#### ST: <u>http://conghung.com</u> , <u>http://conghung.com/forum</u>

## PHƯƠNG PHÁP VIẾT MÃ CSS

Việc viết mã CSS (http://letunglam.com/category/css/) cũng giống như bạn lập trình với ngôn ngữ PHP (http://letunglam.com/category/lap-trinh-php/), C#,... tất cả đều cần có một bố cục khoa học, hệ thống để dễ dàng phát triển cũng như kiểm tra phát hiện lỗi (nếu có). Dưới đây là một vài hướng dẫn giúp bạn làm việc với CSS khoa học hơn:

1. Chú thích cho mã CSS:

Chú thích cho mã CSS giúp người khác đọc file CSS sẽ biết thêm những thông tin cần thiết về file CSS nói riêng và về tác giả nói chung. Việc chú thích mã ở những ngôn ngữ khác quan trọng ra sao thì chú thích mã ở CSS cũng quan trọng như vậy. Sau đây là mẫu chú thích một đoạn mã CSS rất tốt;

CSS:

/\*-----Screen Stylesheet version: 1.0 date: 01/03/07 author: [your email] email: [you at domain dot com] website: [your domain] version history: [location of file]

2. Chia CSS ra thành nhiều phần

Nếu mã CSS của bạn gồm nhiều phần và cho nhiều trang thì bạn nên chia thành nhiều file CSS để dễ quản lí và cũng để giúp cho file CSS của bạn không bị rối. Rất nhiều web developer chưa nhận thức được điều này. Họ gộp tất cả file CSS vào làm một. Và dĩ nhiên, hệ quả là họ mất nhiều thời gian hơn cho việc sửa file CSS của mình. Chia CSS ra thành phiều file và sử dụng chúng cùng với CSS chính bằng phương thức sau:

Chia CSS ra thành nhiều file và sử dụng chúng cùng với CSS chính bằng phương thức sau:

CSS:

/\* Import other stylesheets
-----\*/
@import url("typography.css");

Bên cạnh đó, phân chia ngay chính trong file CSS cũng quan trọng không kém. Hãy gộp chung những phần có cùng 1 đối tượng.

CSS: /\* Header -----\*/ /\* Navigation -----\*/ /\* Footer -----\*/ /\* Homepage -----\*/ /\* Your template -----\*/

Xoá các định dạng mặc định

Đây là điều cực kỉ cần thiết đối với bất kì Web developer nào. Như bạn đã biết thì mỗi trình duyệt (browser) đều hiển thị khác nhau. Phần lớn là do định dạng mặc định ở mỗi browser là khác nhau. Ví dụ sau sẽ giúp bạn xoá định dạng mặc định:

CSS:

\*{margin: 0;padding: 0;border: 0;}

4. Định dạng các đối tượng cơ bản:

Những đối tượng cơ bản hay được sử dụng như h1, h2, h3, ... form, table, cần phải được định dạng trước tiên khi bạn bắt đầu viết mã CSS. Thói quen này giúp bạn đồng bộ được giao diện của các trang web.[

```
CSS:
/* Forms
 -----*/
input.text
{
padding: 3px;
border: 1px solid #999999;
}
CSS:
/* Tables
-----*/
table
{
border-spacing: 0;
border-collapse: collapse;
}
td
{
text-align: left;
font-weight: normal;
```

```
}
```

Chú ý: đây không phải là một chuẩn mực viết mã CSS. Vì hiện tại chưa có 1 chuẩn nào cho việc viết CSS. Dưới đây chỉ là cách viết mã CSS sao cho khoa học được tích luỹ từ những ngày tháng làm việc cùng với CSS. Hi vọng điều này sẽ bố ích cho các bạn.

# LẬP TRÌNH CSS

1. Định dạng chữ mà không cần dùng đơn vị pixel.

Đôi khi, bạn tự hỏi làm sao những người thiết kế web (http://letunglam.com/category/thiet-ke-web/) lại dùng đơn vị em thay vì px? Rất đơn giản, chỉ với một thủ thuật nhỏ bạn cũng có thể làm được với đơn vị em thay vì px. Hãy thêm đoạn mã sau vào CSS (http://letunglam.com/category/css/). (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

```
CSS:
body { font-size: 62.5% }
```

Với đoạn mã trên thì 1.0em sẽ tương đương với 10px và sẽ hoàn toàn dễ hơn cho bạn khi tính toán kích thước font chữ hơn. Và khi đó bạn có thể định dạng như sau: (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

CSS: p { font-size: 1.2em; line-height: 1.5em; }

2. Trình bày CSS sáng sủa hơn Trình bày CSS một cách khoa học sẽ giúp việc phát triển, sửa chữa dễ dàng hơn bao giờ hết. Hãy chia các định dạng (styles) ra thành các block riêng biệt và trình bày chúng như những ngôn ngữ lập trình bạn vẫn thường làm. (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

CSS:

h1 {}
h1#logo { font-size: 2em; color: #000; }
h2 {}
h2.title { font-size: 1.8em; font-weight: normal; }

Bạn có thể tham khảo chi tiết hơn về cách vết mã CSS tại bài viết Phương pháp viết mã CSS

3. Hạn chế dùng div

Khi mới tiếp cận với CSS việc lạm dụng div thường là lỗi mà các lập trình viên dễ mắc phải. Hãy dùng đúng chức năng của mỗi tag.Ví dụ: hãy dùng h1, h2, h3 cho tiêu đề thay vì dùng div để định dạng.

4. Tối ưu mã CSS.

Tối ưu mã CSS ở đây là giảm độ dài của dòng lệnh CSS cũng có nghĩa giảm dung lượng với file CSS. Điều này sẽ giúp trang web được tải về nhanh hơn. Sau đây là một số ví dụ giúp bạn rút gọn mã CSS của mình.

Thay vì dùng #000000 bạn có thể dùng #000 cho định dạng màu đen. Trường hợp khác với #ffffcc bạn có thể dùng #ffc.

CSS: font-size: 1em; font-family: Arial, Helvetica, Sans-Serif; line-height: 1.5em; font-weight: bold;

Đoạn mã trên giúp định dạng font và đoạn cho toàn trang web. Nhưng bạn có thể dùng đoạn mã sau để thay thế:

(http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

CSS:

body { font: 1em/1.5em bold Helvetica, Arial, Sans-serif; }

Hoặc với trường hợp: (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

CSS:

margin-top: 5px; margin-left: 5px; margin-bottom: 0; margin-right: 0;

Bạn có dùng đoạn mã ngắn gọn hơn rất nhiều: (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

CSS: body { margin: 5px 0 0 5px; }

Cú pháp của hàm margin như sau: (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

CSS: margin: [top] [right] [bottom] [left]

2. hoặc trong trường hợp lề trên dưới và trái phải bằng nhau, bạn có thể dùng cú pháp sau:

CSS: margin: [top/bottom] [left/right] (http://letunglam.com/lap-trinh-web/lap-trinh-voicss/#)

Trường hợp khác, ngay trong chính một bài viết (http://letunglam.com/css/lam-viec-tot-

voi-cac-browser/) của mình, xin cám ơn Sonny (http://sonnymotives.com/) và Quietman đã gợi ý một cách lập trình CSS khoa học. Đó là bạn có thể dùng (http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)

```
CSS:
* { margin:0; padding:0; }
```

```
thay vì phải dùng đoạn mã dài lê thê sau:
(http://letunglam.com/lap-trinh-web/lap-trinh-voi-css/#)
```

```
CSS:
body, div, dl, dt, dd, ul, ol, li, h1, h2, h3, h4, h5, h6,
pre, form, fieldset, input, p, blockquote, th, td {
margin: 0;
padding: 0;
}
```

## NHỮNG ĐIỀU CẦN BIẾT KHI VIẾT MÃ CSS

Như chúng ta đã biết ở bài viết trước (http://letunglam.com/css/lam-viec-tot-voi-cacbrowser/) mình đã có lần để cập đến vấn đề "làm việc tốt với các trình duyệt (browser)" bằng cách sử dụng CSS (http://letunglam.com/category/css/) để xóa đi những định dạng mặc định của từng browser. Lần này, các bạn sẽ được biết đến những điều cần tránh trong khi viết mã HTML và CSS nếu bạn muốn website của mình hiển thị tốt ở hầu hết các browser.

1. Không bao giờ sử dụng padding cho các đối tượng có độ rộng cố định (fixed width) Đây là nguyên tắc cơ bản vì cách căn lề của Internet Explorer khác với các trình duyệt khác và cũng là 1 lỗi của trình duyệt này. Lỗi này được gọi là "Box Model Bug (http://en.wikipedia.org/wiki/Internet\_Explorer\_box\_model\_bug)" hình dưới là một hình ảnh minh hoa cho lỗi này.

http://letunglam.com/wp-content/uploads/box-model-bug.png

Bạn có thể thấy rằng độ rộng của đối tượng "được" cộng thêm khi hiển thị ở IE nếu như có padding. Vậy đơn giản là không dùng padding cho các đối tượng có độ rộng cố định (fixed width).

2. Cố định cỡ chữ bằng đơn vị % và em

Thông thường các browser hiển thị cỡ chữ theo cách khác nhau nếu như bạn chưa định cỡ cho chúng. Khi mới bắt đầu học CSS, chúng ta thường định dạng cỡ chữ theo cách sau: (http://letunglam.com/lap-trinh-web/can-biet-khi-viet-ma-css/#)

CSS: h1 {font-size:18px;} h2 {font-size:16px;

Dòng mã là cách cơ bản, nhưng vấn đề là ở IE người xem sẽ không thể thay đổi cỡ chữ. Điều này rất cần thiết với những người để độ phân giải cao. Nhưng bạn đừng lo, có một giải pháp khác tốt hơn, hãy dùng % và em để định dạng cỡ chữ như sau: (http://letunglam.com/lap-trinh-web/can-biet-khi-viet-ma-css/#)

CSS: body {font-size:62.5%;} h1 {font-size:1.8em;} h2 {font-size: 1.6em;}

#### 3. Không nên dùng đơn vị 100%

Sử dụng 100% để căn chiều rộng của một đối tượng rất dễ gặp rủi ro trong việc bố trí kết cấu của website vì mỗi trình duyệt hiển thị chiều rộng 100% rất khác biệt. Theo kinh nghiệm bản thân thì những lần thử định vị 100% thì gần như rất khó có thể đạt được kết quả tốt nhất. Hãy chọn một con số khác có thể là nhỏ hơn 100% (95% có thể sẽ tốt hơn).

Hoặc đơn giản hơn là hãy... cố định nó.

4. Không dùng 0px để định chiều cao Các browser không thích việc bạn chọn 0px để định chiều cao cho đối tượng và thưởng hiển thị nó theo cách bạn... không muốn chút nào. Nếu có thể hãy thay nó bằng 1px sẽ tốt hơn rất nhiều. (http://letunglam.com/lap-trinh-web/can-biet-khi-viet-ma-css/#) CSS:

height:1px

5. Không dùng ">" trong CSS Mình đã từng thấy có người, thậm chí rất nhiều người viết mã CSS như sau: (http://letunglam.com/lap-trinh-web/can-biet-khi-viet-ma-css/#)

CSS:

#menubar>a {color:#cccccc;}

Với dòng mã CSS như trên IE6 sẽ không hiểu và không thể hiển thị được định dạng trên. Hãy dùng đoạn mã dưới đây, sẽ an toàn hơn rất nhiều. (http://letunglam.com/lap-trinh-web/can-biet-khi-viet-ma-css/#) CSS: #menubar a {color:#cccccc;}

# CSS LÀM VIỆC TỐT VỚI CÁC TRÌNH DUYỆT

Môt điều đau đầu với những người làm việc với CSS (http://letunglam.com/category/css/) đó là viêc các trình duyêt (browser) hiến thi website theo cách không hề giống nhau. Điều này là do các định dạng mặc định (default styles) của mỗi browser khác nhau. Vậy, làm sao để han chế được thấp nhất các vấn đề có thể nảy sinh với browser khi làm viêc với CSS. Có thể, ban đã có lưa chon của mình, ở đây sẽ có thêm 1 lưa chon nữa cho ban. Với đoạn mã CSS dưới đây, bạn sẽ loại bỏ được gần như hoàn toàn những định dạng mặc định của các browser. (http://letunglam.com/css/lam-viec-tot-voi-cac-browser/#) CSS: /\* Normalize padding and margins \*/ body, div, dl, dt, dd, ul, ol, li, h1, h2, h3, h4, h5, h6, pre, form, fieldset, input, p, blockquote, th, td { margin: 0; padding: 0; } /\* Normalize header sizes \*/ h1, h2, h3, h4, h5, h6 { font-size: 100%; } /\* Normalize list styles \*/ ol, ul { list-style: none; } /\* Normalize font style and weight on odd elements \*/ address, caption, cite, code, dfn, em, strong, th, var { font-style: normal; font-weight: normal; } /\* Normalize table borders \*/

```
/* Normalize table borders *
table {
border-collapse: collapse;
```

```
border-spacing: 0;
}
/* Normalize other borders */
fieldset, img {
    border: 0;
    }
/* Normalize text-alignment */
caption, th {
    text-align: left;
    }
    /* Normalize (remove) quotation marks */
q:before, q:after {
    content: '';
    }
```

Chú ý: Với đoạn mã trên chỉ giúp bạn tránh được nhiều vấn đề gặp phải với CSS, nhưng không phải tất cả. Vì thế, khi gặp những vấn đề liên quan đến CSS, có lẽ Google (http://letunglam.com/category/google/) là người trợ giúp tốt nhất.

## ĐỊNH DẠNG 1 BẢNG ĐƠN GIẢN BẰNG CSS

Như chúng ta đã biết để định dạng một bảng(table) đối với một số chương trình phát triển Web như Dreamwaver, Fontpage,...thì hết sức đơn giản bởi các chương trình này cung cấp cho chúng ta cách thức thay đối các thông số của bảng một cách trực quan. Nhưng hầu hết các chương trình này sau khi định dạng sẽ sinh ra các mã định dạng của bảng ngay trong mã nguồn của trang HTML, với mỗi bảng các chương trình này sẽ sinh ra các mã định dạng riêng cho từng bảng đó. Như vậy sẽ có một số vấn đề bất cập.

Nếu chúng ta có 10 bảng giống nhau thì chúng ta phải định dạng tới 10 lần. Điều đó sẽ mất rất nhiều thời gian của chúng ta.

Khi chúng ta định dạng nhiều bảng giống nhau như vậy, mà mỗi bảng lại sinh ra một loạt mã định dạng cho riêng mình. Điều đó đồng nghĩa với việc sẽ có rất nhiều các đoạn mã dư thừa trong trang HTML của chúng ta.Để khắc phục được những bất cập trên sau đây tôi xin trình bày với các bạn một kỹ thuật đơn giản để định dạng một bảng dữ liệu đơn giản.

Để định dạng được trong CSS chúng ta sẽ thêm cho bảng cần định dạng một thuộc tính class="tablelist"

```
Số thứ tự
Họ và tên
Giới tính
1
Nguyễn Quang Tho
Nam
2
Trần Mai Dung
Nữ
Tiếp theo bạn thêm đoạn mã sau vào trong file CSS.
```

.tablelist {

font-family: Arial, Helvetica, sans-serif; font-size: 12px; margin: 0; padding: 0; border-collapse: collapse; border-spacing: 0px; border: 1px solid #EBDDBC; } .tablelist th { margin: 0;

margin: 0; padding: 4px 6px; border: 1px solid #EBDDBC; background-color: #F1EFD8;

```
}
```

.tablelist td { margin: 0; padding: 4px; border: 1px solid #EBDDBC; }Nếu bạn có nhiều bảng có định dạng giống nhau thì bạn chỉ việc thêm thuộc tính class="tablelist" vào trong thẻ .

# THIẾT KẾ TAB ĐƠN GIẢN VỚI CSS

Trong chúng ta tôi giám chắc rằng không còn ít người vẫn còn mặc cảm với những menu dạng tab, không phải vì nó xấu mà luôn nghĩ rằng làm nó khó và cấu trúc của nó rất phức tạp. Chính bản thân tôi trước đây cũng như vậy. Nhưng thực ra có phải như vậy không? để trả lời câu hỏi đó bây giờ chúng ta cùng nhau thử Style cho một kiểu tab đơn giản để xem nó có thực sự khó như chúng ta vẫn thường nghĩ không?.

Trong menu dạng này bao giờ cũng gồm 2 phần, phần 1 để chứa các tab và phần thứ hai là phần để chứa nội dung của các tab. để có hình dung rõ hơn chúng ta sẽ xem hình ảnh minh họa dưới đây.

http://www.cssyeah.com/images/content/sample-tab.gif

Đầu tiên chúng ta sẽ định dạng cho phần khung của tab như sau:

```
div#wrapper {
margin: 50px;
padding: 0;
```

}

Chúng ta sẽ dùng định dạng của thẻ để tạo lên các tab ở phần 1. Về nguyên tắc thì chúng ta có thể dùng một trong ba thẻ , , <dl>. Nhưng tại sao tôi lại dùng thẻ , là bởi vì một lý do nào đó mà trình duyệt của người xem không tải được CSS từ Website của bạn thì họ vẫn hiểu được cấu trúc tab của bạn. Định dạng của phần một như sau

```
Trang chủ
<a href="gioithieu.html" title="Giới thiệu">Giới thiệu</a>
<a href="sanpham.html" title="Sản phẩm">Sản phẩm</a>
<a href="tintuc.html" title="Tin tức">Tin tức</a>
<a href="lienhe.html" title="Liên hệ">Liên hệ</a>
<a href="lienhe.html" title="Liên hệ">Liên hệ</a>
Bây giờ chúng ta sẽ định dạng sao cho các tab nằm trên cùng một hàng và có hình dáng
của tab.
```

```
ol.tab {
background: url(../dot.gif) repeat-x left bottom; /* anh 1px */
list-style: none;
```

```
margin: 0;
padding: 6px 0;
position: relative;
}
ol.tab li {
background: #F2F5FA;
border: 1px solid #D3DDED;
display: inline; /* các thẻ li ở trên một dòng */
margin-right: 5px;
padding: 0;
}
Trong cách đinh dang này ta có sử dung môt kỹ thuật nhỏ, đó là chúng ta có sử dung một
```

ản

# CÁCH VIẾT GIẢN LƯỢT TRONG CSS

Nếu bạn là người thực sự muốn tìm hiểu về CSS, thì bạn không thể không tìm hiểu cách viết giản lược(shorthand) trong CSS. Cách viết này thực sự mang lại những lợi ích và tiện lợi khi bạn sử dụng CSS.

Trước khi đi vào phân tích những tiện lợi mà nó mang lại, tôi xin lấy một ví dụ đơn giản như sau. Giả sử rằng chúng ta muốn định dạng cho một thẻ <div> có đường viền bao quanh thì chúng ta cần phải viết.

border-width: 1px; border-style: solid; border-color: #CC0000;Thay vì phải viết như vậy chúng ta chỉ cần viết

border: 1px solid #CC0000;Qua ví dụ đơn giản ở trên ta có thể thấy cách viết này mang lại cho chúng ta một số lợi ích sau. Thứ hai

Thứ nhất: nó giúp chúng ta giảm thiểu được đáng kể thời gian phải viết mã CSS. Thứ hai: bạn cứ tưởng tượng rằng file CSS của bạn có tới vài ngìn dòng và dung lượng lên tới vài trăm Kb, thì cách viết này còn giúp bạn giảm thiểu được đáng kể dung lượng của file CSS và giúp bạn dễ dàng theo dõi hơn, khi số lượng dòng của trang được giảm xuống.Sau đây tôi xin đi vào chi tiết một số thuộc tính trong CSs mà chúng ta có thể dùng cách viết giản lược.

1. Thuộc tính Color

Có rất nhiều cách để định nghĩa một màu trong CSS, chúng ta có thể định nghĩa theo hệ số Hexa(trong hệ màu RGB), hoặc chúng ta có thể viết tên màu (ví dụ: white, red...). Nhưng cách định nghĩa theo hệ số Hexa được sử dụng thông dụng nhất. Để định nghĩa theo hệ Hexa chúng ta sẽ đặt dấu (#) ở trước sau đó đến các thông số màu (ví dụ: #003366). Dãy các thông số màu được chia làm 3 phần tương ứng với ba màu Red, Green, Blue (00: Red | 33: Green | 66: Blue). Trong cách định nghĩa hệ số màu ta cũng có cách viết giản lược như sau: #000000 có thể viết #000 hoặc #003366 có thể viết #036 2. Thuộc tính margin và padding.

margin-top: 10px; margin-right: 15px; margin-bottom: 20px; margin-left: 25px;Được thay thế bằng

margin: 10px 15px 20px 25px; /\* top | right | bottom | left \*/Tương tự với thuộc tính padding

padding-top: 10px; pading-right: 15px; padding-bottom: 20px; padding-left: 25px; padding: 10px 15px 20px 25px; /\* top | right | bottom | left \*/Cå hai thuộc tính margin và padding có hai chú ý như sau: nếu trong trường hợp có hai thông số.

margin: 10px 20px; /\* top bottom | right left \*/ padding: 10px 20px; /\* top bottom | right left \*/Thì thông số thứ nhất tương đương với top và bottom còn thông số thứ hai tương đương với right và left Trong trường hợp margin và padding có 3 thông số:

margin: 10px 20px 15px; /\* top | right left | bottom \*/ padding: 10px 20px 15px; /\* top | right left | bottom \*/Thì thông số thứ nhất tương đương với top, thông số thứ hai tương đương với right và left, thông số thứ 3 tương đương với bottom

3. Thuộc tính border.

border-with: 1px; border-style: solid; border-color: #CC0000;Sẽ viết thành

border: 1px solid #CC0000; /\* border-width | border-style | border-color \*/4. Thuộc tính background.

background-color: #CC0000; background-image: (image url); background-repeat: no-repeat; /\* repeat-x, repeat-y \*/ background-position: top left;Tương đương với

background: #CC0000 url(image url) no-repeat top left;5. Thuộc tính font

font-size: 1em; line-height: 1.5em; font-variant:small-caps; font-weight: bold; font-style: italic; font-famyli: Arial, Verdana, Sans-serif;Dang viết giản lược

font: 1em/1.5em bold italic small-caps Arial, Verdana, Sans-serif; 5. List type

list-style: none;Có nghĩa là

list-style-type: none; Bạn cũng có thể sử dụng thuộc tính list-style-position và list-typeimage để định dạng cho danh sách không có thứ tự unordered lists, sử dụng hình ảnh cho mỗi dòng và sử dụng list-type-style là hình vuông trong trường hợp không hiển thị được ảnh. Và hai cách viết dưới đây là như nhau.

list-style:square inside url(image.gif);là giản lược cho

list-style-type: square; list-style-position: inside; list-style-image: url(image.gif);6. Outline

Thuộc tính này rất ít dùng vì có rất ít các trình duyệt hiện tại hỗ trợ thuộc tính này, hiện tại chỉ có một số trình duyệt hỗ trợ thuộc tính này Safari, OmniWeb và Opera. Cách viết giản lược các thuộc tính này như sau:

outline-color: #000; outline-style: solid; outline-with: 2px;Cách viết giản lược sẽ là outline: #000 solid 2px;

#### Kỹ THUẬT TẢI ẢNH TRƯỚC TRONG CSS (PRELOADER IMAGES)

Khi Website của bạn được từ trên máy chủ (Server) về máy người dùng (Client), các trình duyệt Web sẽ đọc theo thứ tự từng phần tử HTML có trong trang của bạn. Trong đó trình duyệt sẽ ưu tiên dữ liệu dưới dạng văn bản (Text), sau đó đến ảnh dạng <img...> kế tiếp là các ảnh nền (background).

Do vậy trong trường hợp đường truyền mạng của bạn có vấn đề, thì có một số hiệu ứng mà bạn sử dụng dạng ảnh nền (Background) sẽ không thực hiện được đúng như bạn mong muốn (Không hiển thị được ảnh). Để khắc phục được vấn đề này ta có thể sử dụng kỹ thuật tải trước hình nền trong CSS.

Hãy đặt đoạn code CSS này vào trong file CSS của bạn.

```
#preloadImages {
width: 0px;
height: 0px;
display: inline;
background-image: url(images/anh1.jpg);
background-image: url(images/anh2.png);
background-image: url(images/anh3.gif);
background-image: url();
}
```

# Kỹ THUẬT SLIDING DOORS VÀ ỨNG DỤNG

Đối với những desiger thường xuyên làm việc với CSS thì kỹ thuật Sliding Doors không có gì mới mẻ. Nhưng đối với những bạn mới làm quen với CSS thì tôi nghĩ rằng đây là một kỹ thuật mà bạn nên tìm hiểu nó.

Kỹ thuật Sliding Doors(kỹ thuật cửa kéo), là một kỹ thuật được ứng dụng rất nhiều trong việc tạo ra các button, hay các menu ngang kiểu tab,...Trong bài viết này chúng ta sẽ đi tìm hiểu ý tưởng chủ đạo của kỹ thuật và tạo ra một ví dụ nhỏ có sử dụng kỹ thuật trên.

Tôi chắc rằng ít nhất chúng ta đã một lần nhìn thấy chiếc cửa kéo kiểu nhật bản. Một bên cửa cố định và một bên cửa sẽ linh động có thể kéo ra hoặc kéo vào tùy thích. Bản chất của kỹ thuật Sliding Doors cũng giống như vậy.

Chúng ta sẽ chia đối tượng 2 phần (phần bến trái và phần bên phải). Phần bên trái sẽ là phần cố định, còn phần bên phải sẽ là phần động(Việc phân chia này tùy theo ý thích của bạn, bạn có thể phân chia phần bên phải cố định và phần bên trái là phần động). Điều đó cũng có nghĩa là phần bên trái sẽ có kích thước vừa phải, còn phần bên phải cần có kích thước luôn lớn hơn hoặc bằng nội dung mà bạn dự định đưa vào(để đảm bảo rằng đối tượng của chúng ta giống như bị đứt gãy).

Để bạn có một hình dung rõ hơn tôi xin đưa ra hình vẽ mô tả ý tưởng của kỹ thuật này như sau:

http://www.cssyeah.com/images/content/sliding-doors.png Kỹ thuật Sliding Doors.

Như trên hình vẽ bạn đã thấy, phần bên trái sẽ cố định và phần bên phải sẽ có ra hoặc dãn vào tùy theo nội dung bên trong của đối tượng cần tạo. Do vậy với cách làm như trên thì chúng ta cần tạo phần bên phải cần phải đủ dài để tránh trường hợp bị thiếu ảnh khi nội dung dài.

Các bạn thấy không nghe thì có vẻ ghê gớm nhưng về bản chất thì kỹ thuật này chỉ có vậy thôi, nhưng các bạn đừng coi thường, tuy vậy nhưng nó khá quan trọng và được ứng dụng rất rộng rãi trong việc tạo button, tạo menu kiểu tab,...Để minh họa cho điều tôi vừa trình bày bày sau đây tôi xin đưa ra một ví dụ có sử dụng kỹ thuật này.

Trong ví dụ này chúng ta sẽ tạo ra một số button theo phong cách web 2.0, và có độ tùy biến nội dung cao(chúng ta có thể thay đổi nội dung ở trong một cách tùy ý). Sau đây là chi tiết từng bước làm cụ thể.

1. Tạo nút bằng Photoshop

Việc đầu tiên chúng ta cần làm đó là chúng ta cần tạo ra hình dáng chiếc nút thân yêu của ta, trong ví dụ này tôi tạo ra một nút Sign Up Now! như sau: http://www.cssyeah.com/images/content/button.gif

Tạo button bằng Photoshop

2. Cắt nút ra thành hai phần.

Như tôi đã trình bày ở trên, để thực hiện được kỹ thuật này chúng ta cần phải cắt đối tượng ra làm 2 phần. Để có thể cắt hình được chính xác tôi khuyên bạn nên sử dụng công cụ Slice(Slice Tools) trong Adobe Photoshop. Button của chúng ta sẽ được slice như sau: http://www.cssyeah.com/images/content/button-slice.gif Slice thành hai phần. Chúng ta cần chú ý là phần bên phải cần phải đủ dài để đảm bảo rằng không bị thiếu.

3. Thực hiện viết mã HTML và CSS. Chúng ta có định dạng mã HTML như sau:

<a href="you link" title="Your title"> <span>...Nội dung...</span> </a>Giải thích:

Sở dĩ chúng ta cần phải sử dụng 2 phần thử HTML trong việc này là do đối tượng của chúng ta được chia làm 2 phần, và mỗi phần tử sẽ làm nhiệm vụ giữ một phần của đối tượng. Chúng ta sẽ dùng thẻ <a> bao ở ngoài cùng để đảm bảo rằng toàn bộ nút sẽ link được khi di chuột lên trên nút. Đồng thời thẻ <a> này sẽ chứa phần động bên phải. Thẻ <span> sẽ làm nhiệm vụ giữ phần cố định bên trái, đồng thời sẽ có nhiệm vụ che phần dự của ảnh nền bên phải.Để thỏa mãn với yêu cầu đặt ra chúng ta có mã định dạng CSS như sau:

/\* Sign up button style \*/ a.sign-up { font-family: Arial, Helvetica, sans-serif; font-size: 25px; font-weight: bold; color: #FFF; text-decoration: none; margin: 0; padding: 28px 0; display: block; background: url(images/signup-right-bg.gif) no-repeat right center; width: 270px; } a.sign-up:hover, a.sign-up:active { color: #FF0; } a.sign-up span { margin: 0; padding: 28px 0 28px 90px; background: url(images/signup-left-bg.gif) no-repeat left center; }Và kết quả chúng ta được một button rất đẹp như như trong thiết kế: Mã nguồn (<u>http://www.cssyeah.com/?dl=2</u>)

## MỘT VÀI CÁCH STYLE CHO ĐOẠN TRÍCH DẫN

Trong bài viết này tôi xin trình bày với các bạn một số cách để định dạng cho đoạn trích dẫn(blockquote). Để làm cho đoạn trích dẫn của bạn nổi bật trong phần nội dung và gây sự chú ý đối với người đọc, chúng ta có thể thêm một số yếu tố vào trong đó như font chữ, màu sắc, hình ảnh...

Như tiêu đề bài viết đã nêu, trong bài này chúng ta sẽ đi sâu vào các cách định dạng cho đoạn trích dẫn, chứ không đi sâu vào phần hình thức của nó. Bởi mỗi người sẽ có một cách trình bày khác nhau và sẽ thể hiện nó theo cách của mình. Có rất nhiều cách để định dạng cho thẻ <blockquote>, nhưng trong bài này tôi xin trình bày bốn cách cơ bản.

#### 1. Cách thứ nhất

Đầu tiên chúng ta sẽ đi từ cái đơn giản nhất đó là thêm một dấu nháy mở vào đầu của đoạn trích dẫn.Ví dụ minh họa (http://www.cssyeah.com/sample/quote/quote.html)

Mã HTML: <blockguote> Nôi dung đoan trích dẫn... </blockquote> Mã CSS: blockquote { margin: 0; padding-left: 40px; background: url(openquote.gif) left top no-repeat; font: 130% "Times New Roman", Arial, Verdana; font-style: italic; text-align: left; } 2. Cách thứ hai Trong cách thứ 2 này chúng ta sẽ có cả dấu mở và dấu đóng. Ta sẽ dùng thuộc tính :firstletter để thêm dấu nháy kép vào đầu đoan trích dân.Ví du minh hoa (http://www.cssyeah.com/sample/quote/quote.html) <blockquote class="quote-style2"> Nội dung đoạn trích dẫn... </blockquote>Mã CSS: .quote-style2 { line-height:140%: background: url(closequote.gif) no-repeat bottom right !important; padding:0px 25px 5px 0px; font: 130% "Times New Roman", Arial, Verdana; } .quote-style2:first-letter { background: url(openquote.gif) no-repeat left top !important; padding:5px 2px 10px 35px!important; } 3. Cách thứ ba Giống như với cách thứ hai đoạn trích dẫn của ta vẫn có hai dấu nháy kép đóng và mở. Trong cách này chúng ta sẽ dùng một thẻ <span> bao ký tự cuối cùng của đoạn trích dẫn. Thẻ <span> này sẽ chứa dấu nháy đóng của đoạn trích dẫn. Ví dụ minh hoạ (http://www.cssyeah.com/sample/quote/quote.html). Ta có đinh dang HTML của kiểu này như sau: <blockquote class="quote-style3" Nôi dung đoan trích dẫn..<span>.</span> </blockguote>Dinh dang CSS blockquote { margin: 0; padding-left: 40px; background: url(openquote.gif) left top no-repeat; font: 130% "Times New Roman", Arial, Verdana; font-style: italic; text-align: left; } .quote-style3 span { margin: 0; padding: 10px 40px 10px 0; background: url(closequote.gif) no-repeat right center; } 4. Cách thứ tư Về hình thức thì đoạn trích dẫn của chúng ta giống như kiểu 2 và kiểu 3 nhưng cách thêm

dấu nháy đóng của kiểu này hơi khác. Chúng ta sẽ dùng thẻ <div> hoặc thẻ để thêm dấu nháy đóng cho đoạn trích dẫn, Ví dụ minh họa (http://www.cssyeah.com/sample/quote/quote.html). Dưới dây là định dạng HTML và CSS của kiểu này. Đinh dang HTML

<blockquote> Nội dung đoạn trích dẫn... </blockquote>Định dạng CSS

blockquote {
margin: 0;
padding-left: 40px;
background: url(openquote.gif) left top no-repeat;
font: 130% "Times New Roman", Arial, Verdana;
font-style: italic;
text-align: left;
}.quote {
margin: 0;
padding-right: 20px;
background: url(closequote.gif) no-repeat right bottom;
}Các bạn có thể tải mã nguồn tại đây:
Mã nguồn (http://www.cssyeah.com/?dl=1)

### THÊM BACKGROUND VÀ DANH SÁCH CÓ THỨ TỰ (ORDER LIST)

Để làm phong phú cách trình bày văn bản (Typography) trên Website của bạn, bạn có thể thêm ảnh nền vào trong phần số của danh sách có thứ tự (Order list).

Chúng ta thêm thuộc tính class="bigol" vào trong thẻ .

```
<span>Thứ nhất</span>
<span>Thứ hai</span>
<span>Thứ ba</span>

Thêm đoan mã CSS sau vào trong file CSS
ol.bigol {
padding: Opx Opx Opx 27px;
background: url(images/list_disc.png) 0 0 repeat-y;
}
ol.bigol li {
padding: Opx Opx 6px 12px;
font-weight: bold;
color: #ffffff;
line-height: 28px;
}
ol.bigol li span {
color: #646464;
}Xem ví du (http://www.cssyeah.com/Sample/olbackground/sample.html)
```

# ĐÙA NGHỊCH VỚI CÁC THUỘC TÍNH CỦA CSS

Trong khi blog vẫn chưa đi vào phần chính của nó "Questions" (vì quá ít người quan tâm đến CSS), các bạn tạm thời chơi với CSS bằng link này:

http://www.qrone.org/cssdesigner.html (http://www.qrone.org/cssdesigner.html)

Hi vọng khi nghịch ngợm với các thuộc tính CSS bạn sẽ hiểu hơn về nó. Như bạn thấy, CSS ko có quá nhiều "thuộc tính", khoảng 58 bác thôi.

Dần dần bạn sẽ biết cần áp dụng thuộc tính nào trong trường hợp nào http://my.opera.com/community/graphics/smilies/smile.gif, và có phản xạ tốt hơn về layout.

List:

color : black transparent background-color : transparent border-width : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % border-top-width : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12ptcm mm in % border-left-width : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12ptcm mm in % border-right-width : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12ptcm mm in % border-bottom-width : px = pixels pt = 1/72in em = font-size ex = x-height of font pc =12pt cm mm in % border-color : transparent border-top-color : transparent border-left-color : transparent border-right-color : transparent border-bottom-color : transparent border-style : none solid double groove ridge inset outset dashed dotted border-top-style : none solid double groove ridge inset outset dashed dotted border-left-style : none solid double groove ridge inset outset dashed dotted border-right-style : none solid double groove ridge inset outset dashed dotted border-bottom-style : none solid double groove ridge inset outset dashed dotted width : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in %height : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in% overflow : visible hidden scroll auto padding : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in% padding-top : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % padding-left : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % padding-right : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % padding-bottom : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12ptcm mm in % margin : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in% margin-top : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mmin % margin-left : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mmin % margin-right : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % margin-bottom : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % font-family : serif sans-serif cursive fantasy monospace font-size : 9pt 10pt 12pt 14pt 16pt xx-small x-small small large x-large xx-large font-weight : normal bold 600 900 font-style : normal italic oblique

font-variant : normal small-caps

line-height : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mmin % clear : left right both none float : left right none text-align : left right center justify text-decoration : none underline overline underline overline line-through blink text-indent : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mmin % letter-spacing : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % word-spacing : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cmmm in % text-transform : none capitalize lowercase uppercase vertical-align : baseline top middle bottom text-top text-bottom super sub 3em 30% background-image : url(grx/resize-box.gif) background-repeat : repeat repeat-x repeat-y no-repeat background-position : left right top left top right bottom 30% 50% background-attachment : fixed scroll display : none block inline run-in compact list-item marker visibility : visible hidden position : static relative absolute fixed top : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in %left : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in %right : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in %bottom : px = pixels pt = 1/72in em = font-size ex = x-height of font pc = 12pt cm mm in% z-index : cursor :

#### 6 GIAI ĐOẠN CỦA NGƯỜI HỌC CSS

Từ 1 bài viết của pcdinh trên ddth

(http://ddth.com/showpost.php?p=756429&postcount=6) và bài viết trên friendlybit (http://friendlybit.com/css/levels-of-css-knowledge/) tớ tổng hợp thành bài viết này:

bạn đã bao giờ tham gia 1 chat room về CSS chưa? (kiểu như #CSS trên EFNet) bản thân tôi thì chưa? nói chung đại khái người học CSS có mấy mức độ sau:

Giai đoạn quan niệm:
 Bạn còn đang tò mò về CSS.
 Câu nói đặc trưng: CSS là gì nhỉ? Một video game à? có phải Counter Strike ko?

2) Giai đoan thử nghiêm:

2a + Bạn đã bắt đầu nghịch ngợm CSS bằng cách view source 1 số website dùng CSS và bắt đầu dùng CSS vào những trang web mà trước đây bạn toàn dùng HTML, thường là chỉ với những gì mà bạn không thể dùng HTML được (mà chỉ CSS làm được). Câu nói đặc trưng: đôi khi mình cũng bỏ cái gạch chân ở dưới link đi

2b ++ Thay vì chỉ nghịch ngợm với HTML, bạn hiểu nhiều về cú pháp HTML và thường dùng table để tạo layout cho web, giờ bạn muốn tìm hiểu cách tạo layout mà ko dùng table. Thậm chí bạn biết khá rõ về cú pháp CSS nhưng bạn vẫn thấy việc tạo layout bằng CSS là khó.

Câu nói đặc trưng: Tôi không thích div, dùng table dễ hơn.

3) Giai đoạn trải nghiệm: Bạn đã ý thức được lợi ích của CSS và biết khá rõ về CSS cũng như thử dùng nó để làm layout cho web, nhưng có vẻ layout của bạn vỡ tung khi dùng với IE hoặc những trình duyệt cũ.

Câu nói đặc trưng: Tôi biết là dùng có lợi đấy, nhưng tôi không thể dùng nó, vì ....

4) Giai đoạn kiểm nghiệm: Bạn đã khá rành cách dùng div, và bạn bắt đầu sáng tạo và thử nghiệm những kiến thức của mình, (do đó site của bạn làm ra trong giai đoạn này khá tệ

hại http://my.opera.com/community/graphics/smilies/frown.gif và các bạn nên nhanh chóng chuyển sang giai đoạn sau

http://my.opera.com/community/graphics/smilies/frown.gif nếu ko muốn khách hàng phiền lòng)

Câu nói đặc trưng: CSS? Oh, Yes! Tôi luôn dùng div để tạo layout.

5) Giai đoạn chie^m nghiệm:Sau 1 quá trình dài đọc sách, đọc những bài viết của người khác và tự suy nghĩ rất nhiều cũng như bàn luận nhiều về CSS, bạn đã có những hiểu biết sâu sắc về CSS. Lúc này bạn đã có thể dùng CSS vào công việc của mình và biết được bạn nên dùng cái gì, vì sao (tất nhiên lúc đó bạn ko còn hỏi những câu như là: vì sao div dùng cho layout tốt hơn table hoặc là lúc nào dùng %, lúc nào dùng pixel). Lúc này bạn cũng đã làm quen với các hack và filter của CSS (và cũng chỉ nên làm quen với chúng ở giai đoạn này - ý kiến của tôi)

Cấu nổi đặc trưng: Tôi dùng CSS để tạo layout, nó tốt hơn dùng table vì những điểm sau:

6) Giai đoạn sáng tạo:Ok, bây giờ thì bạn là CSS master roài, đối với bạn thì "biết how to do it & why to do it" vẫn chưa đủ. Bạn bắt đầu đi sâu hơn vào bản chất CSS, thậm chí bạn dành thời gian đọc toàn bộ W3C specification về CSS, và bạn biết rõ thuộc tính CSS nào được hỗ trợ bởi những trình duyệt nào. Bạn bắt đầu biết làm những việc mà ít người nghĩ rằng có thể làm được với CSS (http://www.cssplay.co.uk/) và có thể tham gia vào dự án chuẩn web (http://webstandards.org/). Thậm chí khi ai đó thấy lỗi trên site của bạn thì bạn cũng có thể giải thích tại sao.

Câu nói đặc trưng: Phiến bản nào của CSS? À, tôi làm đấy. Cậu đọc quyển xxxCSSyyy của tôi chưa?

Bạn đang ở giai đoạn nào? (bản thân tôi ở giai đoạn 2 thôi. Ngạc nhiên chưa http://my.opera.com/community/graphics/smilies/bigsmile.gif). Mà hình như học bất kỳ cái gì cũng trải qua 6 giai đoạn thế này nhỉ??? He he

Tuy nhiên dù ở giai đoạn nào đi nữa thì bạn cũng nên biết Ctrl+F , viewsoure và google, nếu không thì tôi cũng bótay ko biết nên nói gì với bạn nữa.

# 8 THỦ THUẬT VỀ CSS

Bài này mình copy từ quantrimang

(http://www.quantrimang.com/view.asp?Cat\_ID=3&Cat\_Sub\_ID=0&news\_id=566), còn chẳng rõ bên quantrimang dịch từ nguồn nào http://my.opera.com/community/graphics/smilies/bigsmile.gif (hay họ tự viết)

Tóm tắt:

01. Các ví dụ mẫu http://www.bluerobot.com/web/layouts (http://www.bluerobot.com/web/layouts) http://www.thenoodleincident.com (http://www.thenoodleincident.com/)

http://www.glish.com/css (http://www.glish.com/css)

02. Lưu tâm tới các trình duyệt cũ http://www.alistapart.com/stories/journey/5.html (http://www.alistapart.com/stories/journey/5.html) -> dead link, nhưng dù sao trang alistapart cũng là một trang đáng đọc.

03. Thủ thuật với mô hình hiển thị dạng hộp (box-model) http://www.tantek.com/CSS/Examples/boxmodelhack.html (http://www.tantek.com/CSS/Examples/boxmodelhack.html)

04. Điều khiển in (đúng ra là: "điều chỉnh font"???)

05. Chuyển đổi giữa các mẫu định dạng (đúng ra là: "dùng font thay thế"???) http://www.alistapart.com/stories/alternate/ (http://www.alistapart.com/stories/alternate/)

06. Thêm đường viền

07. Đoạn định dạng (đoạn) kiểu tạp chí

08. Tuỳ biến với hình nền (bằng cách dùng CSS).

## 8 ĐIỀU MỚI LẠ CHO NGƯỜI MỚI HỌC CSS

Vài điều mà các bạn quen thiết kế web chỉ bằng HTML có thể không biết về CSS: (những câu hỏi trong ngoặc vuông có thể khó hiểu, khó trả lời hoặc mơ hồ với bạn thì bạn có thể trả lời sau)

div và span (đặc biệt là div) là 2 loại tag thường được dùng trong trang web sử dụng CSS.

tag div và span có thể coi như những tag ngoại lệ. Trong khi những tag khác mỗi tag đều có 1 chức năng riêng (như tag h1, h2..tạo tiêu đề, tag b tạo chữ đậm ...) thì tag div và span lại có cách dùng rất linh động.

bạn có thể thay đổi chức năng vốn có của các tag bằng cách sử dụng CSS (ví dụ: tag a thường làm link có gạch chân, bạn có thể dùng CSS làm mất gạch chân đó đi; tag b vốn dùng để làm chữ in đậm, bạn có thể dùng CSS để bắt nó làm chữ gạch chân). [bạn có biết làm điều này ko?]

bạn có thể dùng tag div và tag span cùng với CSS để làm thay chức năng của hầu hết các tag khác (ví dụ: dùng tag span để làm chữ in đậm...). [bạn có biết làm điều này ko? tại sao người ta vẫn dùng cac tag khác như h1, h2, p, i, a ... mà ko sử dụng toàn div với lại span]

người ta sử dụng id hoặc class để chỉ định tag sẽ được áp dụng CSS. [bạn có biết lúc nào dùng id, lúc nào dùng class không? nếu chúng ta không khai báo id hoặc class mà vẫn viết css cho id hoặc class đó thì điều gì sẽ xảy ra?]

tag div tuân theo box-model (tạm dịch là mô hình hộp) và thường dùng để tạo layout cho web [sao lại ko dùng tag khác, như tag p, tag ul, chẳng hạn, dù những tag đó cũng tuân theo box-model?]

tag span dùng để định dạng (mark up) cho text [tại sao lại rất ít web dùng tag span?]

chức năng vốn có của các tag và cách làm việc của CSS trên các trình duyệt khác nhau có thể rất khác nhau.

### **CSS CURVES**

Bạn muốn chữ chạy áp theo thân của bức ảnh? nếu ảnh của bạn là hình vuông thì điều đó không có gì khó , chỉ cần dùng thuộc tính foat là xong nhưng nếu ảnh của bạn như hình dưới đây:

Chúng ta phải làm sao? Hôm nay tôi xin giới thiệu với các bạn 1 kỹ thuật Css đơn giản, để dòng chữ có thể chạy theo hình ảnh như trên.

```
Dau tiên ta ca 1 doan ma HTML:
  <div id="wrapper">
  <span id="curve1" class="curve"></span>
  <span id="curve2" class="curve"></span>
  <span id="curve3" class="curve"></span>
  <span id="curve4" class="curve"></span>
  <span id="curve6" class="curve"></span>
  <span id="curve8" class="curve"></span>
  <span id="curve8" class="curve"></span>
```

```
<span id="curve10" class="curve"></span>
<span id="curve11" class="curve"></span>
<span id="curve12" class="curve"></span>
<span id="curve13" class="curve"></span>
</div>
http://i81.photobucket.com/albums/j239/hunterkhoa/12.gif
Và css như sau:
#wrapper {
text-align: left;
width: 500px;
margin: 30px auto;
border: 1px solid #c00;
padding: 15px 15px 5px 15px;
background: url(redCurve.gif) 0 0 no-repeat;
}
/* RED CURVES */
.curve {
float: left;
display: block;
height: 16px;
clear: left;
}
#curve1 { width: 230px; }
#curve2 { width: 225px; }
#curve3 { width: 220px; }
#curve4 { width: 215px; }
#curve5 { width: 210px; }
#curve6 { width: 198px; }
#curve7 { width: 180px; }
#curve8 { width: 170px; }
#curve9 { width: 160px; }
#curve10 { width: 160px; }
#curve11 { width: 140px; }
#curve12 { width: 120px; }
#curve13 { width: 80px; }
```

## CSS STRICKS – MIN – HEIGHT

Khi thiết kế 1 layout đôi lúc chúng ta phải khó chịu với sự khác nhau giữa các trình duyệt, trên FF thế này trên IE thế kia, rồi trên những trình duyệt khác thì....~\_~

Thật là nhức đầu phải không , đôi lúc tôi cũng phải thốt lên răng :" trời đã sanh ra IE ; sao còn tạo ra FF" . ac

Hôm nay tôi xin đề cặp đến 1 vấn đề đó là chèn thuộc tính min-height cho IE, vì khi chúng ta chèn thuộc tính min-height cho 1 <div> nào đó chẳn hạn thì FF và Safari v.v đều hiểu và chạy tốt, còn trên ie thì lại không hiểu đc thuộc tính min-height.

Vậy làm sao đê IE có thể hiểu đc thuộc tính này? rất đơn giản, chúng ta chỉ cần thêm vào 1 đoạn css nhỏ thì IE sẽ hiểu và chạy tốt như những trình duyệt khác ngay:

```
/* Cho tất cả các trình duyệt */
.box {
width:20em;
padding:0.5em;
border:1px solid #000;
min-height:8em;
height:auto;
}
/* Fix cho IE */
/*\*/
```

\* html .box { height: 8em; } /\*\*/ ^\_^ thế là xongXem thêm tại đây (http://www.medizai.com/Medizai,Entry,AT00231Number159ChuyenMucCss.aspx) nguồn www.Medizai.com (http://www.Medizai.com)