

Chương 2:**NGÔN NGỮ CSS****1. TỔNG QUAN VỀ CSS****1.1. Giới thiệu****1.1.1. CSS là gì?**

Trong lĩnh vực xây dựng, chúng ta có trang trí nội thất, trong lĩnh vực thẩm mỹ - làm đẹp, chúng ta có kỹ thuật make-up; còn trong lĩnh vực thiết kế web chúng ta có CSS. CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML,...

1.1.2. Tại sao CSS?

Ngôn ngữ HTML cũng có số thuộc tính định dạng cơ bản cho text, picture, table, ... nhưng nó không thật sự phong phú và chính xác như nhau trên mọi hệ thống. CSS cung cấp cho bạn hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo trong kết hợp các thuộc tính giúp mang lại hiệu quả cao. Ngoài ra, CSS đã được hỗ trợ bởi tất cả các trình duyệt, nên bạn hoàn toàn có thể tự tin trang web của mình có thể hiển thị “như nhau” trên mọi hệ điều hành.

Sử dụng các mã định dạng trực tiếp trong HTML tốn hao nhiều thời gian thiết kế cũng như dung lượng lưu trữ. Trong khi đó CSS đưa ra phương thức áp dụng một khuôn mẫu chung từ một file CSS ở ngoài. Có hiệu quả đồng bộ khi bạn tạo một website có hàng trăm trang hay cả khi bạn muốn thay đổi một thuộc tính trình bày nào đó.

CSS được cập nhật liên tục mang lại các trình bày phức tạp và tinh vi hơn.

1.1.3. Học CSS cần những gì?

- Có là một kiến thức về HTML.

- Một trình soạn thảo văn bản để bạn có thể viết mã CSS. Nên sử dụng một trình soạn thảo đơn giản như Notepad trong Windows hay Pico trong Linux, Simple Text trong Mac. Hay từ các chương trình như khi dùng DreamWeaver, FrontPage, Golive,...

- Một trình duyệt web.

1.2. Một số quy ước về cách viết CSS**1.2.1. Cú pháp CSS:**

Để tìm hiểu cú pháp CSS chúng ta hãy thử xem một ví dụ sau.

Ví dụ: Để định màu nền cho một trang web là xanh nhạt (light cyan):

+ Trong HTML: `<body bgcolor="#00BFF3">`

+ Trong CSS: `body { background-color:#00BFF3; }`

Qua ví dụ trên ta thấy được mối tương đồng giữa các thuộc tính trong HTML và CSS.

Cú pháp CSS cơ bản: `Selector { property:value; }`

Trong đó:

+ Selector: Các đối tượng mà chúng ta sẽ áp dụng các thuộc tính trình bày.

+ Property: Chính là các thuộc tính quy định cách trình bày. Như: background-color, font-family, color, padding, margin,...

Mỗi thuộc tính CSS phải được gán một giá trị. Nếu có nhiều hơn một thuộc tính cho một selector thì chúng ta phải dùng một dấu ; (chấm phẩy) để phân cách các thuộc tính. Tất cả các thuộc tính trong một selector sẽ được đặt trong một cặp ngoặc nhọn sau selector.

Ví dụ: `body { background:#FFF; color:#FF0000; font-size:14pt }`

Để dễ đọc hơn, bạn nên viết mỗi thuộc tính CSS ở một dòng. Tuy nhiên, nó sẽ làm tăng dung lượng lưu trữ CSS của bạn.

Đối với một trang web có nhiều thành phần có cùng một số thuộc tính, chúng ta có thể thực hiện gom gọn lại như sau:

`h1 { color:#0000FF; text-transform:uppercase }`

`h2 { color:#0000FF; text-transform:uppercase; }`

`h3 { color:#0000FF; text-transform:uppercase; }`

➔ `h1, h2, h3 { color:#0000FF; text-transform:uppercase; }`

+ Value: Giá trị của thuộc tính. Như ví dụ trên value chính là #FFF dùng để định màu trắng cho nền trang.

Đối với một giá trị có khoảng trắng, bạn nên đặt tất cả trong một dấu ngoặc kép.

Ví dụ: `font-family:"Times New Roman"`

Đối với giá trị là đơn vị đo, không nên đặt một khoảng cách giữa số đo với đơn vị của nó.

Ví dụ: `width:100 px`. Nó sẽ làm CSS bị vô hiệu trên Mozilla/Firefox hay Netscape.

Chú thích trong CSS:

Cũng như nhiều ngôn ngữ web khác. Trong CSS, chúng ta cũng có thể viết chú thích cho các đoạn code như sau `/* Nội dung chú thích */`

Ví dụ: `/* Màu chữ cho trang web là màu đỏ */` `body { color:red }`

1.2.2. Đơn vị CSS:

Bảng liệt kê các đơn vị chiều dài và màu sắc dùng trong CSS.

Đơn vị chiều dài

Đơn vị	Mô tả
%	Phần trăm
In	Inch (1 inch = 2.54 cm)
cm	Centimeter
mm	Millimeter
pc	Pica (1 pc = 12 pt)
px	Pixels (điểm ảnh trên màn hình máy tính)
pt	Point (1 pt = 1/72 inch)
em	1 em tương đương kích thước font hiện hành, nếu font hiện hành có kích cỡ 14px thì 1 em = 14 px. Đây là một đơn vị rất hữu ích trong việc hiển thị trang web.

Đơn vị màu sắc

Đơn vị	Mô tả
Color-name	Tên màu. Ví dụ: black, white, red, green, blue, cyan, magenta,...
RGB (r,g,b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 255 kết hợp với nhau tạo ra vô số màu.
RGB (%r,%g,%b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 100% kết hợp.
Hexadecimal RGB	Mã màu RGB dạng hệ thập lục. Ví dụ: #FFFF: trắng, #000: đen,

1.2.3. Vị trí đặt CSS:

Chúng ta có ba cách khác nhau để nhúng CSS vào trong một tài liệu HTML

+ **Cách 1:** Nội tuyến (kiểu thuộc tính) Đây là một phương pháp nguyên thủy nhất để nhúng CSS vào một tài liệu HTML bằng cách nhúng vào từng thẻ HTML muốn áp dụng. Lưu ý: Nếu bạn muốn áp dụng nhiều thuộc tính cho nhiều thẻ HTML khác nhau thì không nên dùng cách này.

Ví dụ định nền màu đen cho trang và màu chữ trắng cho đoạn văn bản như sau:

```
<html>
<head> <title>Ví dụ</title> </head>
<body style="background-color:#000;">
  <p style="color:white">^ _ Welcome To MyWebsite ^ _</p>
</body>
</html>
```

+ **Cách 2:** Bên trong (thẻ style) bằng cách rút tất cả các thuộc tính CSS vào trong thẻ style (để tiện cho công tác bảo trì, sửa chữa về sau).

Ví dụ định nền màu trắng cho trang và màu chữ xanh lá cho đoạn văn bản như sau:

```
<html>
<head> <title>Ví dụ</title>
<style type="text/css">
  body { background-color:#000 }
  p { color:white }
</style>
</head>
<body>
  <p>^ _ Welcome To MyWebsite ^ _</p>
</body>
</html>
```

Lưu ý: Thẻ style nên đặt trong thẻ head.

+ **Cách 3:** Bên ngoài (liên kết với một file CSS bên ngoài) Thay vì đặt tất cả các mã CSS trong thẻ style chúng ta sẽ đưa chúng vào trong một file CSS (có phần mở rộng .css).

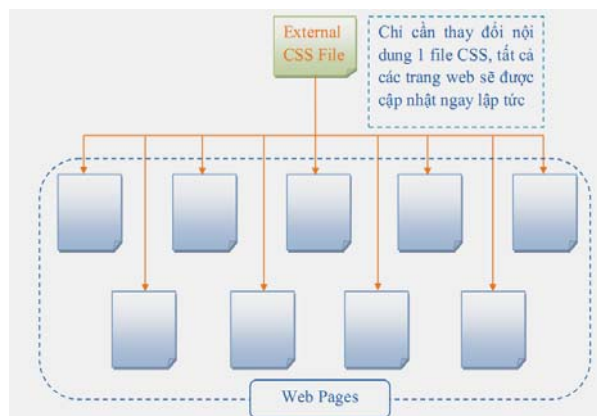
Đây là cách làm được khuyến cáo, đặc biệt hữu ích cho việc đồng bộ hay bảo trì một website lớn sử dụng cùng một kiểu mẫu.

Cú pháp để chèn file css vào trang là:

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

Hoặc

```
<style type="text/css"> @import url("filename.css") </style>
```



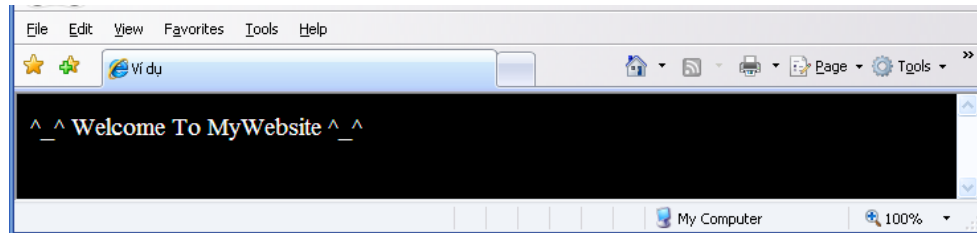
+ Đầu tiên chúng ta sẽ tạo ra một file html

```
<html>
<head> <title>Ví dụ</title>
<link rel="stylesheet" type="text/css" href="filename.css" />
</head>
```

```
<body>
  <p>^_^ Welcome To MyWebsite ^_^</p>
</body>
</html>
```

+ Sau đó hãy tạo một file style.css lưu cùng thư mục với file htm trên với nội dung:

```
body { background-color:#000 }
p{ color:White }
```



1.2.4. Sự ưu tiên:

Trước khi thực thi CSS cho một trang web. Trình duyệt sẽ đọc toàn bộ CSS mà trang web có thể được áp dụng, bao gồm: CSS mặc định của trình duyệt, file CSS bên ngoài liên kết vào trang web, CSS nhúng trong thẻ <style> và các CSS nội tuyến.

Sau đó, trình duyệt sẽ tổng hợp toàn bộ CSS này vào một CSS ảo, và nếu có các thuộc tính CSS giống nhau thì thuộc tính CSS nào nằm sau sẽ được ưu tiên sử dụng.

Theo nguyên tắc đó trình duyệt sẽ ưu tiên theo trình tự : Các CSS nội tuyến → CSS bên trong → CSS bên ngoài → CSS mặc định của trình duyệt.

2. CÁC THUỘC TÍNH ĐỊNH DẠNG

2.1. Định dạng nền trang

2.1.1. Màu nền (thuộc tính background-color)

Giá trị mã màu của background-color cũng giống như color. Ví dụ sử dụng thuộc tính background-color để định màu nền trang, các thành phần h1, h2 lần lượt là xanh lơ, đỏ và cam.

```
body { background-color:cyan }
h1 { background-color:red }
h2 { background-color:orange }
```

2.1.2. Ảnh nền (thuộc tính background-image)

Để chèn ảnh nền cho một thành phần trên trang web chúng ta sử dụng thuộc tính background-image. Ví dụ file background.css

```
body {
  background-image:url(logo.jpg)
}
h1 { background-color:red }
h2 { background-color:orange }
p { background-color: FDC689 }
```

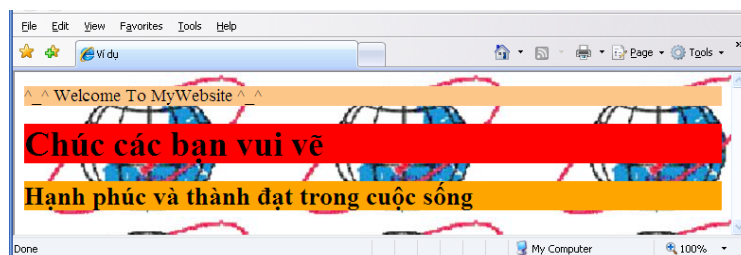
và file html

```
<html>
<head> <title>Ví dụ</title>
<link rel="stylesheet" type="text/css" href="background.css" />
```

```

</head>
<body>
<p>^_^ Welcome To MyWebsite ^_^</p>
<h1>Chúc các bạn vui vẻ</h1>
<h2>Hạnh phúc và thành đạt trong cuộc sống</h2>
</body>
</html>

```



2.1.3. Lặp lại ảnh nền (thuộc tính background-repeat):

Nếu sử dụng một ảnh có kích thước quá nhỏ để làm nền cho một đối tượng thì theo mặc định trình duyệt sẽ lặp lại ảnh nền để phủ kín không gian còn thừa. Thuộc tính background-repeat giúp điều khiển trình trạng lặp lại của ảnh nền. Thuộc tính này có 4 giá trị:

- + repeat-x: Chỉ lặp lại ảnh theo phương ngang.
- + repeat-y: Chỉ lặp lại ảnh theo phương dọc.
- + repeat: Lặp lại ảnh theo cả 2 phương, đây là giá trị mặc định.
- + no-repeat: Không lặp lại ảnh.

2.1.4. Khóa ảnh nền (thuộc tính background-attachment)

Background-attachment là một thuộc tính cho phép bạn xác định tính cố định của ảnh nền so với với nội dung trang web. Thuộc tính này có 2 giá trị:

- + scroll: Ảnh nền sẽ cuộn cùng nội dung trang web, đây là giá trị mặc định.
- + fixed: Cố định ảnh nền so với nội dung trang web, ảnh nền sẽ đứng yên khi bạn đang cuộn trang web (mở bất động).

2.1.5. Định vị ảnh nền (thuộc tính background-position)

Theo mặc định ảnh nền khi được chèn sẽ nằm ở góc trên, bên trái màn hình. Với thuộc tính background-position bạn sẽ có thể đặt ảnh nền ở bất cứ vị trí nào.

Ví dụ:

Giá trị	Ý nghĩa
Background-position:5cm 2cm	Ảnh được định vị 5cm từ trái qua và 2cm từ trên xuống.
Background-position:20% 30%	Ảnh được định vị 20% từ trái qua và 30% từ trên xuống.
Background-position:bottom left	Ảnh được định vị ở góc trái phía dưới

Thuộc tính background rút gọn

Khi sử dụng quá nhiều thuộc tính CSS sẽ gây khó khăn cho người đọc, công tác chỉnh sửa cũng như tốn nhiều dung lượng nên CSS đưa ra một cấu trúc rút gọn cho các thuộc tính cùng nhóm.

Ví dụ: Chúng ta có thể nhóm lại đoạn CSS sau

```

background-color:transparent;
background-image: url(logo.jpg);

```

```
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: right bottom;
```

thành một dòng ngắn gọn:

```
background:transparent url(logo.jpg) no-repeat fixed right bottom;
```

Từ ví dụ trên chúng ta có thể khái quát cấu trúc rút gọn cho nhóm background:

```
background:<background-color> | <background-image> | <background-repeat> |  
          <background-attachment> | <background-position>
```

2.2. Định dạng ký tự

2.2.1. Định dạng Font

a. Thuộc tính font-family:

Thuộc tính font-family có công dụng định nghĩa một danh sách ưu tiên các font sẽ được dùng để hiển thị một thành phần trang web.

Có hai loại tên font được dùng trong font-family: family-names và generic families.

+ Family-names: Tên cụ thể của một font. Ví dụ: Arial, Verdana, Tohama,...

+ Generic families: Tên của một họ gồm nhiều font. Ví dụ: sans-serif, serif,...

Ví dụ sau chúng ta sẽ viết CSS để quy định font chữ dùng cho cả trang web là Times New Roman, Tohama, sans-serif, và font chữ dùng để hiển thị các tiêu đề h1, h2, h3 sẽ là Arial, Verdana và các font họ serif.

```
body { font-family:"Times New Roman",Tohama,sans-serif }  
h1, h2, h3 { font-family:arial,verdana,serif }
```

Chú ý: Đối với các font có khoảng trắng trong tên như Times New Roman cần được đặt trong dấu ngoặc kép.

b. Thuộc tính font-style:

Thuộc tính font-style định nghĩa việc áp dụng các kiểu in thường (normal), in nghiêng (italic) hay xiên (oblique) lên các thành phần trang web. Ví dụ thực hiện áp dụng kiểu in nghiêng cho thành phần h1 và kiểu xiên cho h2.

```
h1 { font-style:italic; }  
h2 { font-style:oblique; }
```

c. Thuộc tính font-variant:

Thuộc tính font-variant được dùng để chọn giữa chế độ bình thường và small-caps của một font chữ. Một font small-caps là một font sử dụng chữ in hoa có kích cỡ nhỏ hơn in hoa chuẩn để thay thế những chữ in thường. Nếu như font chữ dùng để hiển thị không có sẵn font small-caps thì trình duyệt sẽ hiện chữ in hoa để thay thế. Trong ví dụ sau chúng ta sẽ sử dụng kiểu small-caps cho phần h1

```
h1 { font-variant:small-caps }
```

d. Thuộc tính font-weight:

Thuộc tính font-weight mô tả cách thức thể hiện của font chữ là ở dạng bình thường (normal) hay in đậm (bold). Thử in đậm phần p:

```
p { font-weight:bold }
```

e. Thuộc tính font-size:

Kích thước của một font được định bởi thuộc tính font-size. Thuộc tính này nhận các giá trị đơn vị đo hỗ trợ bởi CSS bên cạnh các giá trị xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger. Tùy theo mục đích sử dụng của website bạn có thể lựa chọn những đơn vị phù hợp.

Ở ví dụ trang web có kích cỡ font là 20px, h1 là 3 em = 3 x 20 = 60px, h2 là 2em = 40px.

```
body { font-size:20px }
h1 { font-size:3em }
h2 { font-size:2em }
```

Thuộc tính font rút gọn

Tương tự như các thuộc tính background, chúng ta cũng có thể rút gọn các thuộc tính font lại thành một thuộc tính đơn như ví dụ sau:

```
h1 { font-style: italic; font-variant:small-caps; font-weight: bold; font-size: 35px;
                                           font-family: arial,verdana,sans-serif; }
```

Thành: h1 { font: italic bold 35px arial,verdana,sans-serif; }

Cấu trúc rút gọn cho các thuộc tính nhóm font:

Font :<font-style> | <font-variant> | <font-weight> | <font-size> |<font-family>

2.2.2. Màu chữ (thuộc tính color)

Để định màu chữ cho một thành phần nào đó trên trang web chúng ta sử dụng thuộc tính color. Giá trị của thuộc tính này là các giá trị màu CSS hỗ trợ. Ví dụ sau chúng ta sẽ viết CSS để định màu chữ chung cho một trang web là đen, cho tiêu đề h1 màu xanh da trời, cho tiêu đề h2 màu xanh lá chúng ta sẽ làm như

```
body { color:#000 }
h1 { color:#0000FF }
h2 { color:#00FF00 }
```

2.2.3. Thuộc tính text-indent :

Thuộc tính text-indent cung cấp khả năng tạo ra khoảng thụt đầu dòng cho dòng đầu tiên trong đoạn văn bản. Giá trị thuộc tính này là các đơn vị đo cơ bản dùng trong CSS. Trong ví dụ sau chúng ta sẽ định dạng thụt đầu dòng một khoảng 30px cho dòng văn bản đầu tiên trong mỗi đoạn văn bản đối với các thành phần <p>

```
p { text-indent:30px }
```

2.2.4. Thuộc tính text-align :

Thuộc tính text-align giúp canh chỉnh văn bản cho các thành phần trong trang web. Thuộc tính này có tất cả 4 giá trị : left (canh trái – mặc định), right (canh phải), center (canh giữa) và justify (canh đều). Trong ví dụ sau chúng ta sẽ thực hiện canh phải các thành phần h1, h2 và canh đều đối với thành phần <p>

```
h1, h2 { text-align:right }
p { text-align:justify }
```

2.2.5. Thuộc tính letter-spacing:

Thuộc tính letter-spacing được dùng để định khoảng cách giữa các ký tự trong một đoạn văn bản. Muốn định khoảng cách giữa các ký tự trong thành phần h1, h2 là 7px và thành phần <p> là 5px chúng ta sẽ viết CSS sau:

```
h1, h2 { letter-spacing:7px }
p { letter-spacing:5px }
```

2.2.6. Thuộc tính text-decoration:

Thuộc tính text-decoration giúp bạn thêm các hiệu ứng gạch chân (underline), gạch xiên (line-through), gạch đầu (overline), và một hiệu ứng đặc biệt là văn bản nhấp nháy (blink). Ví dụ sau chúng ta sẽ định dạng gạch chân cho thành phần h1, gạch đầu thành phần h2

```
h1 { text-decoration:underline }
h2 { text-decoration:overline }
```

2.2.7. Thuộc tính text-transform:

Text-transform là thuộc tính qui định chế độ in hoa hay in thường của văn bản mà không phụ thuộc vào văn bản gốc trên HTML. Thuộc tính này có tất cả 4 giá trị: uppercase (in hoa), lowercase (in thường), capitalize (in hoa ở ký tự đầu tiên trong mỗi từ) và none (không áp dụng hiệu ứng – mặc định). Trong ví dụ dưới đây chúng ta sẽ định dạng cho thành phần h1 là in hoa, h2 là in hoa đầu mỗi ký tự.

```
h1 { text-transform:uppercase }
h2 { text-transform:capitalize }
```

2.3. Định dạng liên kết

Một thành phần rất quan trọng trong mọi website chính là liên kết. Cũng như một đối tượng văn bản thông thường, chúng ta hoàn toàn có thể áp dụng các thuộc tính định dạng văn bản thông thường cho một liên kết. Hơn nữa, CSS còn cung cấp một điều khiển đặc biệt được gọi là pseudo-classes.

Pseudo-classes cho phép bạn xác định các hiệu ứng định dạng cho một đối tượng liên kết ở một trạng thái xác định như khi liên kết chưa được thăm (a:link), khi rê chuột lên liên kết (a:hover), khi liên kết được thăm (a:visited) hay khi liên kết đang được kích hoạt – đang giữ nhấn chuột (a:active).

Với điều khiển pseudo-classes cùng với các thuộc tính CSS đã học chắc chắn sẽ mang lại rất nhiều ý tưởng về trang trí liên kết cho trang web. Sau đây chúng ta sẽ tiến hành một số ví dụ để tìm hiểu thêm về các khả năng trang trí cho một liên kết dựa trên pseudo-classes.

Ví dụ chúng ta sẽ áp dụng 4 màu sắc khác nhau cho từng trạng thái liên kết: các liên kết chưa thăm có màu xanh lá; các liên kết mouse over sẽ có màu đỏ tươi; các liên kết đã thăm sẽ có màu đỏ và các liên kết đang kích hoạt có màu tím.

```
a:link { color:#00FF00 }
a:hover { color:#FF00FF }
a:visited { color:#FF0000 }
a:active { color:# 662D91 }
```

Ví dụ Tạo các hiệu ứng tương ứng với trình trạng liên kết: các liên chưa thăm có màu xanh lá, kích cỡ font 14px; liên kết mouse over có màu đỏ tươi, kích cỡ font 1.2em, hiệu ứng nhấp nháy; liên kết đã thăm sẽ có màu xanh da trời, không có đường gạch chân; các liên kết đang kích hoạt có màu tím và font dạng small-caps.

```
a:link { color:#00FF00; font-size:14px }
a:hover { color:#FF00FF; font-size:1.2em; text-decoration:blink }
a:visited { color:#FF0000; text-decoration:none }
a:active { color:# 662D91; font-variant:small-caps }
```

Ví dụ cũng tạo cho liên kết hiệu ứng màu sắc giống ví dụ trên nhưng sẽ có thêm 1 số hiệu ứng: các liên kết sẽ có khung viền màu đen, kích cỡ font 14px; liên kết mouse over có nền light cyan; các liên kết đã thăm có nền light yellow.

```
a { border:1px solid #000; font-size:14px }
a:link { color:#00FF00; }
a:hover{background-color:#00BFF3;color:#FF00FF;font-size:1.2em;text-decoration:blink }
```



```
a:visited { background-color:#FFF568; color:#FF0000; text-decoration:none }
a:active { color:#662D91; font-variant:small-caps }
```

Áp dụng các ví dụ trên vào file html sau

```
<html>
<head> <title>Ví dụ</title>
      <link rel="stylesheet" type="text/css" href="link.css" />
</head>
<body>
      <a href=index.htm>^_^ Welcome To MyWebsite ^_^</a>
</body>
</html>
```



2.4. Nhóm các phần tử- Class & ID

Chúng ta đã sử dụng các thuộc tính CSS về background, color, font,... Tuy nhiên, bạn cũng nhận ra là khi áp dụng một thuộc tính CSS cho một thành phần nào đó ví dụ như h1, h2, p, a, img,... thì toàn bộ các thành phần này trong trang web đều nhận thuộc tính này. Vậy có cách nào để nhóm lại một số thành phần nào đó để áp dụng một thuộc tính đặc biệt. Ví dụ như bạn muốn các liên kết trên menu trang web sẽ được in hoa, và có kích cỡ lớn hơn so với liên kết trong nội dung đây chính là vấn đề mà chúng ta sẽ giải quyết như sau:

2.4.1. Nhóm các phần tử với class :

Ví dụ chúng ta có một đoạn mã HTML sau đây :

```
<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
  <li>Hà Nội</li>
  <li>TP. Hồ Chí Minh</li>
  <li>Đà Nẵng</li>
  <li>Thừa Thiên Huế</li>
  <li>Khánh Hòa</li>
  <li>Quảng Ninh</li>
  <li>Tiền Giang</li>
</ul>
```

Yêu cầu đặt ra là làm thế nào để tên các thành phố là màu đỏ và tên các tỉnh là màu xanh da trời. Để giải quyết vấn đề này chúng ta sẽ dùng một thuộc tính HTML gọi là class để tạo thành 2 nhóm là thành phố và tỉnh.

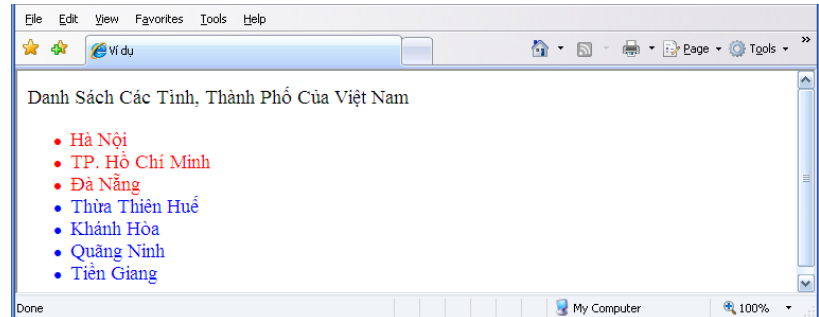
Ta sẽ viết lại đoạn HTML sau thành như thế này:

```
<html>
<head> <title>Ví dụ</title>
      <link rel="stylesheet" type="text/css" href="class.css" />
</head>
<body>
  <p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
  <ul>
    <li class="tp">Hà Nội</li>
    <li class="tp">TP. Hồ Chí Minh</li>
    <li class="tp">Đà Nẵng</li>
```

```
<li class="tinh">Thừa Thiên Huế</li>
<li class="tinh">Khánh Hòa</li>
<li class="tinh">Quảng Ninh</li>
<li class="tinh">Tiền Giang</li>
</ul>
</body>
</html>
```

Với việc dùng class để nhóm các đối tượng như trên thì công việc của chúng ta sẽ trở nên đơn giản hơn nhiều, file class.css sẽ là

```
li.tp { color:FF0000 }
li.tinh { color:0000FF }
```



2.4.2. Nhận dạng phần tử với id:

Cũng với đoạn HTML như ví dụ về class. Nhưng yêu cầu đặt ra là Hà Nội sẽ có màu đỏ sậm, TP. Hồ Chí Minh màu đỏ, Đà Nẵng màu đỏ tươi còn các tỉnh màu xanh da trời.

Để giải quyết vấn đề này chúng ta sẽ sử dụng thuộc tính HTML là id để nhận dạng mỗi thành phố và dùng class để nhóm các tỉnh. Đoạn HTML của chúng ta bây giờ sẽ là :

```
<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
  <li id="hanoi">Hà Nội</li>
  <li id="hcmc">TP. Hồ Chí Minh</li>
  <li id="danang">Đà Nẵng</li>
  <li class="tinh">Thừa Thiên Huế</li>
  <li class="tinh">Khánh Hòa</li>
  <li class="tinh">Quảng Ninh</li>
  <li class="tinh">Tiền Giang</li>
</ul>
```

Và đoạn CSS cần dùng sẽ là :

```
#hanoi { color:# 790000 }
#hcmc { color:#FF0000 }
#danang { color:#FF00FF }
.tinh { color:#0000FF }
```

2.4.3. Nhóm phần tử với thẻ

Thẻ trong HTML là một thẻ trung hòa, nó không thêm hay bớt bất cứ một thứ gì vào một tài liệu HTML. Nhưng chính nhờ tính chất trung hòa này mà nó lại là một công cụ đánh dấu tuyệt vời để qua đó chúng ta có thể viết CSS định dạng cho các phần tử mong muốn.

Ví dụ: Chúng ta có đoạn HTML sau trích dẫn câu nói của chủ tịch Hồ Chí Minh

```
<p>Không có gì quý hơn độc lập, tự do.</p>
```

Yêu cầu ở đây là chúng ta hãy dùng CSS tô đậm 2 từ độc lập, tự do. Để giải quyết vấn đề này, chúng ta sẽ thêm thẻ `<span>` vào đoạn HTML như sau:

```
<p>Không có gì quý hơn <span class="nhanmanh">độc lập</span>,<br>
<span class="nhanmanh">tự do</span>.</p>
```

Và bây giờ chúng ta có thể viết CSS cho yêu cầu trên:

```
.nhanmanh { font-weight:bold }
```

2.4.4. Nhóm khối phần tử với thẻ `<div>`

Cũng như `<span>`, `<div>` cũng là một thẻ trung hòa và được thêm vào tài liệu HTML với mục đích nhóm các phần tử lại cho mục đích định dạng bằng CSS. Tuy nhiên, điểm khác biệt là `<span>` dùng để nhóm một khối phần tử trong khi đó `<div>` có thể nhóm một hoặc nhiều khối phần tử.

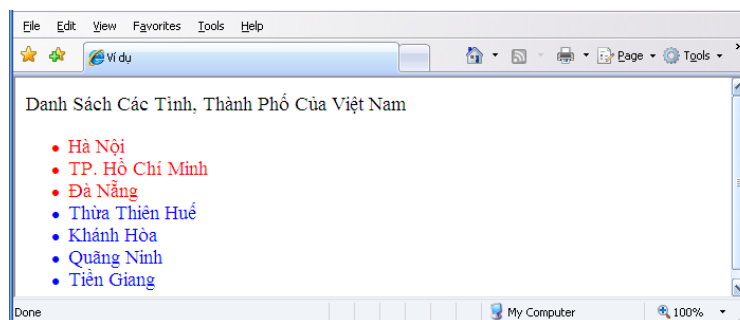
Trở lại ví dụ về danh sách tỉnh, thành trong phần class bài trước chúng ta sẽ giải quyết vấn đề bằng cách nhóm các phần tử với `<div>` như sau:

```
<html>
<head> <title>Ví dụ</title>
<link rel="stylesheet" type="text/css" href="div.css" />
</head>
<body>
<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul> <div id="tp">
<li>Hà Nội</li>
<li>TP. Hồ Chí Minh</li>
<li>Đà Nẵng</li>
</div>
<div id="tinh">
<li>Thừa Thiên Huế</li>
<li>Khánh Hòa</li>
<li>Quảng Ninh</li>
<li>Tiền Giang</li>
</div>
</ul>
</body>
</html>
```

Và đoạn CSS cho mục đích này sẽ là:

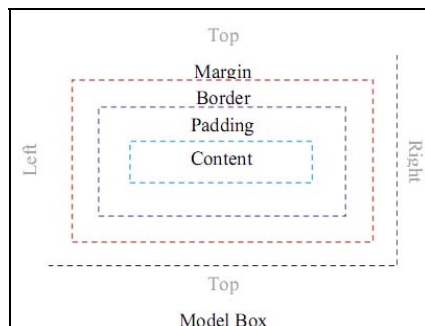
```
#tp { color:#FF0000 }
```

```
#tinh { color:0000FF }
```



2.5. Box Model

Trong CSS, box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần. Nó bao gồm padding (vùng đệm), border (viền) và margin (canh lề) và các tùy chọn. Hình bên dưới mô tả cấu trúc minh họa mô hình hộp cho một thành phần web.



Mô hình hộp trên chỉ là một mô hình lý thuyết lý tưởng. Bên dưới đây chúng ta sẽ xét mô hình hộp của một đối tượng web cụ thể:

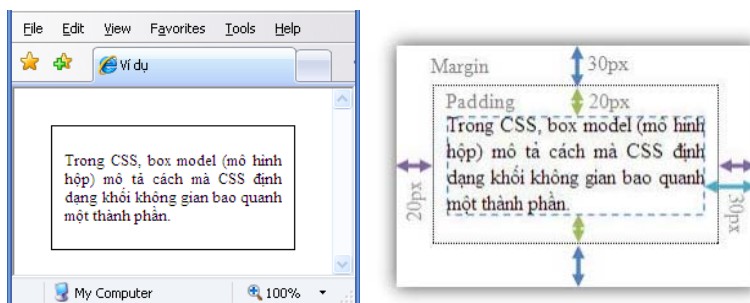
Ví dụ: Chúng ta có một đoạn HTML sau

<p> Trong CSS, box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần. </p>

Phần CSS cho đoạn HTML trên:

```
p{width:200px; margin:30px 20px; padding:20px 10px; border:1px solid #000; text-align:justify }
```

Với ví dụ trên chúng ta sẽ khái quát được mô hình hộp như sau:



2.6. Margin & Padding

2.6.1. Thuộc tính margin:

Thuộc tính margin trong CSS cũng được dùng để canh lề cho cả trang web hay một thành phần web này với các thành phần web khác hay với viền trang. Ví dụ sau sẽ chỉ cho chúng ta biết cách canh lề cho một trang web.

```
body { margin-top:80px; margin-bottom:40px; margin-left:50px; margin-right:30px;
border:1px dotted #FF0000 }
```

Hoặc gọn hơn chúng ta sẽ viết như sau:

```
body { margin:80px 30px 40px 50px; border:1px dotted #FF0000 }
```

Cú pháp như sau:

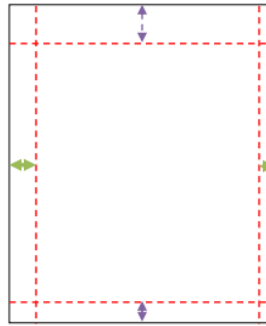
```
margin:<margin-top> | <margin-right> | <margin-bottom> | <margin-left>
```

Hoặc:

```
margin:<value1>|< value2>
```

với value 1 là giá trị margin-top và margin-bottom và value2 là giá trị margin-left và margin-right.

Kết quả của ví dụ trên sẽ được mô hình hóa như sau:



Ví dụ kế tiếp sẽ thể hiện rõ hơn về việc dùng margin để canh lề cho các đối tượng trong trang web. Các bạn hãy quan sát các đường viền và nhận xét.

```
body { margin:80px 30px 40px 50px; border:1px solid #FF0000 }
#box1 { margin:50px 30px 20px 40px; border:1px solid #00FF00 }
#box2 { margin:50px 30px 20px 40px; border:1px solid #0000FF }
```

2.6.2. Thuộc tính padding

Padding có thể hiểu như là một thuộc tính đệm. Padding không ảnh hưởng tới khoảng cách giữa các đối tượng như margin mà nó chỉ quy định khoảng cách giữa phần nội dung và viền của một đối tượng (xem lại ảnh minh họa về boxmodel).

Cú pháp: Tương tự margin.

Padding:<padding-top> | <padding-right> | <padding-bottom> | <padding-left>

Thực hiện lại ví dụ nhưng thay margin bằng padding, và quan sát vị trí đường viền.

2.7. Khung viền - Border

Border là một thành phần quan trọng trong một trang web. Nó thường được dùng trong trang trí, đóng khung cho một đối tượng cần nhấn mạnh, phân cách các đối tượng giúp trang web trông dễ nhìn hơn,....

2.7.1. Thuộc tính border-width

Border-width là một thuộc tính CSS quy định độ rộng cho viền của một đối tượng web. Thuộc tính này có các giá trị: thin (mảnh), medium (vừa), thick (dày), hay là một giá trị đo cụ thể như pixels.

2.7.2. Thuộc tính border-color

Border-color là thuộc tính CSS quy định màu viền cho một đối tượng web. Thuộc tính này nhận tất cả đơn vị màu CSS hỗ trợ.

2.7.3. Thuộc tính border-style

Border-style là thuộc tính CSS quy định kiểu viền thể hiện của một đối tượng web. CSS cung cấp tất cả 8 kiểu viền tương ứng với 8 giá trị: dotted, dashed, solid, double, groove, ridge, inset và outset. Để hiểu rõ hơn, chúng ta hãy xem hình minh họa dưới đây.



Ví dụ sau sẽ định viền cho 3 thành phần h1, h2, p như sau:

```
h1 { border-width:thin; border-color:#FF0000; border-style:solid }
h2 { border-width:thick; border-color:#CCC; border-style:dotted }
p { border-width:5px; border-color:#FF00FF; border-style:double }
```

Ngoài ra, chúng ta cũng có thể dùng riêng các thuộc tính `border-top`, `border-right`, `border-bottom` hay `border-left` để chỉ định viền riêng cho các đối tượng.

Ví dụ:

```
h1 {  
  border-top-width:thin; border-top-color:#FF0000; border-top-style:solid;  
  border-right-width:thick; border-right-color:#AFAFAF; border-right-style:dotted;  
  border-bottom-width:2px; border-bottom-color:blue; border-bottom-style:double;  
  border-left-width:5px; border-left-color:violet; border-left-style:groove }
```

Thuộc tính `border` rút gọn sử dụng cấu trúc rút gọn của CSS như sau:

```
Border:<border-width> |<border-color> |<border-style>
```

2.8. Height & Width

2.8.1. Thuộc tính `width`:

`Width` là một thuộc tính CSS dùng để quy định chiều rộng cho một thành phần web. Ví dụ sau chúng ta sẽ định chiều rộng cho thành phần `p` của một trang web.

```
p { width:700px; }
```

2.8.2. Thuộc tính `max-width`:

`Max-width` là thuộc tính CSS dùng để quy định chiều rộng tối đa cho một thành phần web.

2.8.3. Thuộc tính `min-width`:

`Min-width` là thuộc tính CSS dùng để quy định chiều rộng tối thiểu cho 1 thành phần web.

2.8.4. Thuộc tính `height`:

`Height` là một thuộc tính CSS dùng để quy định chiều cao cho một thành phần web.

Ví dụ sau chúng ta sẽ định chiều cao cho thành phần `p` của một trang web.

```
p { height:300px }
```

2.8.5. Thuộc tính `max-height`:

`Max-height` là thuộc tính CSS dùng để quy định chiều cao tối đa cho một thành phần web.

2.8.6. Thuộc tính `min-height`:

`Min-height` là thuộc tính CSS dùng để quy định chiều cao tối thiểu cho 1 thành phần web.

Lưu ý: Thông thường chiều cao một thành phần web do văn bản trong trang web đó quyết định. Việc định chiều cao chính xác cho một thành phần sẽ tạo thanh cuộn văn bản nếu chiều cao văn bản lớn hơn chiều cao đã định. Các thuộc tính `max/min-width/height` được sử dụng trong những trường hợp bạn không chắc giá trị chính xác cho `width`, `height` cả một thành phần.

Ví dụ, bạn vùng chứa bài post của một forum có bề ngang 500px, bạn có thể định `max-width:500px` cho phần hình ảnh trong phần đó để tránh những ảnh lớn bị lệch ra ngoài.

3 CÁC THUỘC TÍNH KHÁC

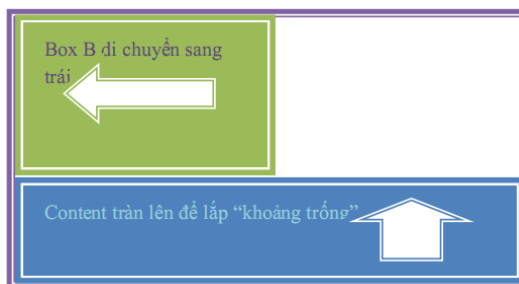
3.1. Float & Clear

3.1.1. Thuộc tính `float`:

`Float` (theo như nghĩa tiếng Việt là thả trôi) là một thuộc tính CSS dùng để cố định một thành phần web về bên trái hay bên phải không gian bao quanh nó. Đây là một thuộc tính rất cần thiết khi dàn trang (như tạo các trang web 2 column layout hay 3 column layout), hiển thị văn bản thành cột (giống như kiểu

Format > Columns trong MS Word vậy), hay thực hiện việc định vị trí ảnh và text (như một số kiểu text wrapping của MS Word).

Để rõ hơn chúng ta hãy xem hình minh họa sau về cơ chế hoạt động của float:



Thuộc tính float có 3 giá trị:

- + Left: Cố định phần tử về bên trái.
- + Right: Cố định phần tử về bên phải.
- + None: Bình thường.

Nhìn vào hình minh họa trên chúng ta thấy là ban đầu trong box lớn có hai thành phần là Box B và phần Content.

Lúc đầu Box B nằm bên trên và Content nằm bên dưới, nhưng khi chúng ta đặt thuộc tính float cho Box B thì Box B bị cố định về bên trái và chừa lại khoảng trống bên trái nó. Còn phần Content thì vốn nằm bên dưới sẽ tự động tràn lên để lấp đầy khoảng trống do Box B tạo ra.

Ở ví dụ sau, chúng ta sẽ thực hiện float ảnh logo sang trái để phần nội dung bên dưới tràn lên nằm cạnh logo.

```
#logo { float:left; }
```

Một ví dụ khác là chúng ta sẽ thử dùng float để chia 2 cột văn bản.

```
.column1, .column2 { width:45%; float:left; text-align:justify; padding:0 20px; }
.column1 { border-right:1px solid #000 }
```

3.1.2. Thuộc tính clear:

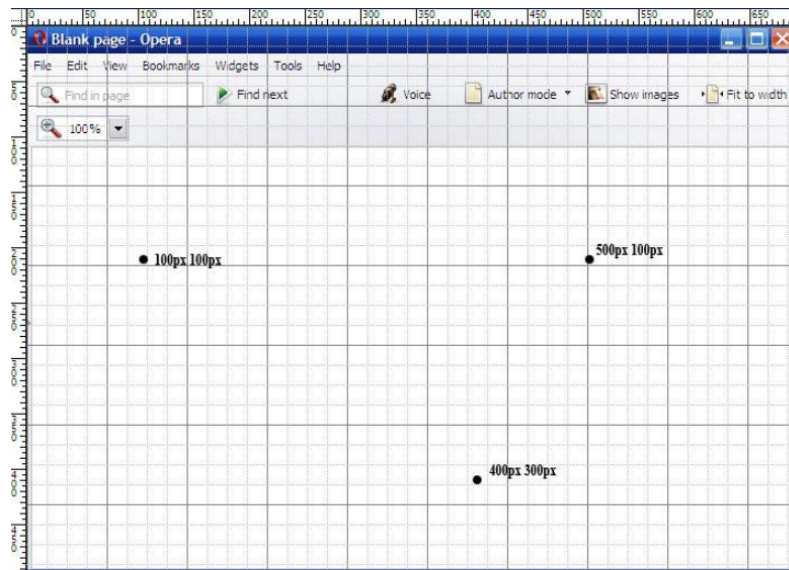
Thuộc tính clear là một thuộc tính thường được gán vào các phần tử liên quan tới phần tử đã được float để quyết định hướng xử sự của phần tử này.

Ở ví dụ trên, khi chúng ta float tấm ảnh qua trái thì mặc nhiên văn bản sẽ được tràn lên để lấp vào chỗ trống. Nhưng khi chúng ta đặt vào văn bản thuộc tính clear thì chúng ta có quyền quyết định xem phần văn bản đó có được tràn lên hay không. Thuộc tính clear có tất cả 4 thuộc tính: left (tràn bên trái), right (tràn bên phải), both (không tràn) và none.

3.2. Position

3.2.1. Nguyên lý hoạt động của position

Hãy tưởng tượng cửa sổ trình duyệt của bạn giống như một hệ tọa độ và với position bạn có thể đặt một đối tượng web ở bất cứ vị trí nào trên hệ tọa độ này.



Giả sử chúng ta muốn định vị một ảnh ở vị trí 70px cách đỉnh và 90px từ bên trái tài liệu, chúng ta sẽ viết CSS như sau:

```
img { position:absolute; top:70px; left:90px }
```

Như bạn đã thấy, sự định vị bằng CSS là một công nghệ chính xác để định vị một thành phần. Nó dễ dàng hơn so với việc dùng bảng.

3.2.2. Absolute position:

Định vị tuyệt đối là sự định vị mà trong đó các thành phần được định vị không để lại bất cứ một khoảng trống nào trong tài liệu. Một thành phần được định vị tuyệt đối sẽ nhận giá trị position là absolute. Các đối tượng đã định vị tuyệt đối sẽ dùng kết hợp với các thuộc tính top, left, right, bottom để xác định tọa độ. Ví dụ sau sẽ chỉ cho chúng ta cách đặt bốn ảnh ở bốn góc tài liệu bằng định vị tuyệt đối.

```
#logo1 { position:absolute; top:50px; left:70px }
#logo2 { position:absolute; top:0; right:0 }
#logo3 { position:absolute; bottom:0; left:0 }
#logo4 { position:absolute; bottom:70px; right:50px }
```

3.2.3. Relative position

Sự định vị tương đối cho một thành phần là sự định vị được tính từ vị trí gốc trong tài liệu. Các thành phần đã được định vị tương đối sẽ để lại khoảng không trong tài liệu. Các thành phần được định vị tương đối sẽ nhận giá trị position là relative.

Chúng ta hãy làm lại ví dụ trên nhưng thay absolute thành relative. Các bạn ghi nhận lại vị trí 4 ảnh logo lúc áp dụng thuộc tính position là none, absolute và relative rồi rút ra nhận xét.

3.3. Layers

Layer là thuộc tính giúp ta đặt một thành phần này lên trên một thành phần khác. Với mục đích này, bạn sẽ gán cho mỗi phần tử một con số. Theo đó, phần tử có số cao hơn sẽ nằm trên, phần tử có số thấp hơn sẽ nằm dưới. Ví dụ sau chúng ta sẽ đặt 5 bức ảnh logo ở 5 lớp.

```
#logo1 { position:absolute; top:70px; left:50px; z-index:1 }
#logo2 { position:absolute; top:140px; left:100px; z-index:2 }
#logo3 { position:absolute; top:210px; left:150px; z-index:3 }
#logo4 { position:absolute; top:280px; left:200px; z-index:4 }
#logo5 { position:absolute; top:350px; left:250px; z-index:5 }
```


Với trang html là:

```
<html>
<head>
<title>Ví Dụ Về Layers</title>
<link rel="stylesheet" type="text/css" href="layer.css" />
</head>
<body>





</body>
</html>
```

3.4. Web Standards

Trong công việc thiết kế web thì một trong những vấn đề quan trọng đó là làm sao đảm bảo trang web của bạn có thể hiển thị tốt trên hầu hết các trình duyệt. Hiểu rõ vấn đề này W3C (World Wide Web Consortium) – một tổ chức có nhiệm vụ quản lý và đặc ra các tiêu chuẩn web cùng với các đối tác khác như Microsoft, Mozilla Foundation,... đã đặt ra các tiêu chuẩn về mã cho web. Nó cho phép nhà phát triển web có thể tự tin hơn khi thực hiện dự án cũng như đảm bảo cho các trang web thỏa mãn tiêu chuẩn có thể hiển thị tốt ở nhiều trình duyệt. Để dễ dàng hơn. Trong CSS, W3C đã tạo ra một công cụ gọi là CSS Validator để đọc và thẩm định tính hợp chuẩn cho CSS của bạn.

Đầu tiên, các bạn truy cập vào địa chỉ sau: <http://jigsaw.w3.org/css-validator/>. Đặt url file CSS của bạn ở ô url rồi nhấn nút “click to check stylesheet” để chương trình đọc file CSS của bạn. Sau khi đọc xong, nếu file CSS của bạn không phù hợp tiêu chuẩn, chương trình sẽ hiển thị danh sách lỗi. Nếu file CSS của bạn hợp chuẩn thì chương trình sẽ hiện ra bức ảnh chứng nhận. Bạn có thể đặt bức ảnh đó trên trang web của bạn để thể hiện nó đã được xây dựng trên các mã chuẩn.