



Simple CSS

Standard Edition

2008



Simple CSS

Standard Edition

By WallPearl

©2008 WallPearl'sBlog. All Rights Reserved.

Mục Lục

Nội dung	Trang
Lời mở đầu	6
Bài 1: Giới thiệu	8
1.1. CSS là gì?.....	8
1.2. Tại sao CSS?	8
1.3. Học CSS cần những gì?	9
Bài 2: Một số quy ước về cách viết CSS.....	10
2.1. Cú pháp CSS	10
2.2. Đơn vị CSS	13
2.3. Vị trí đặt CSS	14
2.4. Sự ưu tiên	18
Bài 3: Màu chữ và màu nền	21
3.1. Thuộc tính background-color.....	21
3.2. Thuộc tính background-image	21
3.3. Thuộc tính background-repeat	22
3.4. Thuộc tính background-attachment	23
3.5. Thuộc tính background-position	23
Bài 4: Font chữ	26
4.1. Thuộc tính font-family	26
4.2. Thuộc tính font-style.....	27

4.3. Thuộc tính font-variant	27
4.4. Thuộc tính font-weight	28
4.5. Thuộc tính font-size	28
Bài 5: Text	30
5.1. Thuộc tính color	30
5.2. Thuộc tính text-indent.....	30
5.3. Thuộc tính text-align.....	31
5.4. Thuộc tính letter-spacing	31
5.5. Thuộc tính text-decoration.....	32
5.6. Thuộc tính text-transform	32
Bài 6: Pseudo-classes for Links	33
Bài 7: Class & id	36
7.1. Nhóm phần tử với class.....	36
7.2. Nhận dạng phần tử với id.....	38
Bài 8: Span & div	40
8.1. Nhóm phần tử với 	40
8.2. Nhóm phần tử với <div>.....	40
Bài 9: Box Model	43
Bài 10: Margin & padding	45
10.1. Thuộc tính margin.....	45
10.2. Thuộc tính padding	47

Bài 11: Border	48
11.1. Thuộc tính border-width	48
11.2. Thuộc tính border-color	48
11.3. Thuộc tính border-style	48
Bài 12: Height & width	50
12.1. Thuộc tính width	50
12.2. Thuộc tính max-width	50
12.3. Thuộc tính min-width	50
12.4. Thuộc tính height	50
12.5. Thuộc tính max-height	51
12.6. Thuộc tính min-height	51
Bài 13: Float & clear	52
13.1. Thuộc tính float	52
13.2. Thuộc tính clear	53
Bài 14: Position	54
14.1. Absolute position	55
14.2. Relative position	56
Bài 15: Layers	57
Bài 16: Web standard	59
Phụ lục	60

Lời Mở Đầu

Đầu tiên, Pearl xin thưa với tất cả các bạn đọc giả (hay những bạn nào có quyển sách này) đó là Pearl không phải là một chuyên gia về CSS mà chỉ là một “tay ngang trong nghề” mà thôi (cụ thể là Pearl cũng chỉ lên mạng học từ các tài liệu tiếng Anh chứ cũng không phải qua trường lớp đào tạo bài bản gì cả). Còn về quyển sách này thì phải nói từ hồi đầu tháng 6 năm rồi (rối chuyện thế đấy, nhưng ai quen Pearl đều biết Pearl có tính hay “tường trình” vậy mà (smile)), Pearl vào blog [anh Tân](#) thấy anh có dịch mấy bài tut CSS từ [HTML.net](#) mà trong bài mở đầu lại có ghi tên các “cao thủ” CSS trên Opera: [chị Hiền](#), [chị Liên](#), [Phạm Lâm](#), và cả Pearl (còn ai nữa không thì quên rồi, tại anh giấu mấy bài này rồi). Đúng là rối chuyện thật, chả nhẽ lại đi khiêu nại cái chuyện “Pearl chỉ là tay ngang thôi” sao. Thôi thì có người bảo mình là “cao thủ” cũng cứ chịu vậy, nhưng ngặt cái “có tiếng phải có miếng” chứ, các “cao thủ” kia thì điều có viết tut, tip hướng dẫn làm blog, sửa skin cả. Trong khi đó, mình lại chả viết gì cho cộng đồng thì đâu có được (không phải Pearl lười, tại mấy người kia viết hết rồi đó chứ, với lại Pearl cũng không thích viết tut, tip như họ, trông cứ như công thức nấu ăn vậy (big green)), nên Pearl mới nhào vô xin dịch mấy bài coi như là mình đóng góp tí đỉnh cho cộng đồng (ặc, cũng là có ý dụ dỗ các “cao thủ” kia vào chia phần thôi, cứ bao nhiêu “cao thủ” như thế này thì có thêm chục bài nữa cũng 1 tuần là xong (roll eyes)... anh mừng quá, cho dịch 2 bài cuối (chuối, người ta đang muốn làm ngay mà, bảo dịch bài cuối chẳng phải dụ dỗ Pearl lười biếng đó sao, mà kệ)... Tới cuối tháng 6, anh về quê nghỉ hè, về được rồi thì trốn biệt luôn, Pearl cũng quên mất luôn... tới tháng 8 mới thấy anh lên blog: không rảnh, lâu lâu về quên, cần thời gian với gia đình (ai đọc blog anh này cũng biết “người yêu vợ” rồi)... đã thế Pearl dịch tất cho xong... tới trung tuần tháng 8, bắt đầu dịch,... người học dở Anh văn mà dịch mau ghê, dịch hơn 10 ngày là hết 16 tut luôn... định tháng 9, biên tập lại rồi phát hành... Định là thế, nhưng tới tháng 9, Pearl cài lại máy 1 phát, rồi lo lung sục, nhồi nhét phần mềm, sách, nhạc vào lại ổ cứng, tốn những hai tuần,... quên tập 2, mấy bài tut đó đành xếp xó tiếp... Mãi tới tháng rồi, cụ thể là sau Giáng Sinh 3 ngày, Pearl lại lôi 16 bài dịch đó ra biên tập lại (hem phải tại siêng đột xuất, mà tại sắp hết năm với lại đây cũng là hoạt động kỷ niệm ngày thành lập WallPearl's Blog), biên tới, biên lui tốn cả tuần mới xong. Đọc lại mới thấy, 16 tut của HTML dot net còn thiếu nhiều cái lắm. Thế là lại lấy thêm tài liệu từ [W3 School](#) để bổ

khuyết, rồi lại muốn lấy thêm nhiều tài liệu khác thêm vô, thêm cả CSS3 vô, rồi lại trông nó hỗn hợp quá nên sau cùng Pearl phân ra 2 bản:

- Bản Simple CSS Standard Edition: Bao gồm nội dung chủ yếu từ 16 tut của HTML.Net, chỉ bổ sung 1 số chỗ về cú pháp CSS và độ ưu tiên của CSS.
- Bản Simple CSS Advanced Edition: Bao gồm nội dung của HTML.Net trộn với W3 School có thêm phần phụ lục về CSS3 và nhiều thứ cần thiết khác.

Phần “tường trình” tới đây là kết thúc. Chắc chắn là ít có quyển sách nào lại có phần tường trình như quyển sách này (chỉ ít thì không có quyển sách in nào lại viết vụ này). Điều này chẳng qua là Pearl dở văn, không viết uốn éo, éo lả được, chỉ biết viết theo những gì mình nghĩ, theo những gì mình biết thôi (smile)

Do đây là lần đầu tiên Pearl thực hiện một quyển sách hướng dẫn về một đề tài tin học, hơn nữa do khả năng kiến thức, kiến văn hạn hẹp nên chắc chắn quyển sách này vẫn còn rất nhiều thiếu sót, nên Pearl rất mong nhận được ý kiến đánh giá của mọi người.

Sau cùng, Pearl xin chân thành gửi lời cảm ơn tới HTML.Net, W3 School, [CSS3.Info](#) và nhiều trang web khác đã cung cấp tài liệu để Pearl hoàn thành quyển sách này. Cảm ơn tất cả các bạn blogger đã ủng hộ, cổ vũ Pearl trong thời gian vừa qua.

Mọi ý kiến đóng góp có thể post trực tiếp trên các blog của Pearl hoặc qua địa chỉ email wallpearl@gmail.com hoặc wallpearl@inbox.com

Tháng 1, năm 2008

∴WallPearl∴

Bài 1: Giới Thiệu



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. Đây chỉ là một định nghĩa giàu hình ảnh của Pearl thôi (nhưng cũng thực tế nhỉ (smile)). Còn CSS (Cascading Style Sheets mà Pearl tạm dịch là tờ mẫu theo Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML,...

1.2. Tại sao CSS?

Nếu bạn đã từng học qua HTML thì cũng biết HTML cũng hỗ trợ một 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 cao trong kết hợp các thuộc tính giúp mang lại hiệu quả. Ngoài ra, hiện tại 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 hầu như “như nhau” dù trên một hệ thống sử dụng Windows, Linux hay trên một máy Mac miễn là bạn đang sử dụng một phiên bản trình duyệt mới nhất.

Sử dụng các mã định dạng trực tiếp trong HTML tốn hao nhiều thời gian thiết kế cũng như dung lượng lưu trữ trên đĩa cứng. Trong khi đó CSS đưa ra phương thức “tờ mẫu ngoại” giúp áp dụng một khuôn mẫu chuẩn từ một file CSS ở ngoài. Nó thật sự 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. Hãy thử tưởng tượng bạn có một website với hàng trăm trang và bạn muốn thay đổi font chữ hay màu chữ cho một thành phần nào đó. Đó thật sự sẽ là một công việc buồn chán và tốn nhiều thời gian. Nhưng với việc sử dụng CSS việc đó là hoàn toàn đơn giản cũng như là bạn có một trò ma thuật nào đó.

Ngoài ra, CSS còn cho phép bạn áp đặt những kiểu trình bày thích hợp hơn cho các phương tiện khác nhau như màn hình máy tính, máy in, điện thoại,...

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

1.3. Học CSS cần những gì?

Thật sự không có một điều kiện gì được quy định khi học CSS. Nhưng ở một khía cạnh nào đó thì một sự chuẩn bị cho một cuộc hành trình dù là dễ nhất vẫn không thừa vì ít nhất nó sẽ giúp bạn làm tốt hơn.

Hành trang thứ nhất mà bạn nên có là một kiến thức về HTML, nó không thật sự cần thiết nếu bạn chỉ dùng CSS để trình bày cho một trang HTML có sẵn (như làm skin cho blog chẳng hạn), nhưng bạn vẫn cần biết ý nghĩa một số thẻ HTML, nó sẽ có ích khi bạn viết CSS. Tuy nhiên, nếu bạn muốn tự thiết kế, trình bày một trang web của riêng mình thì tùy theo quy mô trang web, bạn cần phải học thêm cả HTML, XHTML, Javascript và một số ngôn ngữ lập trình web khác.

Hành trang thứ hai chính là một trình soạn thảo văn bản để bạn có thể viết mã CSS. Ở đây, Pearl khuyên bạn nên sử dụng một trình soạn thảo đơn giản như Notepad, Wordpad trong Windows hay Pico trong Linux, Simple Text trong Mac. Nó sẽ giúp bạn chắc là code là của bạn và không có bất kỳ một sự can thiệp nào từ chương trình như khi dùng DreamWeaver, FrontPage, Golive,...

Hành trang thứ ba của bạn chính là một phiên bản mới nhất của trình duyệt mà bạn thường dùng.

Và một điều nữa mà Pearl muốn đề nghị đó là các bạn hãy dành một ít thời gian để thực hành CSS sau mỗi bài học nó sẽ thật sự có ích cho bạn. Thực hành chẳng những giúp bạn vận dụng nhuần nhuyễn các bài học mà còn có tác dụng explain ngược lại những lý thuyết mà bạn chưa hiểu.

Bây giờ nếu bạn đã thật sự chuẩn bị chúng ta hãy chuyển qua chương tiếp theo để thật sự bước chân vào thế giới CSS.

Bài 2: Một Số Quy Ước Về Cách Viết CSS



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) chúng ta dùng code sau:

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

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

Nhìn qua ví dụ trên ít nhiều chúng ta cũng thấy được mối tương đồng giữa các thuộc tính trong HTML và CSS cho nên nếu bạn đã học qua HTML thì cũng sẽ rất dễ dàng tiếp thu CSS. Đó là một chút lợi thế của câu chuyện hành trình mà Pearl đã nói ở bài trước. Nhưng không sao cả, bây giờ hãy nhìn vào ví dụ của chúng ta và các bạn xem nó có giống với cấu trúc sau không nhé.

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. Nó là các tag HTML, **class** hay **id** (chúng ta sẽ học về 2 thành phần này ở bài học sau). Ví dụ: `body`, `h2`, `p`, `img`, `#title`, `#content`, `.username`,...

Trong CSS ngoài viết tên selector theo tên tag, class, id. Chúng ta còn có thể viết tên selector theo phân cấp như để chỉ các ảnh ở trong `#entry`, chúng ta viết selector là `#entry img`, như vậy thì các thuộc tính chỉ định sẽ chỉ áp dụng riêng cho các ảnh nằm trong `#entry`.

Khi viết tên cho class, đôi khi sẽ có nhiều thành phần có cùng class đó, ví dụ như thẻ `img` và thẻ `a` cùng có class tên `vistors` nhưng đây lại là hai đối tượng khác nhau, 1 cái là ảnh của người thăm, 1 cái là liên kết tới trang người thăm. Nên nếu

khi viết CSS ta ghi là `.visitors { width:50 }` thì sẽ ảnh hưởng tới cả hai thành phần. Nên trong trường hợp này, nếu bạn có ý dùng CSS đó chỉ riêng phần ảnh thì chỉ nên ghi là `img .visitors` thôi.

Một lỗi viết tên selector nữa đó là dựa trên tên các thuộc tính có trong HTML. Ví dụ trong HTML ta có đoạn mã như vậy: `<input name="Search" type="Text" value="Key Word">`. Để áp dụng thuộc tính CSS cho riêng ô tìm kiếm này chúng ta sẽ dùng selector `input[name="Search"]`.

Ngoài việc viết tên selector cụ thể, chúng ta cũng có thể dùng một selector đại diện như `* { color:red }` sẽ tác động đến tất cả các thành phần có trên trang web làm cho chúng có text màu đỏ.

+ *Property*: Chính là các thuộc tính quy định cách trình bày. Ví dụ: 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.

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

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. Ví dụ: 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 các 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 của bạn 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 để dễ dàng tìm, sửa chữa trong những lần cập nhật sau. Chú thích trong CSS được viết như sau /* Nội dung chú thích */

Ví dụ:

```
/* Màu chữ cho trang web */
```

```
body {
    color:red
}
```

2.2. Đơn vị CSS:

Trong CSS2 hỗ trợ các loại đơn vị là đơn vị đo chiều dài và đơn vị đo góc, thời gian, cường độ âm thanh và màu sắc. Tuy nhiên, sử dụng phổ biến nhất vẫn là đơn vị đo chiều dài và màu sắc. Sau đây là bảng liệt kê các đơn vị chiều dài và màu sắc dùng trong CSS.

Đơn vị chiều dài

Đơn vị	Mô tả	Đơn vị	Mô tả
%	Phần trăm	ex	1 ex bằng chiều cao của chữ x in thường của font hiện hành. Do đó, đơn vị này không những phụ thuộc trên kích cỡ font chữ mà còn phụ thuộc loại font chữ vì cùng 1 cỡ 14px nhưng chiều cao chữ x của font Times và font Tohama là khác nhau.
in	Inch (1 inch = 2.54 cm)		
cm	Centimeter		
mm	Millimeter		
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.	pt	Point (1 pt = 1/72 inch)
		pc	Pica (1 pc = 12 pt)
		px	Pixels (điểm ảnh trên màn hình máy tính)

Đơn vị màu sắc

Đơn vị	Mô tả
Color-name	Tên màu tiếng Anh. 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: #FFFFFF: trắng, #000000: đen, #FF00FF: đỏ tươi.

2.3. Vị trí đặt CSS:

Ở trên chúng ta đã tìm hiểu về cú pháp viết CSS, nhưng còn đặt nó ở đâu trong tài liệu HTML? Trong phần này, Pearl xin giới thiệu với các bạn về vấn đề này.

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. Và dĩ nhiên trong trường hợp này chúng ta sẽ không cần selector trong cú pháp.

Lưu ý: Nếu bạn muốn áp dụng nhiều thuộc tính cho nhiều thẻ HTML khác nhau thì không nên dùng cách này.

Ở ví dụ sau chúng ta sẽ tiến hành định nền màu trắng cho trang và màu chữ xanh lá cho đoạn văn bản như sau:

```
<html>
<head>
<title>Ví dụ</title>
</head>
<body style="background-color=#FFF;">
<p style="color:green">^_^ Welcome To WallPearl's Blog ^_^</p>
</body>
</html>
```

+ Cách 2: Bên trong (thẻ style)

Thật ra nếu nhìn kỹ chúng ta cũng nhận ra đây chỉ là một phương cách thay thế cách thứ nhất bằng cách rút tất cả các thuộc tính CSS vào trong thẻ style (để tiện cho công tác bảo trì, sửa chữa ấy mà).

Cũng ví dụ làm trang web có màu nền trắng, đoạn văn bản chữ xanh lá, chúng ta sẽ thể hiện như sau:

```
<html>

<head>

<title>Ví dụ</title>

<style type="text/css">

body { background-color:#FFF }
p { color:#00FF00 }

</style>

</head>

<body>

<p>^ _ ^ Welcome To WallPearl's Blog ^ _ ^</p>

</body>

</html>
```

Lưu ý: Thẻ style nên đặt trong thẻ head.

Đối với những trình duyệt cũ, không thể nhận ra thẻ <style>. Theo mặc định, thì khi một trình duyệt không nhận ra một thẻ thì nó sẽ hiện ra phần nội dung chứa trong thẻ. Như ở ví dụ trên, nếu trình duyệt không hỗ trợ thẻ style thì 2 dòng CSS: body {background-color:#FFF } p { color:#00FF00 } sẽ hiện ra trên trình duyệt. Để tránh tình trạng này, bạn nên đưa vào thêm dấu <!-- ở trước và --> ở sau khối code CSS. Như ví dụ trên sẽ viết lại là:

```
<style type="text/css">

<!-- body { background-color:#FFF }

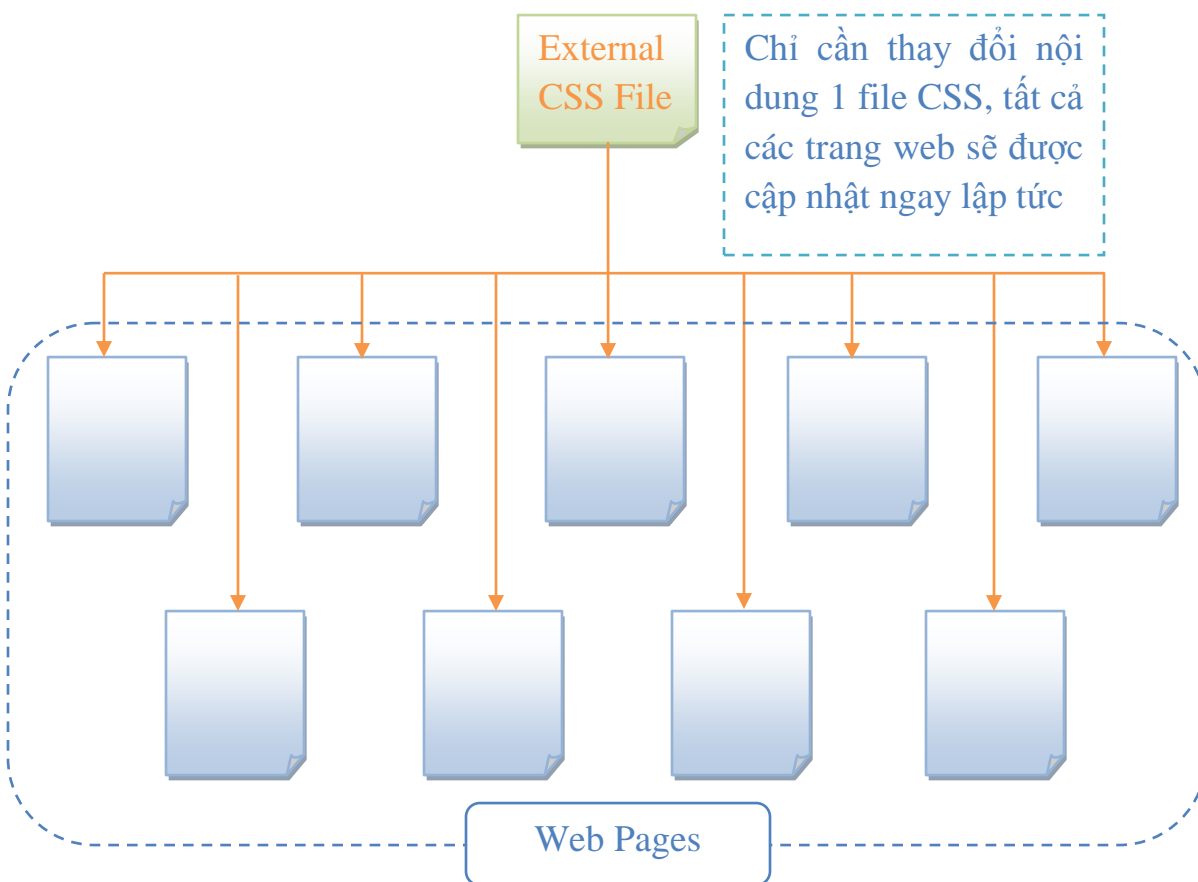
      p { color:#00FF00 } -->

</style>
```

+ Cách 3: Bên ngoài (liên kết với một file CSS bên ngoài)

Tương tự như cách 2 nhưng 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) bên ngoài và liên kết nó vào trang web bằng thuộc tính href trong thẻ link.

Đây là cách làm được khuyến cáo, nó đặc biệt hữu ích cho việc đồng bộ hay bảo trì một website lớn sử dụng cùng một kiểu mẫu. Các ví dụ trong sách này cũng được trình bày theo kiểu này.



Nào bây giờ chúng ta hãy mở Notepad lên và thử thực hiện theo ví dụ sau:

Đầu tiên chúng ta sẽ tạo ra một file vidu.html có nội dung như sau:


```
<html>

<head>

<title>Ví dụ</title>

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

</head>

<body>

<p>^ _ ^ Welcome To WallPearl's Blog ^ _ ^</p>

</body>

</html>
```

Sau đó hãy tạo một file style.css với nội dung:

```
body {
    background-color:#FFF
}

p {
    color:#00FF00
}
```

Hãy đặt 2 file này vào cùng một thư mục, mở file vidu.html trong trình duyệt của bạn và xem thành quả.

Lưu ý:

Để lưu 1 file với 1 đuôi khác .txt trong Notepad chúng ta chọn Save as type là All Files. Có thể chọn Encoding là UTF-8, nếu bạn chú thích CSS bằng tiếng Việt.

Trong CSS chúng ta còn có thể sử dụng thuộc tính @import để nhập một file CSS vào CSS hiện hành. Cú pháp: @import url(link)

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 (cái này cũng giống như chương trình “Ai Là Triệu Phú” trên truyền hình vậy, chỉ câu trả lời sau cùng mới được chấp nhận (smile)). Theo nguyên tắc đó trình duyệt của bạn sẽ ưu tiên cho các CSS nội tuyến > CSS bên trong > CSS bên ngoài > CSS mặc định của trình duyệt.

Ví dụ:

Trong một trang web có liên kết tới file style.css có nội dung như sau:

```
p {
    color:#333;
    text-align:left;
    width:500px
}
```

trong thẻ <style> giữa thẻ <head> cũng có một đoạn CSS liên quan:

```
p {
    background-color:#FF00FF;
    text-align:right;
    width:100%;
    height:150px
}
```

trong phần nội dung trang web đó cũng có sử dụng CSS nội tuyến:

```
<p style="height:200px; text-align:center; border:1px solid #FF0000; color:#000" >
```

Khi thực thi CSS trình duyệt sẽ đọc hết tất cả các nguồn chứa style rồi sẽ tổng hợp lại vào một CSS ảo và nếu có sự trùng lặp các thuộc tính CSS thì nó sẽ lấy thuộc

tính CSS có mức ưu tiên cao hơn. Như ví dụ trên chúng ta sẽ thấy CSS cuối cùng mà phân tử p nhận được là:

```
p {  
    background-color:#FF00FF;  
    width:100%;  
    height:200px;  
    text-align:center;  
    border:1px solid #FF0000;  
    color:#000  
}
```

Vậy có cách nào để thay đổi độ ưu tiên cho một thuộc tính nào đó? Thật ra thì trong CSS đã có sẵn một thuộc tính giúp chúng ta thực hiện điều này, đó chính là thuộc tính !important. Chỉ cần bạn đặt thuộc tính này sau một thuộc tính nào đó theo cú pháp selector { property:value !important } thì trình duyệt sẽ hiểu đây là một thuộc tính được ưu tiên. Bây giờ, chúng ta cùng xét lại ví dụ trên nhưng có đặt thêm một số thuộc tính !important vào xem kết quả như thế nào nhé.

```
p {  
    width:500px;  
    text-align:left !important;  
    color:#333 !important  
}  
p {  
    background-color:#FF00FF;  
    width:100%;  
    height:150px !important;  
    text-align:right;  
}
```

```
<p style="text-align:center; height:200px; border:1px solid #FF0000; color:#000" >
```

Phần CSS sẽ tác động lên thuộc tính p là:

```
p {  
    background-color:#FF0000;  
    width:100%;  
    height:150px !important;  
    text-align:left !important;  
    border:1px solid #FF0000;  
    color:#333 !important  
}
```

Lưu ý: Cùng một thuộc tính cho một selector thì nếu cả hai thuộc tính đều đặt !important thì cái sau được lấy.

Bài 3: Background



Trong bài học này chúng ta sẽ được học về cách định màu nền/ảnh nền cho một trang web cũng như các kỹ thuật định vị, điều chỉnh ảnh nền.

3.1. Màu nền (thuộc tính background-color):

Thuộc tính background-color giúp định màu nền cho một thành phần trên trang web. Các giá trị mã màu của background-color cũng giống như color nhưng có thêm giá trị transparent để tạo nền trong suốt.

Ví dụ sau đây sẽ chỉ cho chúng ta biết cách sử dụng thuộc tính background-color để định màu nền cho cả trang web, 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  
}
```

3.2. Ảnh nền (thuộc tính background-image):

Việc sử dụng ảnh nền giúp trang web trông sinh động và bắt mắt hơn. Để chèn ảnh nền vào một thành phần trên trang web chúng ta sử dụng thuộc tính background-image.



Bây giờ chúng ta sẽ cùng làm một ví dụ minh họa để xem thuộc tính background-image sẽ hoạt động ra sao. Đầu tiên hãy tìm một tấm ảnh mà bạn thích, ở đây Pearl sẽ lấy tấm ảnh logo của blog Pearl

Sau đó, chúng ta sẽ viết CSS để đặt logo này làm ảnh nền trang web như sau:

```
body {
    background-image:url(logo.png)
}
```

```
h1 {
    background-color:red
}
h2 {
    background-color:orange
}
p {
    background-color: FDC689
}
```

Như các bạn đã thấy chúng ta sẽ phải chỉ định đường dẫn của ảnh trong cặp ngoặc đơn sau url. Do ảnh đặt trong cùng thư mục với file style3.css nên chúng ta chỉ cần ghi abc.jpg. Nhưng nếu chúng ta tạo thêm một thư mục img trong thư mục thì chúng ta sẽ phải ghi là background-image:url(img/abc.jpg). Đôi khi nếu không chắc lắm bạn có thể dùng đường dẫn tuyệt đối cho ảnh.

3.3. 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 lớn hơn thì theo mặc định trình duyệt sẽ lặp lại ảnh nền để phủ kín không gian còn thừa. Thuộc tính background-repeat cung cấp cho chúng ta các điều khiển giúp kiểm soát 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.

Bây giờ, chúng ta hãy thêm thuộc tính background-repeat này vào ví dụ trên thử xem sao.

```
body {
    background-image:url(logo.png);
    background-repeat:no-repeat;
}
```

Các bạn xem, có phải ảnh nền đã không bị lặp lại như trong ví dụ trước, hãy thử thay đổi qua lại giữa các giá trị và xem kết quả tạo ra.

3.4. 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 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. Khi áp dụng giá trị này, ảnh nền sẽ đứng yên khi bạn đang cuộn trang web.

3.5. 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. Tuy nhiên với thuộc tính background-position bạn sẽ có thể đặt ảnh nền ở bất cứ vị trí nào (trong không gian của thành phần mà nó làm nền).

Background-position sẽ dùng một cặp 2 giá trị để biểu diễn tọa độ đặt ảnh nền. Có khá nhiều kiểu giá trị cho thuộc tính position. Như đơn vị chính xác như centimeters, pixels, inches,... hay các đơn vị qui đổi như %, hoặc các vị trí đặt biệt như top, bottom, left, right.

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 ổ cứng cho 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.png);
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
```

thành một dòng ngắn gọn:

```
background:transparent url(logo.png) 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>
```

Theo mặc định thì các thuộc tính không được đề cập sẽ nhận các giá trị mặc định.

Ví dụ: Chúng ta sẽ bỏ qua hai thuộc tính background-attachment và background-position ở dòng mã trên đi:

```
background:transparent url(logo.png) no-repeat;
```




Hai thuộc tính không được chỉ định sẽ đơn thuần được thiết lập tới giá trị mặc định mà chúng ta điều biết là scroll và top left.

Bài 4: Font Chữ



Ở bài học trước, chúng ta đã được tìm hiểu về cách định dạng màu chữ bằng CSS. Trong bài này, chúng ta sẽ tiếp tục tìm hiểu thêm về các thuộc tính CSS liên quan tới font chữ của các thành phần trong một trang web.

4.1. 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. Theo đó, thì font đầu tiên được liệt kê trong danh sách sẽ được dùng để hiển thị trang web. Nếu như trên máy tính truy cập chưa cài đặt font này thì font thứ hai trong danh sách sẽ được ưu tiên...cho đến khi có một font phù hợp.

Có hai loại tên font được dùng để chỉ định 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,...

Khi lên danh sách font dùng để hiển thị một trang web bạn sẽ chọn những font mong muốn trang web sẽ được hiển thị để đặt ở các vị trí ưu tiên. Tuy nhiên, có thể những font này sẽ không thông dụng lắm nên bạn cũng cần chỉ định thêm một số font thông dụng dự phần như Arial, Tohama hay Times New Roman và bạn cũng được đề nghị đặt vào danh sách font của mình một generic families (thường thì nó sẽ có độ ưu tiên thấp nhất). Thực hiện theo cách này thì sẽ đảm bảo trang web của bạn có thể hiển thị tốt trên bất kỳ hệ thống nào.

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 }
```

Mở trang web trong trình duyệt và kiểm tra kết quả. Chúng ta thấy phần tiêu đề sẽ được ưu tiên hiển thị bằng font Arial, nếu trên máy không có font này thì font Verdana sẽ được ưu tiên và kế đó sẽ là các font thuộc họ serif.

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

4.2. 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. Trong ví dụ bên dưới chúng ta sẽ thử thực hiện áp dụng kiểu in nghiêng cho thành phần h1 và kiểu xiên cho h2.

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

4.3. 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  
}
```

4.4. 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). Ngoài ra, một số trình duyệt cũng hỗ trợ mô tả độ in đậm bằng các con số từ 100 – 900.

Thử in đậm phần p:

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

4.5. 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. Ví dụ trang web của bạn phục vụ chủ yếu là những người già, thị lực kém hay những người dùng sử dụng các màn hình máy tính kém chất lượng thì bạn có thể cân nhắc sử dụng các đơn vị qui đổi như em hay %. Như vậy sẽ đảm bảo font chữ trên trang web của bạn luôn ở kích thước phù hợp.

Ở ví dụ sau trang web sẽ có kích cỡ font là 20px, h1 là $3em = 3 \times 20 = 60px$, 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>

Đi qua bài học này, bạn đã nắm bắt được cách định font chữ cho một thành phần trang web cũng như cách sử dụng các kiểu font in nghiêng, in đậm, font small-caps và cách qui định kích thước font. Trong bài học kế chúng ta sẽ được tìm hiểu thêm về các thuộc tính CSS về định dạng văn bản.

Bài 5: Text



Định dạng và thêm vào các kiểu định dạng đặc biệt cho phần nội dung một trang web là một vấn đề quan trọng cho bất cứ nhà thiết kế web nào. Như đã nói, ở bài học này chúng ta sẽ được tìm hiểu về các thuộc tính CSS về định dạng văn bản.

5.1. 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  
}
```

5.2. 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  
}
```

5.3. Thuộc tính text-align :

Thuộc tính text-align giúp bạn thêm các canh chỉnh văn bản cho các thành phần trong trang web.

Cũng tương tự như các lựa chọn canh chỉnh văn bản trong các trình soạn thảo văn bản thông dụng như MS Word, 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  
}
```

5.4. 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 }
```

5.5. 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  
}
```

5.6. 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  
}
```


Bài 6: Pseudo-classes For Links



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 đã học ở 2 bài trước như định font chữ, gạch chân, màu chữ,... 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 1: Ví dụ này 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 2: 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 3: Ví dụ này cũng tạo cho liên kết hiệu ứng màu sắc giống ví dụ 2 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 }
```

Ba ví dụ trên chỉ là một tí gợi ý về khả năng kết hợp các thuộc tính CSS với pseudo-classes để tạo nên nhiều hiệu ứng hấp dẫn cho trang web. Cũng xin nói luôn là các ví dụ Pearl trình bày trong bài này cũng như những bài khác thật ra trông không dễ nhìn, đó là do Pearl không có nhiều thời gian để chăm chút các ví dụ của mình. Cái mà Pearl muốn nói chỉ là làm thế nào các bạn hiểu tác dụng một thuộc tính nào đó để có thể vận dụng cho trang web của chính mình.

Bài 7 : Class & ID



Trong các bài học trước, chúng ta đã được học 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 các 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 thì phải làm thế nào? Đây chính là vấn đề mà chúng ta sẽ cùng giải quyết trong chương này.

7.1. 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:

```
<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>
```

Với việc dùng class để nhóm các đối tượng như trên thì công việc của chúng ta sẽ trở nên đơn giản hơn nhiều:

```
li .tp {  
    color:FF0000  
}
```

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

Lưu ý: Không nên đặt tên class với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

7.2. Nhận dạng phần tử với id:

Ví dụ:

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 }
```

Lưu ý: Không nên đặt tên id với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

Trả qua hai ví dụ trên chúng ta có thể rút ra những kết luận sau:

- Class dùng để nhóm các đối tượng có cùng thuộc tính, do tính chất đó nó có thể được sử dụng nhiều lần.
- Id dùng để nhận dạng một đối tượng đặc trưng, id có tính duy nhất.

Trong bài học này, chúng ta đã được học về các sử dụng class và id để áp dụng các đặc tính đặc biệt cho một thành phần web. Ở bài kế tiếp chúng ta sẽ được học thêm về hai thẻ `<div>` và `` trong HTML và ý nghĩa 2 thẻ này đối với việc viết CSS.

Bài 8: Span & Div



Như đã hứa, ở bài học này chúng ta sẽ cùng tìm hiểu về 2 thẻ `` và `<div>` trong HTML và xem chúng có lợi ích gì cho công việc viết CSS của chúng ta.

8.1. Nhóm phần tử với thẻ ``:

Thẻ `` trong HTML thật ra là một thẻ trung hòa, nó không thêm hay bớt bất cứ một thứ gì vào một tài liệu HTML cả. 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 `` vào đoạn HTML như sau:

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

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

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

Rất đơn giản phải không nào.

8.2. Nhóm khối phần tử với thẻ `<div>`:

Cũng như ``, `<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 `` 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:

<p>Danh Sách Các Tỉnh, Thành Phố Việt Nam:</p>

<div id="tp">

Hà Nội

TP. Hồ Chí Minh

Đà Nẵng

</div>

<div id="tinh">

Thừa Thiên Huế

Khánh Hòa

Quảng Ninh

Tiền Giang

</div>

Và đoạn CSS cho mục đích này sẽ là:

```
#tp {  
    color:#FF0000  
}  
#tinh {  
    color:0000FF  
}
```

Trong hai bài học trên, chúng ta đã được học qua về id, class, <div>, và lợi ích của nó trong việc nhóm, đánh dấu phần tử để có thể dùng CSS để tạo ra các kiểu trình bày đặc biệt. Có thể một số bạn chưa học qua HTML sẽ khó khăn trong việc nắm bắt các kiến thức này. Tuy nhiên, yêu cầu mà Pearl đưa ra ở đây chỉ là các bạn có thể vận dụng các phần tử này trong CSS.

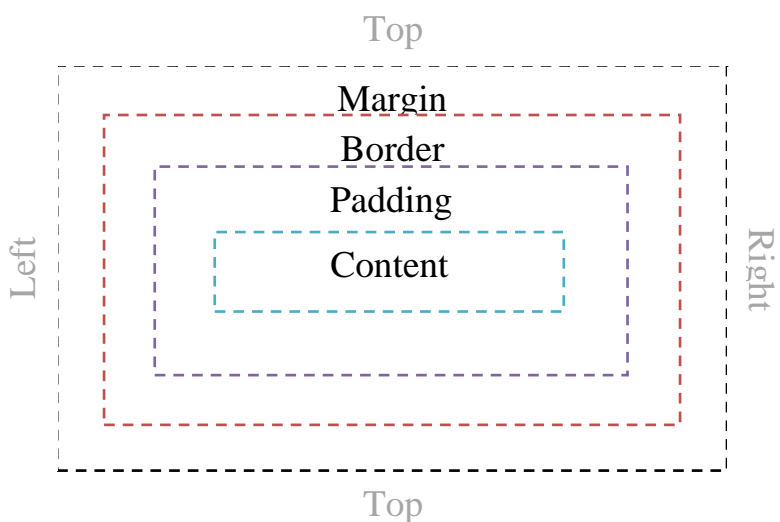
Bài 9: Box Model



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>

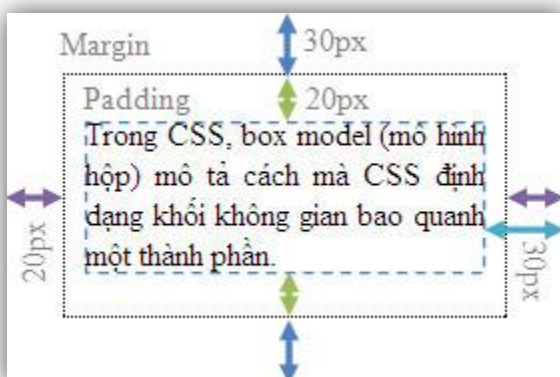
Model Box

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:



Có lẽ bạn sẽ cảm thấy hơi khó hiểu. Nhưng không sao cả, tạm thời bạn cứ chấp nhận như vậy. Còn về những “cái hộp” này thì chúng ta sẽ cùng tìm hiểu cụ thể hơn trong các bài học tiếp theo.

Bài 10: Margin & Padding



Như đã giới thiệu ở bài trước, trong bài học này chúng ta sẽ cùng tìm hiểu chi tiết về 2 thành phần margin và padding của mô hình hộp.

10.1. Thuộc tính margin:

Như tất cả những ai đã học qua MS Word đều biết là trong phần thiết lập Page Setup của Word cũng có một thiết lập margin để định lề cho trang in. Tương tự, 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 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:

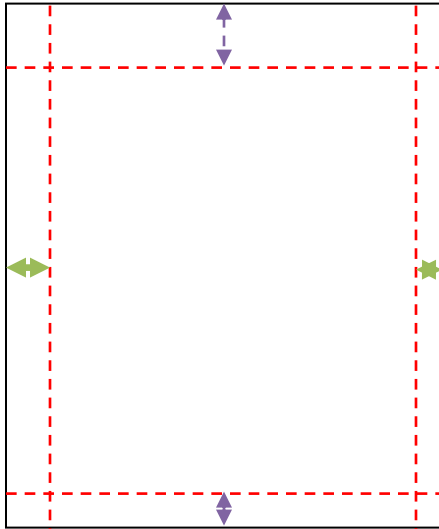
```
body {  
    margin:80px 30px 40px 50px;  
    border:1px dotted #FF0000  
}
```

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
}
```

10.2. 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>

Các bạn hãy thử thực hiện lại 2 ví dụ ở phần margin, nhưng thay margin bằng padding, nhớ quan sát vị trí đường viền.

Bài 11: 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,... Trong bài học này, chúng ta sẽ cùng tìm hiểu về cách dùng CSS để định border cho các đối tượng web.

11.1. Thuộc tính border-width:

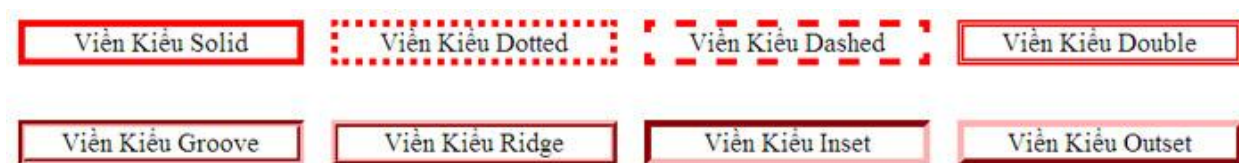
Border-width là một thuộc tính CSS quy định độ rộng cho viền của một đối tượng web. Thuộc tính này có các giá trị: thin (mảnh), medium (vừa), thick (dày), hay là một giá trị đo cụ thể như pixels. Xem hình minh họa bên dưới.

11.2. 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.

11.3. 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. Ngoài ra, hai giá trị none hay hidden dùng để ẩn đường viền. Để hiểu rõ hơn, chúng ta hãy xem hình minh họa dưới đây (tất cả viền có màu vàng).



Ví dụ: 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:**

Để dễ nhớ chúng ta có thể sử dụng cấu trúc rút gọn của CSS như sau:

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

Bài 12: Height & Width



Trong bài học này, chúng ta sẽ tìm hiểu thêm về cách dùng thuộc tính height và width để định kích cỡ cho một thành phần web.

12.1. 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;  
}
```

12.2. Thuộc tính max-width:

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

12.3. Thuộc tính min-width:

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

12.4. 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  
}
```

12.5. Thuộc tính max-height:

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

12.6. Thuộc tính min-height:

Min-height là thuộc tính CSS dùng để quy định chiều cao tối thiểu cho một 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.

Bài 13: Float & Clear



13.1. 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  
}
```

13.2. Thuộc tính clear:

Đi cùng với thuộc tính float, trong CSS còn có một thuộc tính là 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.

Thử dùng thuộc tính clear với các giá trị khác nhau đối với đoạn văn bản trong ví dụ float ảnh logo ở trên.

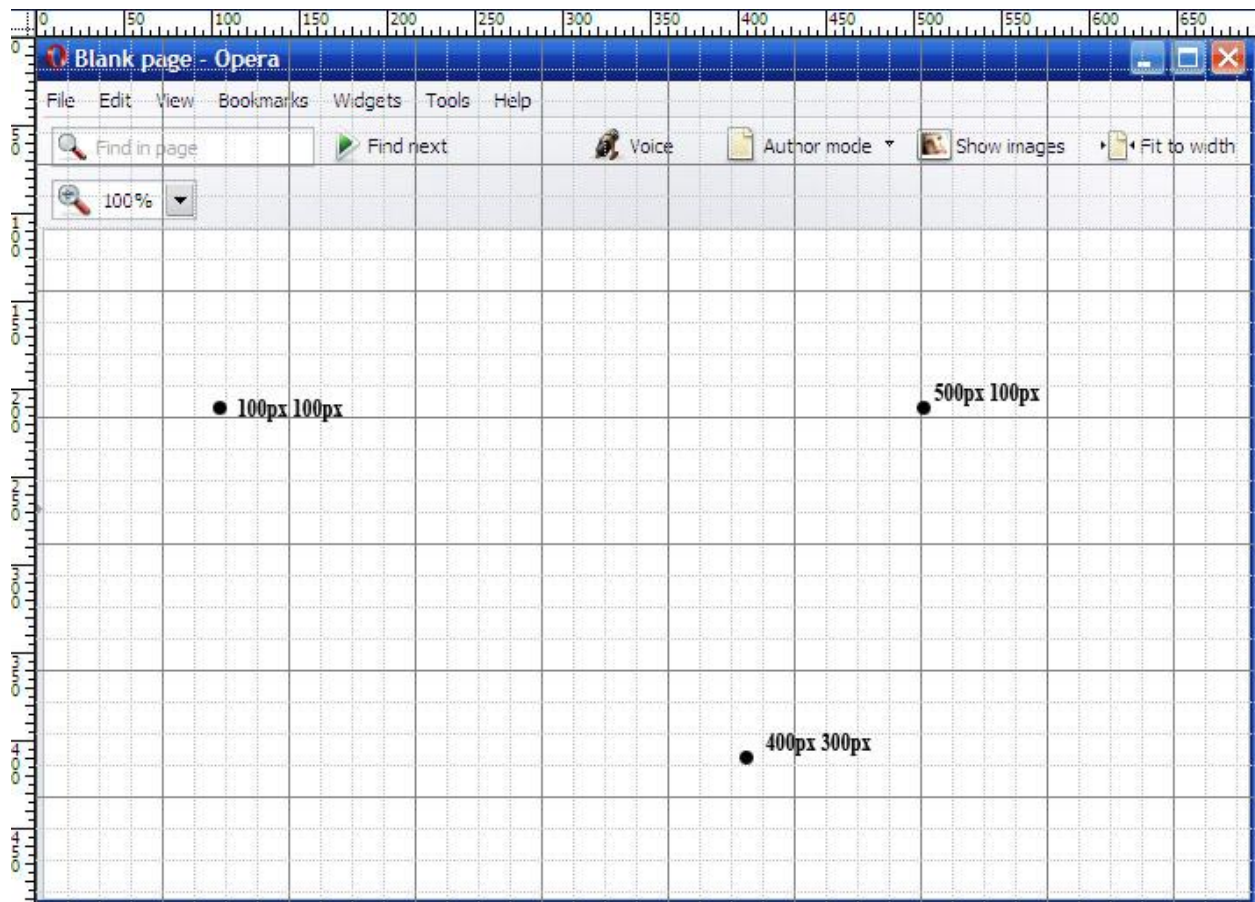
Bài 14: Position



Kết hợp với thuộc tính float đã học, thuộc tính position mang lại nhiều khả năng để tạo một cách trình bày tiên tiến và chính xác cho trang web.

❖ 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, ảnh trong suốt hay bất kỳ thứ gì khác.

14.1. 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 }
```

14.2. 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.

Hai bài học trên, bạn đã được học về 2 thuộc tính CSS là float và position. Hai thuộc tính này mang lại cho bạn nhiều sự lựa chọn hơn trong việc dàn trang. Nó chính xác và dễ thực hiện hơn so với các phương pháp dùng bảng hay ảnh trong suốt.

Bài 15: Layers



CSS hoạt động trên cả 3 chiều: cao, rộng, sâu. Hai chiều đầu tiên, chúng ta đã được nhìn thấy trong các bài học trước. Trong bài học này, chúng ta sẽ được học về cách đặt các thành phần web ở các lớp khác nhau với thuộc tính z-index. Nói đơn giản hơn thì đó là cách bạn 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  
}
```

Bài 16: 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 [click here](#).

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.

Phụ Lục: Bảng Thuộc Tính & Giá Trị Thuộc Tính

Thuộc tính	Mô tả	Giá trị
Background		
background	Thuộc tính rút gọn cho tất cả các thuộc tính nền.	background-color background-image background-repeat background-attachment background-position
background-color	Thiết lập màu nền cho đối tượng.	<color> transparent
background-image	Thiết lập ảnh nền cho đối tượng.	url none
background-repeat	Thiết lập chế độ lặp ảnh nền.	repeat repeat-x repeat-y no-repeat
background-attachment	Thiết lập ảnh nền cuộn/cố định.	scroll fixed
background-position	Thiết lập vị trí thể hiện ảnh nền.	top left top center top right center left center center center right bottom left bottom center bottom right x% y% x y
Font		
font	Thuộc tính ngắn cho tất cả các thiết lập về font.	font-style font-variant

		font-weight font-size font-family
font-style	Thiết lập chế độ in nghiêng, xiên hay bình thường.	normal italic oblique
font-variant	Thiết lập font bình thường hay small-caps	normal small-caps
font-weight	Thiết lập in đậm, thường.	normal bold bolder lighter 100 – 900
font-size	Thiết lập kích cỡ font.	xx-small x-small small medium large x-large xx-large smaller larger <length> %
font-family	Thiết lập loại font hiển thị trang web/ đối tượng web.	family-name generic-family
Text		
color	Thiết lập màu chữ.	<color>
text-indent	Thiết lập khoảng thụt đầu dòng.	<length> %
text-align	Thiết lập chế độ canh văn bản.	left right center justify
letter-spacing	Thiết lập khoảng cách giữa các ký tự.	normal <length>
text-decoration	Thêm hiệu ứng đặc biệt cho văn bản.	none underline

		overline line-through blink
text-transform	Change case văn bản.	none upper lower capitalize
Pseudo-classes		
:link	Liên kết chưa thăm.	
:hover	Mouse over 1 thành phần.	
:visited	Liên kết đã thăm.	
:active	Kích hoạt 1 thành phần	
Margin		
margin	Thuộc tính ngăn cho các thiết lập margin.	margin-top margin-right margin-bottom margin-left
margin-top	Thiết lập canh lề trên cho một thành phần.	auto <length> %
margin-right	Thiết lập canh lề phải cho một thành phần.	auto <length> %
margin-bottom	Thiết lập canh lề dưới cho một thành phần.	auto <length> %
margin-left	Thiết lập canh lề trái cho một thành phần.	auto <length> %
Padding		
padding	Thuộc tính ngăn cho các thiết lập padding.	padding-top padding -right padding -bottom padding -left
padding-top	Thiết lập đệm trên cho một thành phần.	<length> %
padding-right	Thiết lập đệm phải cho một thành phần.	<length> %

padding-bottom	Thiết lập đệm dưới cho một thành phần.	<length> %
padding-left	Thiết lập đệm trái cho một thành phần.	<length> %
Border		
border	Thuộc tính ngăn cho tất cả các thiết lập border cho một thành phần.	border-width border-color border-style
border-width	Thiết lập độ rộng đường viền.	thin medium thick <length>
border-color	Thiết lập màu cho đường viền.	<color>
border-style	Thiết lập kiểu đường viền.	none hidden solid dotted dashed double groove ridge inset outset
border-top-width	Thiết lập độ rộng viền phía trên.	<border-width>
border-top-color	Thiết lập màu viền phía trên.	<border-color>
border-top-style	Thiết lập kiểu viền phía trên.	<border-style>
border-right-width	Thiết lập độ rộng viền phải.	<border-width>
border-right-color	Thiết lập màu viền phải.	<border-color>
border-right-style	Thiết lập kiểu viền phải.	<border-style>
border-bottom-width	Thiết lập độ rộng viền bên dưới.	<border-width>
border-bottom-color	Thiết lập màu viền bên dưới.	<border-color>
border-bottom-style	Thiết lập kiểu viền bên dưới.	<border-style>
border-left-width	Thiết lập độ rộng viền trái.	<border-width>
border-left-color	Thiết lập màu viền trái.	<border-color>

border-left-style	Thiết lập kiểu viền trái.	<border-style>
Width		
width	Thiết lập chiều rộng đối tượng.	auto <length> %
max-width	Thiết lập chiều rộng tối đa cho đối tượng.	none <length> %
min-width	Thiết lập chiều rộng tối thiểu cho một đối tượng.	<length> %
Height		
height	Thiết lập chiều cao cho một đối tượng.	auto <length> %
max-height	Thiết lập chiều cao tối đa cho một đối tượng.	none <length> %
min-height	Thiết lập chiều cao tối thiểu cho một đối tượng.	<length> %
Layout Position		
float	Cố định đối tượng.	left right none
clear	Cách thức xử sự của một đối tượng liên quan với đối tượng floated.	left right both none
position	định vị đối tượng	relative absolute
top	Thiết lập tọa độ đỉnh đối tượng.	auto <length> %
right	Thiết lập tọa độ bên phải đối tượng.	auto <length> %
bottom	Thiết lập tọa độ đáy đối tượng.	auto <length> %

left	Thiết lập tọa độ bên trái đối tượng.	auto <length> %
z-index	Định lớp đối tượng.	auto number