang, chân trang, v.v.
<source>	Xác định các tài nguyên phương tiện thay thế cho các phần tử phương tiện như <audio> hoặc <video>.
<summary>	Xác định tóm tắt cho <details> phần tử.

Nhãn	Sự miêu tả
<svg>	Nhúng nội dung SVG (Đồ họa Vector có thể mở rộng) vào tài liệu HTML.
<template>	Xác định các đoạn HTML cần được ẩn khi trang được tải, nhưng có thể được sao chép và chèn vào tài liệu bằng JavaScript.
<time>	Đại diện cho một thời gian và / hoặc ngày tháng.
<track>	Xác định các rãnh văn bản cho các phần tử phương tiện như <audio> hoặc <video>.
<video>	Nhúng nội dung video vào tài liệu HTML.
<wbr>	Đại diện cho cơ hội ngắt dòng.

Bảng trên là danh sách các thẻ HTML5 hiện hành. Tùy vào hoàn cảnh mà sử dụng, thông thường các thẻ sau thường dùng để tạo giao diện trang web:

- <header> - Phần đầu của trang web
- <nav> - Phần thanh công cụ, thanh điều khiển, menu
- <section> - Phần chứa tài liệu, nội dung trang
- <article> - Phần chứa các mục văn bản
- <aside> - Phần thanh công cụ bên trái hoặc bên phải
- <footer> - Phần cuối của trang web.



7. HTML Form

Một biểu mẫu (Form) HTML được sử dụng để thu thập thông tin đầu vào của người dùng. Đầu vào của người dùng thường được gửi đến một máy chủ để xử lý.

First name:

Last name:

`<form>` là thẻ HTML được sử dụng để tạo một biểu mẫu HTML cho đầu vào của người dùng.

7.1. Các thuộc tính của Form:

- Action: action định nghĩa các hành động được thực hiện khi biểu mẫu được gửi. Thông thường, dữ liệu biểu mẫu được gửi đến tệp trên máy chủ khi người dùng nhấp vào nút gửi (submit).

Trong ví dụ dưới đây, dữ liệu biểu mẫu được gửi đến một tệp có tên "action_page.php". Tệp này chứa tập lệnh phía máy chủ xử lý dữ liệu biểu mẫu:

Ví dụ:

```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname"  
value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname"  
value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

- Target: các target quy định cụ thể thuộc tính nơi để hiển thị câu trả lời đó là nhận được sau khi nộp mẫu đơn, giống với thẻ `<a>` có các giá trị sau: `_blank`, `_self`, `_parent`, `_top`.

Ví dụ:

```
<form action="/action_page.php" target="_blank">
```

- Method: các method quy định cụ thể phương thức HTTP được sử dụng khi nộp dữ liệu mẫu.

Dữ liệu biểu mẫu có thể được gửi dưới dạng biến URL (với method="get") hoặc dưới dạng giao dịch bài đăng HTTP (với method="post").

Phương thức HTTP mặc định khi gửi dữ liệu biểu mẫu là GET.

Ví dụ:

```
<form action="/action_page.php" method="get">
```

```
<form action="/action_page.php" method="post">
```

- Autocomplete: có 2 giá trị là on hoặc off, khi bật (on) tính năng tự động hoàn thành, trình duyệt sẽ tự động hoàn thành các giá trị dựa trên các giá trị mà người dùng đã nhập trước đó.

Ví dụ:

```
<form action="/action_page.php" autocomplete="on">
```

7.2. Các thẻ của biểu mẫu:

<form> HTML có thể chứa một hoặc nhiều thẻ biểu mẫu sau:

- <input>
- <label>
- <select>
- <textarea>
- <button>
- <fieldset>
- <legend>
- <datalist>
- <output>
- <option>
- <optgroup>

7.2.1. Thẻ input:

Một trong những thẻ biểu mẫu được sử dụng nhiều nhất là thẻ <input>. Thẻ <input> có thể được hiển thị theo nhiều cách, tùy thuộc vào thuộc tính type.

Ví dụ: Với khai báo như sau

```
<label for="fname">First name:</label>
```

```
<input type="text" id="fname" name="fname">
```

Kết quả sẽ là:

First name:

Dưới đây là các kiểu nhập khác nhau mà bạn có thể sử dụng trong HTML:

Kiểu nhập	Diễn giải
<code><input type="button"></code>	Xác định là 1 nút
<code><input type="checkbox"></code>	Xác định một hộp kiểm
<code><input type="color"></code>	Được sử dụng cho các lĩnh vực đầu vào mà nên chứa một màu. Tùy thuộc vào sự hỗ trợ của trình duyệt, bộ chọn màu có thể hiển thị trong trường nhập liệu
<code><input type="date"></code>	Được sử dụng cho các lĩnh vực đầu vào mà nên chứa một ngày. Tùy thuộc vào sự hỗ trợ của trình duyệt, bộ chọn ngày có thể hiển thị trong trường nhập liệu
<code><input type="datetime-local"></code>	Chỉ định trường nhập ngày và giờ, không có múi giờ. Tùy thuộc vào sự hỗ trợ của trình duyệt, bộ chọn ngày có thể hiển thị trong trường nhập liệu
<code><input type="email"></code>	Được sử dụng cho các lĩnh vực đầu vào mà nên chứa một địa chỉ e-mail. Tùy thuộc vào sự hỗ trợ của trình duyệt, địa chỉ e-mail có thể được tự động xác nhận khi được gửi. Một số điện thoại thông minh nhận dạng loại email và thêm ".com" vào bàn phím để khớp với kiểu nhập email
<code><input type="file"></code>	Xác định trường chọn tệp và nút "Browse" để tải tệp lên
<code><input type="hidden"></code>	Nghĩa một trường nhập ẩn (không hiển thị cho người dùng). Trường ẩn cho phép các nhà phát triển web bao gồm dữ liệu mà người dùng không thể nhìn thấy hoặc sửa đổi khi gửi biểu mẫu.
<code><input type="month"></code>	Phép người dùng lựa chọn một tháng và năm. Tùy thuộc vào sự hỗ trợ của trình duyệt, bộ chọn ngày có thể hiển thị trong trường nhập liệu

<input type="number">	<p>Nghĩa một trường nhập số .</p> <p>Bạn cũng có thể đặt các hạn chế về những con số nào được chấp nhận, các hạn chế thường dùng là: min, max, step, value ...</p>
<input type="password">	Xác định một trường mật khẩu
<input type="radio">	<p>Xác định một nút radio .</p> <p>Các nút radio cho phép người dùng CHỈ chọn MỘT trong số các lựa chọn giới hạn</p>
<input type="range">	<p>Xác định một điều khiển để nhập một số có giá trị chính xác không quan trọng (như điều khiển thanh trượt). Phạm vi mặc định là 0 đến 100. Tuy nhiên, bạn có thể thiết lập các hạn chế về những gì con số được chấp nhận với min, max và step các thuộc tính</p>
<input type="reset">	Xác định nút đặt lại sẽ đặt lại tất cả các giá trị biểu mẫu về giá trị mặc định của chúng
<input type="search">	Được sử dụng cho các lĩnh vực tìm kiếm (một cur xử lĩnh vực tìm kiếm như một trường văn bản thường)
<input type="submit">	<p>Xác định một nút để gửi dữ liệu biểu mẫu đến trình xử lý biểu mẫu .</p> <p>Trình xử lý biểu mẫu thường là một trang máy chủ có tập lệnh để xử lý dữ liệu đầu vào.</p> <p>Trình xử lý biểu mẫu được chỉ định trong action thuộc tính của biểu mẫu :</p>
<input type="tel">	Được sử dụng cho các lĩnh vực đầu vào mà nên chứa một số điện thoại.
<input type="text">	Xác định trường nhập văn bản một dòng
<input type="time">	<p>Phép người dùng lựa chọn một thời gian (không múi giờ).</p> <p>Tùy thuộc vào sự hỗ trợ của trình duyệt, bộ chọn thời gian có thể hiển thị trong trường nhập liệu.</p>
<input type="url">	<p>Được sử dụng cho các lĩnh vực đầu vào mà nên chứa một địa chỉ URL.</p> <p>Tùy thuộc vào sự hỗ trợ của trình duyệt, trường url có thể được tự động xác thực khi được gửi.</p>

	Một số điện thoại thông minh nhận ra loại url và thêm ".com" vào bàn phím để khớp với đầu vào url.
<code><input type="week"></code>	Phép người dùng lựa chọn một tuần và năm. Tùy thuộc vào sự hỗ trợ của trình duyệt, bộ chọn ngày có thể hiển thị trong trường nhập liệu.

7.2.2. Thẻ label:

`<label>` xác định nhãn cho một số phần tử biểu mẫu. Hữu ích cho người dùng trình đọc màn hình, vì trình đọc màn hình sẽ đọc to nhãn khi người dùng tập trung vào phần tử đầu vào.

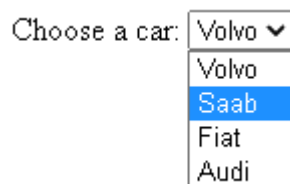
`<label>` cũng giúp người dùng gặp khó khăn khi nhấp vào các vùng rất nhỏ (chẳng hạn như nút radio hoặc hộp kiểm) - bởi vì khi người dùng nhấp vào văn bản trong `<label>`, nó sẽ chuyển nút radio / hộp kiểm.

7.2.3. Thẻ select:

Xác định danh sách thả xuống (combo box).

Ví dụ:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```



Các `<option>` xác định một tùy chọn có thể được chọn. Theo mặc định, mục đầu tiên trong danh sách thả xuống được chọn. Để xác định một tùy chọn đã chọn trước, hãy thêm `selected` vào tùy chọn:

Ví dụ:

```
<option value="fiat" selected>Fiat</option>
```


Sử dụng size để chỉ định số lượng giá trị hiển thị và multiple để cho phép người dùng chọn nhiều giá trị

Ví dụ:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

7.2.4. Thẻ textarea:

Xác định trường nhập nhiều dòng (vùng văn bản). Có rows xác định số có thể nhìn thấy các dòng trong một vùng văn bản, cols xác định chiều rộng hữu hình của một vùng văn bản.

Ví dụ:

```
<textarea name="message" rows="10" cols="30">
```

Ghi chú.

```
</textarea>
```



7.2.5. Thẻ button:

Xác định một nút có thể nhấp

Ví dụ:

```
<button type="button" onclick="alert('Xin chào!')">Nhấn
vào đây!</button>
```

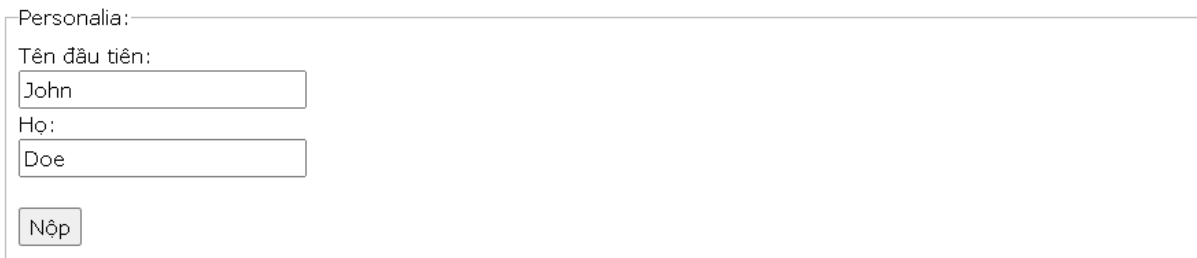
Nhập vào đây!

7.2.6. Thẻ `<fieldset>` và `<legend>`:

`<fieldset>` được sử dụng để nhóm dữ liệu liên quan trong một biểu mẫu. `<legend>` xác định chú thích cho `<fieldset>` phần tử.

Ví dụ:

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname"
value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname"
value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



7.2.7. Thẻ `<datalist>`:

`<datalist>` chỉ định danh sách các tùy chọn được xác định trước cho một `<input>`. Người dùng sẽ thấy danh sách thả xuống gồm các tùy chọn được xác định trước khi họ nhập dữ liệu. Các list của `<input>`, phải tham khảo các id của `<datalist>`.

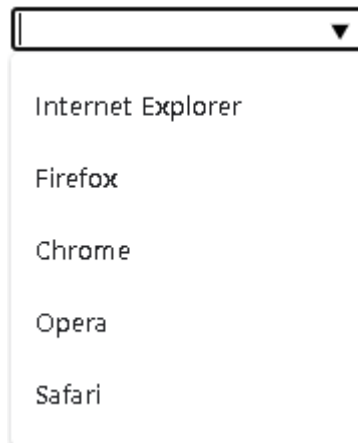
Ví dụ:

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
```

```

    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>

```



7.2.8. Thẻ output:

Đại diện cho kết quả của một phép tính (giống như một phép tính được thực hiện bởi tập lệnh).

Ví dụ:

```

<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>

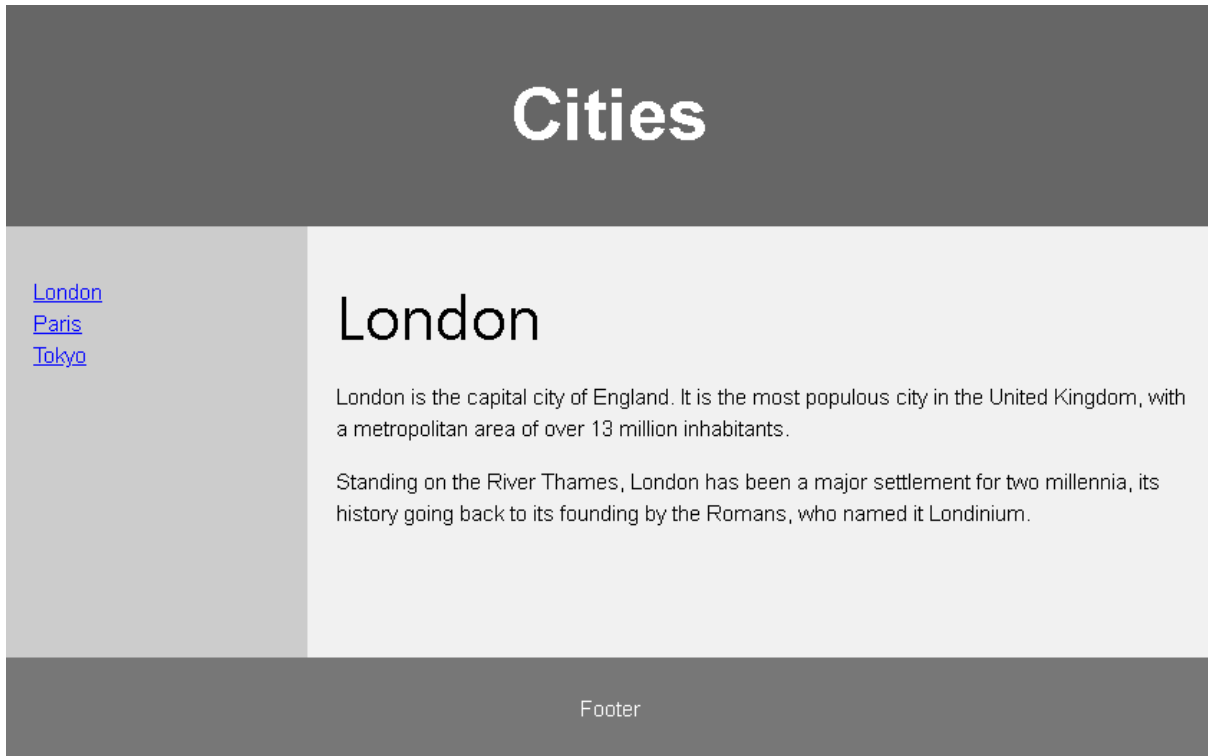
```

0  100 + = 106

Submit

Bài tập:

Thiết kế trang web như hình sau:



Chú ý: 3 thành phố trên thanh sidebar bên trái có link, link có thể # khi thực hiện. Phần màu sắc vì chưa học CSS nên có thể bỏ qua hoặc thực hiện thay đổi thuộc tính style="background-color: tên màu" để hoàn thành bài.

Bài tập nâng cao:

Thiết kế trang Web theo hình sau:



Tạo Tài khoản Google

Bạn có thể sử dụng chữ cái, số và dấu chấm

Sử dụng địa chỉ email hiện tại của tôi

Sử dụng 8 ký tự trở lên và kết hợp chữ cái, chữ số và biểu tượng

Hiện mật khẩu



Một tài khoản. Hoạt động trên mọi sản phẩm và dịch vụ của Google.

[Đăng nhập](#)

[Tiếp theo](#)

Chú ý: phần hình ảnh có thể thay bằng hình khác, bài này kiểm tra kiến thức thiết kế web không có mục đích vi phạm bản quyền.

Những trọng tâm cần chú ý trong bài:

- Sử dụng được các thẻ HTML thông dụng.
- Sử dụng được các thẻ HTML 5.
- Xây dựng được giao diện trang web hoàn chỉnh dùng các thẻ HTML 5

Yêu cầu về đánh giá kết quả học tập:

Nội dung:

- + Về kiến thức: Trình bày được cấu trúc của các HTML thông dụng, một số thẻ HTML thường dùng
- + Về kỹ năng: Sử dụng được các thẻ HTML thiết kế 1 giao diện Website hoàn chỉnh
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

- + Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp
- + Về kỹ năng: Đánh giá kỹ sử dụng các thẻ HTML kết hợp với nhau tạo thành trang web.
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 3: CSS (Cascading Style Sheets)

Mã bài: MĐ 11 - 03

Giới thiệu:

Bài học này nhằm giới thiệu cách sử dụng CSS (Cascading Style Sheets) và các thuộc tính thường dùng trong CSS để hoàn thiện một trang web

Mục tiêu của bài:

- Áp dụng được CSS vào bên trong giao diện trang web..
- Sử dụng thành thạo các thuộc tính khác nhau để trình bày giao diện trang web..
- Hoàn thành một giao diện trang web hoàn chỉnh.

Nội dung chính:

1. Cách sử dụng:

Cascading Style Sheets (CSS) được sử dụng để định dạng bố cục của một trang web. Với CSS, chúng ta có thể kiểm soát màu sắc, phông chữ, kích thước của văn bản, khoảng cách giữa các phần tử, cách các phần tử được định vị và bố trí, hình ảnh nền hoặc màu nền sẽ được sử dụng, các màn hình khác nhau cho các thiết bị và kích thước màn hình khác nhau, và nhiều hơn nữa.

CSS có thể được thêm vào tài liệu HTML theo 3 cách:

Nội tuyến - bằng cách sử dụng style bên trong các phần tử HTML

Nội bộ - bằng cách sử dụng một thẻ <style> trong thẻ <head> ở đầu trang Web

Bên ngoài - bằng cách sử dụng một <link> phần tử để liên kết đến một tệp CSS bên ngoài

1.1. Các định nghĩa thường dùng trong CSS:

Khi sử dụng CSS để định dạng, chúng ta cần khai báo các thuộc tính CSS cho thẻ hoặc class hoặc id.

- Thẻ: chính là thẻ trong HTML

- Class: Khai báo bằng cách đặt dấu chấm (ví dụ: .class), class này có thể dùng áp dụng cho thẻ HTML bằng cách gọi thuộc tính class của thẻ.

Ví dụ: <p class= “doanvanban”>Đoạn 1</p>

Khi dùng class chúng ta có thể kết hợp nhiều class với nhau

Ví dụ: <p class= “doanvanban indam inngiang”>Đoạn 1</p>

- ID: Khai báo bằng cách đặt dấu # (ví dụ: #id), id này có thể dùng áp dụng cho thẻ HTML bằng cách gọi thuộc tính ID của thẻ

- Ví dụ: `<p id= "doanvanban">Đoạn 1</p>`

Khi dùng ID chúng ta chỉ có thể gọi 1 ID cho 1 thẻ.

1.2. Cách dùng CSS:

- Nội tuyến: CSS nội tuyến được sử dụng để áp dụng một kiểu duy nhất cho một phần tử HTML. CSS nội tuyến sử dụng style của một phần tử HTML.

Ví dụ sau đặt màu văn bản của `<h1>` phần tử thành màu xanh lam và màu văn bản của `<p>` phần tử thành màu đỏ:

Ví dụ:

```
<h1 style="color:blue;">A Blue Heading</h1>
```

```
<p style="color:red;">A red paragraph.</p>
```

A Blue Heading

A red paragraph.

- Nội bộ: CSS nội bộ được sử dụng để xác định kiểu cho một trang HTML. CSS nội bộ được định nghĩa trong `<head>` phần của trang HTML, bên trong một `<style>` phần tử.

Ví dụ sau đặt màu văn bản của TẤT CẢ các `<h1>` phần tử (trên trang đó) thành màu xanh lam và màu văn bản của TẤT CẢ các `<p>` phần tử thành màu đỏ. Ngoài ra, trang sẽ được hiển thị với màu nền "bột màu":

Ví dụ:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```



```
</body>
```

```
</html>
```

- Bên ngoài: Biểu định kiểu bên ngoài được sử dụng để xác định kiểu cho nhiều trang HTML. Để sử dụng biểu định kiểu bên ngoài, hãy thêm liên kết đến biểu định kiểu này trong <head> của mỗi trang HTML. Cú pháp link:

```
<link rel="stylesheet" href="file.css" type="text/css">
```

Ví dụ:

File HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <link rel="stylesheet" href="styles.css">
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

File style.css

```
body {
```

```
    background-color: powderblue;
```

```
}
```

```
h1 {
```

```
    color: blue;
```

```
}
```

```
p {
```

```
    color: red;
```

```
}
```

2. Các thuộc tính thường dùng:

CSS có rất nhiều thuộc tính để định kiểu cho phần hiển thị của HTML, sau đây sẽ trình bày một số thuộc tính thông dụng

2.1. CSS Backgrounds:

Thuộc tính nền CSS được sử dụng để thêm hiệu ứng nền cho các thẻ

- Màu nền CSS: Các background-color quy định màu nền của một thẻ

Ví dụ: `body { background-color: lightblue; }`

Chú ý: Trong CSS, một màu có thể được chỉ định bằng cách sử dụng tên màu, giá trị RGB, giá trị HEX, giá trị HSL, giá trị RGBA và giá trị HSLA.

Ví dụ: `body { background-color: #ff0000; }`

- Hình nền CSS: Các background-image quy định cụ thể một hình ảnh để sử dụng như là nền tảng của một thẻ. Theo mặc định, hình ảnh được lặp lại để nó bao phủ toàn bộ thẻ.

Ví dụ: `body { background-image: url("paper.gif"); }`

Trong đó url chính là đường dẫn đến hình như khi dùng thẻ img.

- Hình nền lặp: như đã nói ở trên, khi dùng hình nền thì mặc định hình nền sẽ lặp lại để lấp đầy màn hình. Tuy nhiên, chúng ta có thể thay đổi điều đó bằng cách dùng thuộc tính background-repeat

Ví dụ:

```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;  
}
```

Khi lặp chúng ta có các giá trị: repeat-x, repeat-y, no-repeat có nghĩa tuần tự là lặp theo chiều ngang, theo chiều dọc và không lặp

- Vị trí nền CSS: Các background-position được sử dụng để xác định vị trí của các hình nền.

Ví dụ:

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Hình nền sẽ được neo tại vị trí bên phải phía trên.

Chú ý: có thể kết hợp các thuộc tính liên quan đến background-image trên 1 dòng

Ví dụ: `body { background: #ffffff url("img_tree.png") no-repeat right top; }`

2.2. CSS Borders:

Thuộc tính đường viền CSS cho phép bạn chỉ định kiểu, chiều rộng và màu sắc của đường viền của một thẻ.

Tôi có biên giới ở tất cả các bên.

Tôi có một đường viền dưới cùng màu đỏ.

Tôi có đường viền tròn.

Tôi có một đường viền bên trái màu xanh lam.

- Kiểu đường viền (border-style)

dotted	Xác định một đường viền chấm
dashed	Xác định đường viền đứt nét
solid	Xác định một đường viền vững chắc
double	Xác định đường viền kép
groove	Xác định đường viền có rãnh 3D. Hiệu ứng phụ thuộc vào giá trị màu viền
ridge	Xác định đường viền 3D. Hiệu ứng phụ thuộc vào giá trị màu viền
inset	Xác định đường viền in 3D. Hiệu ứng phụ thuộc vào giá trị màu viền
outset	Xác định đường viền ban đầu 3D. Hiệu ứng phụ thuộc vào giá trị màu viền
none	Xác định không có đường viền
hidden	Xác định một đường viền ẩn

Ví dụ:

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
```

```
p.hidden {border-style: hidden;}
```

```
p.mix {border-style: dotted dashed solid double;}
```

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

- Chiều rộng đường viền CSS (border-width): Các border-width tài sản quy định độ rộng của bốn biên giới. Chiều rộng có thể được đặt thành một kích thước cụ thể (tính bằng px, pt, cm, em, v.v.) hoặc bằng cách sử dụng một trong ba giá trị được xác định trước: mỏng, trung bình hoặc dày (thin, medium, thick)

Ví dụ:

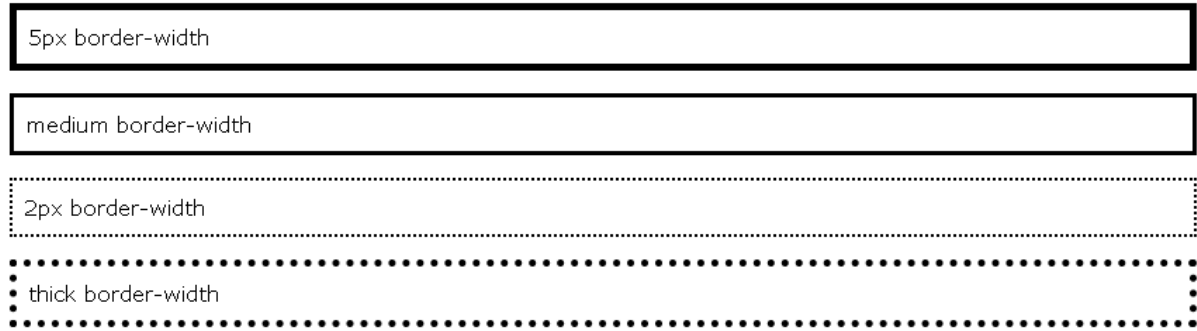
```
p.one {  
  border-style: solid;  
  border-width: 5px;  
}
```

```
p.two {  
  border-style: solid;  
  border-width: medium;  
}
```

```
p.three {  
  border-style: dotted;  
  border-width: 2px;  
}
```

```
p.four {  
  border-style: dotted;  
  border-width: thick;  
}
```

}



Các border-width có thể có từ một đến bốn giá trị (biên trên, biên phải, biên dưới, và biên trái):

Ví dụ:

```
p.three {  
  border-style: solid;  
  border-width: 25px 10px 4px 35px;  
}
```

- Màu viền CSS (border-color): được sử dụng để thiết lập màu sắc của bốn biên, các border-color có thể có từ một đến bốn giá trị (trên, phải, dưới, trái).

- Đường viền CSS - các mặt riêng lẻ: có thể chỉ định một đường viền khác nhau cho mỗi bên (top, right, bottom, left) ta có border-top-style, border-right-style, border-bottom-style, border-left-style.

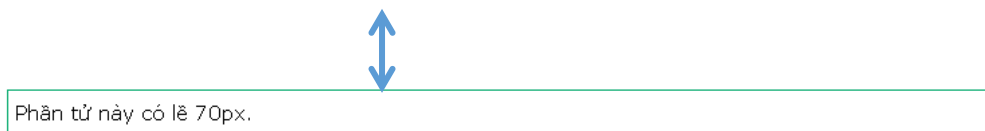
Chú ý: có thể kết hợp các thuộc tính của border trên một dòng

Ví dụ:

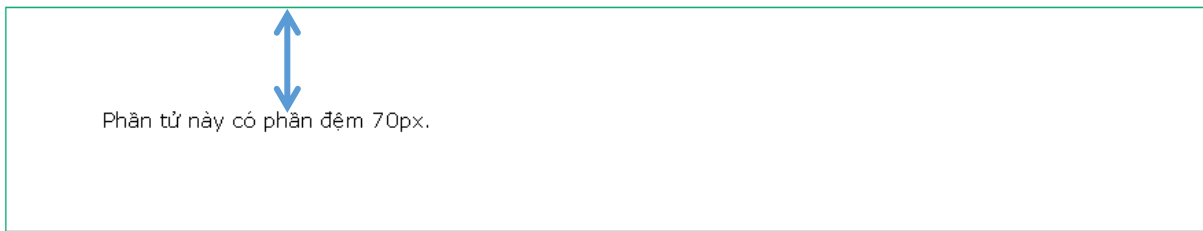
```
p { border: 5px solid red; }  
p { border-bottom: 6px solid red; }
```

2.3. Margin và Padding:

- Margin CSS: được sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền xác định nào. Có thể có 4 bên như đường viền là top, right, bottom, left



- Padding CSS: được sử dụng để tạo không gian xung quanh nội dung của phần tử, bên trong bất kỳ đường viền xác định nào. Có thể có 4 bên như đường viền là top, right, bottom, left



2.4. Chiều cao và chiều rộng (Height & Width):

CSS `height` và `width` thuộc tính được sử dụng để đặt chiều cao và chiều rộng của một thẻ. Có thể dùng từ khóa `min` hoặc `max` phía trước `height` - `width` để đặt giá trị tối thiểu và tối đa cho chiều rộng và chiều cao cho 1 thẻ.

Ví dụ:

```
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
div {
  min-height: 200px;
  max-width: 50%;
  background-color: powderblue;
}
```

2.5. Văn bản:

Văn bản có thể đặt ở bất kỳ đâu, sau đây là các thuộc tính có thể áp dụng lên văn bản

- Màu sắc (`color`): định màu sắc cho văn bản, có thể sử dụng màu sắc là tên hoặc các bảng màu như màu nền.

- Canh lề văn bản (`text-align`): Văn bản có thể được căn trái, căn phải, căn giữa, căn đều (`left`, `right`, `center`, `justify`)

Ví dụ:

```
h1 { text-align: center; }
h2 { text-align: left; }
h3 { text-align: right; }
h4 { text-align: justify; }
```

- Canh lề theo chiều dọc (`vertical-align`): thiết lập sự liên kết dọc của một thẻ, có các giá trị là trên 1 dòng, trên văn bản, dưới văn bản, trên 1 mức, dưới 1 mức (`baseline`, `text-top`, `text-bottom`, `super`, `sub`)

- Kiểu cho văn bản: có các kiểu in đậm, in nghiêng, gạch dưới

+ In đậm: `font-weight` có giá trị là `bold`.

+ In nghiêng: `font-style` có giá trị là `italic`

+ Gạch dưới: text-decoration có giá trị là underline

- Font chữ (font-family): Các font-family giữ nhiều tên phong chữ như một hệ thống "dự phòng", để đảm bảo khả năng tương thích tối đa giữa các trình duyệt / hệ điều hành. Bắt đầu với phong chữ bạn muốn và kết thúc bằng một họ chung (để cho phép trình duyệt chọn một phong chữ tương tự trong họ chung, nếu không có sẵn các phong chữ khác). Tên phong chữ nên được phân tách bằng dấu phẩy.

Ví dụ:

```
.p1 { font-family: "Times New Roman", Times, serif; }
```

→ font chữ mong muốn là Times New Roman

- Cỡ chữ (font-size): có 2 dạng thông dụng là px (định giá trị tuyệt đối) và em (định giá trị tương đối - tùy theo kích thước chữ của trình duyệt (mặc định của trình duyệt là 16px)

Ví dụ:

```
h1 { font-size: 40px; } → Cỡ chữ là 40px
```

```
h2 { font-size: 2.5em; } → Cỡ chữ là 40px vì 40/16=2.5
```

2.6. Liên kết (thẻ a):

Liên kết có thể được chỉnh với bất cứ CSS nào (ví dụ như color, font-family, background, vv). Ngoài ra, liên kết có 1 số thuộc tính riêng là

a:link	một liên kết bình thường, không được kiểm tra
a:visited	một liên kết mà người dùng đã truy cập
a:hover	một liên kết khi người dùng di chuột qua nó
a:active	một liên kết ngay khi nó được nhấp vào

2.7. Hiện thị (Display):

Display quy định cụ thể làm thế nào một thẻ được hiển thị. Mọi phần tử HTML đều có giá trị hiển thị mặc định tùy thuộc vào loại phần tử đó. Giá trị hiển thị mặc định cho hầu hết các phần tử là block hoặc inline.

Một phần tử cấp khối (block) luôn bắt đầu trên một dòng mới và chiếm toàn bộ chiều rộng có sẵn (trái dài sang trái và phải hết mức có thể).

Một phần tử nội tuyến (inline) không bắt đầu trên một dòng mới và chỉ chiếm nhiều chiều rộng khi cần thiết.

Display có các giá trị là: block, inline, none.

Ví dụ: a { display: block; }

Chú ý: với display là none, thẻ sẽ bị ẩn đi và không ảnh hưởng bố cục, nếu sử dụng thuộc tính visibility là hidden, thẻ cũng ẩn đi nhưng sẽ ảnh hưởng bố cục.

2.8. Vị trí (position):

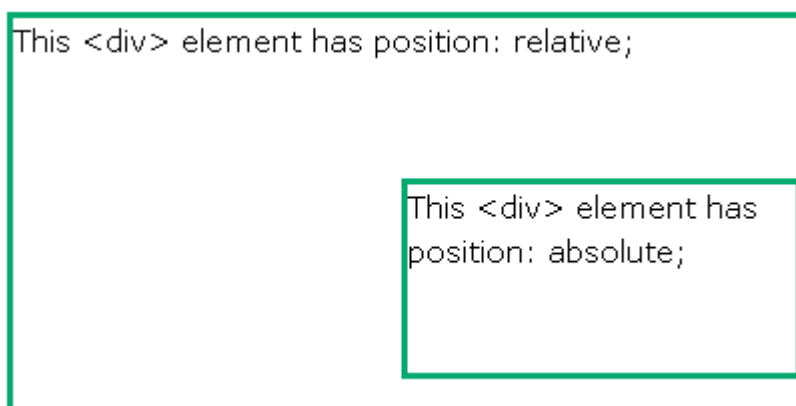
Quy định cụ thể các loại phương pháp định vị sử dụng cho một thẻ là tĩnh, tương đối, cố định, tuyệt đối hay dính (static, relative, fixed, absolute, sticky)

- static: Các thẻ được định vị tĩnh không bị ảnh hưởng bởi các thuộc tính top, bottom, left và right. Một phần tử với position: static; không được định vị theo bất kỳ cách đặc biệt nào; nó luôn được định vị theo dòng bình thường của trang:

- relative: Được định vị so với vị trí bình thường của nó. Đặt các thuộc tính top, right, bottom và left của một phần tử có vị trí tương đối sẽ khiến nó bị điều chỉnh khỏi vị trí bình thường của nó. Các nội dung khác sẽ không được điều chỉnh để phù hợp với bất kỳ khoảng trống nào do phần tử để lại.

- fixed: Được định vị so với chế độ xem, có nghĩa là nó luôn ở cùng một vị trí ngay cả khi trang được cuộn. Các thuộc tính top, right, bottom và left được sử dụng để định vị thẻ

- absolute: Được định vị tương đối với thẻ trên nó được định vị gần nhất. Nếu trên nó không có thẻ khác thì sẽ lấy body làm thẻ trên.



- sticky: Được định vị dựa trên vị trí cuộn của người dùng. Một phần tử cố định chuyển đổi giữa relative và fixed, tùy thuộc vào vị trí cuộn. Nó được định vị tương đối cho đến khi một vị trí bù nhất định được đáp ứng trong khung nhìn - sau đó nó "đính" vào vị trí (như vị trí: fixed)

2.9. Float:

Thuộc tính CSS chỉ định cách một phần tử sẽ nổi. Các float được sử dụng để định vị và định dạng nội dung ví dụ như để cho một hình ảnh để lại cho văn bản trong một container. Các float có thể có một trong các giá trị sau:

left	Phần tử nổi ở bên trái vùng chứa của nó
right	Phần tử nổi ở bên phải vùng chứa của nó
none	Phần tử không nổi (sẽ được hiển thị ngay tại nơi nó xuất hiện trong văn bản). Đây là mặc định
inherit	Phần tử kế thừa giá trị float của phần tử cha của nó

Ví dụ: float của hình ảnh là right

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus unadict, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices ne congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, favisis sed ornare eu, viem tuyền bã ở odio. Praesent bim bim biếc urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac ...



Bài tập:

1. Thiết kế trang web như hình sau:



Simple and useful HTML layout

There is a clickable image with beautiful hover effect and active title link for each post item. Left side is a sticky menu bar. Right side is a blog content that will scroll up and down.

Travel . Events

June 24, 2045

36 comments

by Admin Nat



Multi-purpose blog template

Xtra Blog is a multi-purpose HTML CSS template from TemplateMo website. Blog list, single post, about, contact pages are included. Left sidebar fixed width and content area is a fluid full-width.

Creative . Design . Business

June 16, 2045

48 comments

by Admin Sam

Chú ý: Phần hình ảnh có thể thay bằng hình khác, nội dung có thể copy - không quan trọng, tập trung vào phần bố cục

2. Truy cập đường link: <https://baocantho.com.vn/>, tiến hành thực hiện tạo menu giống như hình bên dưới.



Chú ý: Các thông tin có thể thay đổi, bài này chủ yếu là tạo được menu.

Bài tập nâng cao:

Tham khảo trang web của Trường Cao Đẳng Nghề Cần Thơ, thực hiện thiết kế 1 trang Web hoàn chỉnh giống như vậy.

Chú ý: phần hiệu ứng có thể bỏ qua.

Những trọng tâm cần chú ý trong bài:

- Áp dụng được CSS vào bên trong giao diện trang web..
- Sử dụng thành thạo các thuộc tính khác nhau để trình bày giao diện trang web..
- Hoàn thành một giao diện trang web hoàn chỉnh

Yêu cầu về đánh giá kết quả học tập:**Nội dung:**

- + Về kiến thức: Trình bày được các cách sử dụng CSS
- + Về kỹ năng: Sử dụng được các thuộc tính của CSS để thiết kế 1 giao diện Website hoàn chỉnh
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

- + Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp
- + Về kỹ năng: Đánh giá kỹ sử dụng các thẻ HTML kết hợp với CSS tạo thành trang web.
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

BÀI 4: Javascript

Mã bài: MĐ 11 - 04

Giới thiệu:

Bài học này giới thiệu javascript và cách sử dụng javascript trên 1 trang web.

Mục tiêu của bài:

- Áp dụng được javascript vào bên trong giao diện trang web..
- Sử dụng thành thạo javascript để tạo hiệu ứng cho trang web.

Nội dung chính:

1. Cách sử dụng:

Trong HTML, mã JavaScript được chèn vào giữa các thẻ <script> và </script>. Có thể đặt bất kỳ số lượng script nào trong một tài liệu HTML. Tập lệnh có thể được đặt trong <body> hoặc trong <head> phần của trang HTML hoặc trong cả hai

Ví dụ:

```
<script>

document.getElementById("demo").innerHTML = "My First
JavaScript";

</script>

<!DOCTYPE html>

<html>

<head>

<script>

function myFunction() {

    document.getElementById("demo").innerHTML = "Paragraph
changed.";

}

</script>

</head>

<body>

<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph</p>
```

```
<button type="button" onclick="myFunction()">Try  
it</button>  
  
</body>  
  
</html>
```

Chúng ta cũng có thể tạo 1 tập tin bên ngoài và nhúng vào trang web như css như sau:

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

Trong đó, myScript.js là tập tin javascript.

2. Kiến thức cơ bản về ngôn ngữ javascript:

- Khai báo biến bằng các từ khóa var, let, const
- Dấu = để gán trị
- Các phép toán sử dụng thông thường +, -, *, /
- Chuỗi trong javascript để trong dấu "", có thể kết hợp chuỗi bằng dấu +
- Javascript có phân biệt chữ hoa và chữ thường
- Ghi chú có thể dùng // cho 1 dòng hoặc /* */ cho 1 đoạn
- Khai báo hàm dùng từ khóa function - hàm có thể trả kết quả hoặc không

3. Kết hợp javascript với HTML:

- Một trong nhiều phương pháp kết hợp HTML và JavaScript là getElementById().

Ví dụ: document.getElementById("demo").innerHTML = "Hello JavaScript";

→ tìm một thẻ HTML (với id = "demo") và thay đổi nội dung phần tử (innerHTML) thành "Hello JavaScript"

- JavaScript có thể "hiển thị" dữ liệu theo những cách khác nhau: innerHTML, document.write(), window.alert(), console.log(). Thông thường chúng ta chỉ dùng innerHTML (viết nội dung vào thẻ HTML) và window.alert (hiện hộp thoại cảnh báo)

- Sự kiện HTML là "những thứ" xảy ra với các phần tử HTML. Khi JavaScript được sử dụng trong các trang HTML, JavaScript có thể "phản ứng" trên các sự kiện này. Các sự kiện thường gặp là:

Sự kiện	Sự miêu tả
onchange	Một phần tử HTML đã được thay đổi
onclick	Người dùng nhấp vào một phần tử HTML
onmouseover	Người dùng di chuyển chuột qua một phần tử HTML

onmouseout

Người dùng di chuyển chuột khỏi phần tử HTML

onkeydown

Người dùng nhấn một phím bàn phím

onload

Trình duyệt đã tải xong trang

Ví dụ:

```
<button onclick="this.innerHTML = Date()">The time is?</button>
```

Chú ý: javascript có thể thay đổi thuộc tính của 1 thẻ HTML

Ví dụ:

```

```

```
<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn on the light</button>
```

→ Đoạn javascript có nghĩa là: khi sự kiện click của button được gọi, nó sẽ kiểm tra HTML có ID là myImage và tiến hành đổi thuộc tính src của thẻ đó.

Bài tập:

1. Thiết kế trang web có 1 input dạng text và 1 button. Tạo javascript cho sự kiện nút button được click. Khi button được click sẽ hiển thị nội dung trong ô text ra hộp cảnh báo alert

Chú ý: Phần này có dùng window.alert()

Bài tập nâng cao:

Tạo trang web có 1 máy tính cầm tay, dùng thực hiện các phép toán cơ bản +, -, *, /.

Những trọng tâm cần chú ý trong bài:

- Áp dụng được javascript vào bên trong giao diện trang web..
- Sử dụng thành thạo javascript để tạo hiệu ứng cho trang web.

Yêu cầu về đánh giá kết quả học tập:

Nội dung:

- + Về kiến thức: Trình bày được các cách sử dụng javascript
- + Về kỹ năng: Sử dụng thành thạo các sự kiện cơ bản khác nhau để áp dụng vào các thẻ của trang web
- + Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

Phương pháp:

- + Về kiến thức: Được đánh giá bằng hình thức kiểm tra viết, trắc nghiệm, vấn đáp

+ Về kỹ năng: Đánh giá kỹ sử dụng các thẻ HTML kết hợp với CSS tạo thành trang web.

+ Năng lực tự chủ và trách nhiệm: Tỉ mỉ, cẩn thận, chính xác, ngăn nắp trong công việc.

TÀI LIỆU THAM KHẢO

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

[2]. Kỹ sư Phạm Quang Huy , Giáo trình thiết kế Web và làm hoạt hình, NXB Giao thông vận tải, năm 2003