



Chương 4C- CSS3

DHTML

JAVA
SCRIPT

LẬP TRÌNH
WEB

PHP

Trình bày: Nguyễn Phú Trường

Giới thiệu CSS

Các loại CSS

Cách dùng CSS

Các CSS cơ bản

Các CSS nâng cao

- ❖ CSS được dùng để điều khiển các kiểu (style) và bố cục (layout) của các trang Web.
- ❖ CSS3 là chuẩn mới nhất của CSS.
- ❖ CSS3 tương thích hoàn toàn với các phiên bản trước đó, nên không phải thay đổi các thiết kế đã có.
- ❖ Các trình duyệt sẽ luôn hỗ trợ CSS2.



- ❖ CSS3 được chia thành nhiều “module”.
- ❖ Một số module CSS3 quan trọng gồm:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Text Effects
 - 2D/3D Transformations
 - Animations
 - Multiple Column Layout
 - User Interface

- ❖ Bộ đặc tả CSS3 vẫn đang được phát triển bởi W3C.
- ❖ Tuy nhiên, nhiều thuộc tính mới của CSS3 đã được cài đặt trong các trình duyệt hiện nay.

- ❖ Với CSS3, chúng ta có thể tạo các
 - Đường viền tròn góc,
 - Thêm bóng nền tới các hộp và
 - Dùng hình ảnh như một đường viền mà không cần dùng chương trình thiết kế như Photoshop.
- ❖ Các thuộc tính tương ứng:
 - border-radius
 - box-shadow
 - border-image

TRÌNH DUYỆT HỖ TRỢ



Property	Browser Support				
border-radius					
box-shadow					
border-image					

- ❖ Để thêm các góc tròn vào CSS2 là sự tinh tế.
- ❖ Chúng ta phải sử dụng các hình ảnh cho mỗi góc.
- ❖ Trong CSS3, tạo các góc tròn rất dễ dàng.
- ❖ Thuộc tính **border-radius** trong CSS3 được dùng để tạo góc tròn

❖ Thí dụ:*div*

{

*border:2px solid;**border-radius:25px;**-moz-border-radius:25px; /* Old Firefox */*

}

❖ Trong CSS3, thuộc tính **box-shadow** được dùng để thêm bóng đổ cho hộp:

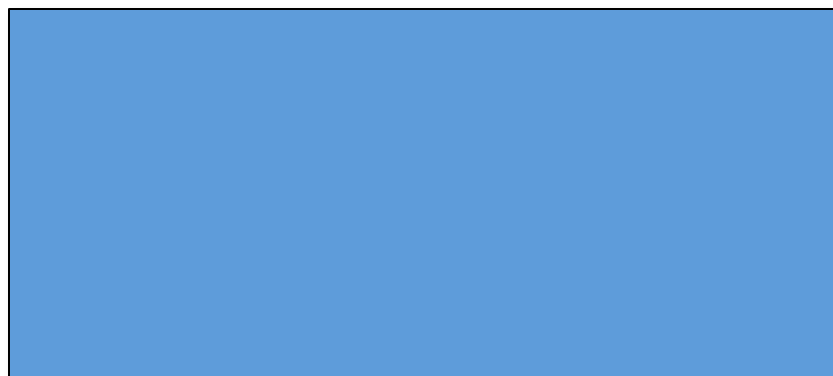
❖ Thí dụ:

- *div*

{

box-shadow: 10px 10px 5px #888888;

}



- ❖ Trong CSS3, thuộc tính **border-image** dùng ảnh để tạo đường viền.
- ❖ *Thí dụ:*
 - *div*
{
border-image:url(border.png) 30 30 round;
-moz-border-image:url(border.png) 30 30 round; / Old Firefox */*
-webkit-border-image:url(border.png) 30 30 round; / Safari and Chrome */*
-o-border-image:url(border.png) 30 30 round; / Opera */*
}

❖ **border-image:** *source slice width outset repeat*;

❖ Viết tắt của các thuộc tính:

Thuộc tính	Giải thích
<u>border-image-source</u>	Đường dẫn chỉ đến ảnh được dùng làm đường viền
<u>border-image-slice</u>	Khoảng cách bên trong của đường viền ảnh
<u>border-image-width</u>	Độ rộng của đường viền ảnh
<u>border-image-outset</u>	Khoảng cách vùng ảnh đường viền mở rộng tới hộp đường viền
<u>border-image-repeat</u>	Đường viền ảnh nên được lặp, tròn hay kéo dẫn

❖ Những giá trị bị bỏ qua được thiết lập giá trị mặc định.

❖ **Chú ý:** dùng các thuộc tính border-image-* để tạo các nút ảnh!

❖ **border-radius: 1-4 length| % / 1-4 length| %;**

❖ 4 giá trị cho mỗi radius theo thứ tự:

- border-top-left-radius
- border-top-right-radius
- border-bottom-right-radius
- border-bottom-left-radius

Giá trị	Mô tả
Length	Định nghĩa dạng của góc
%	Định nghĩa dạng của góc theo %

❖ Nếu bottom-left bị bỏ qua nó sẽ giống top-right.

❖ Nếu bottom-right bị bỏ qua nó sẽ giống top-left.

❖ Nếu top-right bị bỏ qua nó sẽ giống top-left.

❖ Thí dụ 1:

- border-radius:2em; tương đương:
border-top-left-radius:2em;
border-top-right-radius:2em;
border-bottom-right-radius:2em;
border-bottom-left-radius:2em;

❖ Thí dụ 2:

- border-radius: 2em 1em 4em / 0.5em 3em; tương đương
border-top-left-radius: 2em 0.5em;
border-top-right-radius: 1em 3em;
border-bottom-right-radius: 4em 0.5em;
border-bottom-left-radius: 1em 3em;


THUỘC TÍNH BOX-SHADOW



box-shadow: *h-shadow v-shadow blur spread color inset*;

Value	Description
<i>h-shadow</i>	Bắt buộc. Vị trí bóng đổ nằm ngang. Được phép dùng giá trị âm.
<i>v-shadow</i>	Bắt buộc. Vị trí bóng đổ nằm đứng. Được phép dùng giá trị âm.
<i>blur</i>	Tùy chọn. Khoảng cách mờ
<i>spread</i>	Tùy chọn. Kích thước bóng đổ
<i>color</i>	Tùy chọn. Màu bóng đổ.
<i>inset</i>	Tùy chọn. Chuyển bóng đổ từ bóng ngoài (outset) tới bóng trong

- ❖ CSS3 chứa nhiều thuộc tính background mới, cho phép điều khiển các thành phần background.
- ❖ Các thuộc tính background phổ biến:
 - background-size
 - background-origin
- ❖ Cách dùng nhiều ảnh nền.

Property	Browser Support				
<code>background-size</code>					
<code>background-origin</code>					

- ❖ Firefox 3.6 và cũ hơn không hỗ trợ thuộc tính `background-origin`, và yêu cầu bổ sung tiền tố `-moz-` để hỗ trợ thuộc tính `background-size`.
- ❖ Safari 4 yêu cầu tiền tố `-webkit-` để hỗ trợ các thuộc tính background mới.
- ❖ Internet Explorer 9, Firefox 4, Chrome, Safari 5 và Opera hỗ trợ các thuộc tính background mới.

- ❖ Thuộc tính background-size xác định kích thước của ảnh nền.
- ❖ Trước CSS3, kích thước ảnh nền được xác định bởi kích thước thật của ảnh.
- ❖ Trong CSS3, có thể xác định kích thước của ảnh nền mà cho phép chúng ta tái sử dụng lại ảnh nền trong ngữ cảnh khác.
- ❖ Xác định kích thước ảnh bằng pixels hay %.
- ❖ Nếu xác định kích thước ảnh bằng % thì kích thước tương đối so với độ rộng và chiều cao của phần tử cha.

❖ Thí dụ 1: định lại kích thước

- div

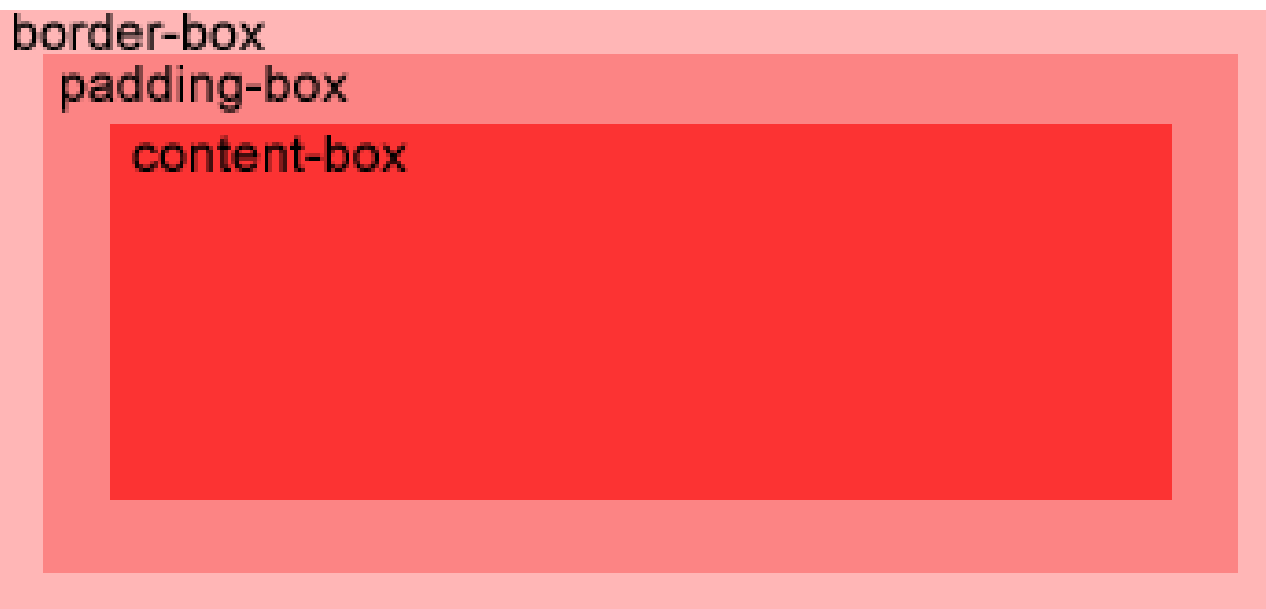
```
{
background:url(img_flwr.gif);
-moz-background-size:80px 60px; /* Old Firefox */
background-size:80px 60px;
background-repeat:no-repeat;
}
```

❖ Thí dụ 2: kéo dẫn kích thước ảnh để điền toàn bộ vùng

- div

```
{
background:url(img_flwr.gif);
-moz-background-size:100% 100%; /* Old Firefox */
background-size:100% 100%;
background-repeat:no-repeat;
}
```

- ❖ Xác định vùng vị trí của các ảnh nền.
- ❖ Ảnh nền có thể được đặt trong content-box, padding-box, hay border-box area.



- ❖ CSS3 cho phép dùng nhiều ảnh nền cho một phần tử.
- ❖ Thí dụ: thiết lập 2 ảnh nền cho phần tử body
 - body
 - {
 - background-image:url(img_flwr.gif),url(img_tree.gif);
 - }

❖ Thí dụ: xác định vùng sơn của nền

- div

```
{
background-color:yellow;
background-clip:content-box;
-webkit-background-clip:content-box; /* Safari */
}
```

- ❖ Thuộc tính background-clip được hỗ trợ bởi IE9+, Firefox 4+, Opera, và Chrome.
- ❖ Safari hỗ trợ một biến thể, thuộc tính **-webkit-background-clip**.
- ❖ Cú pháp:
 - ***background-clip: border-box/padding-box/content-box;***

Giá trị	Mô tả
border-box	Nền được cắt rời tới border box
padding-box	Nền được cắt rời tới padding box
content-box	Nền được cắt rời tới content box

❖ CSS3 chứa một số đặc điểm văn bản mới.

- text-shadow
- word-wrap

❖ Hỗ trợ trình duyệt

Property	Browser Support				
text-shadow					
word-wrap					

- ❖ Trong CSS3, thuộc tính text-shadow cung cấp bóng nền tới văn bản

Text shadow effect!

- ❖ Có thể thiết lập bóng nền ngang, đứng, khoảng cách mờ và màu của bóng đổ
- ❖ Thí dụ:
 - h1
{
text-shadow: 5px 5px 5px #FF0000;
}



- ❖ Nếu một từ quá dài so với vùng hiển thị, nó sẽ mở rộng ra ngoài vùng hiển thị đó.

This paragraph
contains a very long
word:
thisisaveryveryveryveryveryverylongword.
The long word will
break and wrap to the
next line.

This paragraph
contains a very long
word:
thisisaveryveryveryver
yveryverylongword.
The long word will
break and wrap to the
next line.

- ❖ Trong CSS3, thuộc tính **word-wrap** bắt buộc xuống dòng ngay cả phải chia từ.
- ❖ Thí dụ:
 - **p {word-wrap:break-word;}**

- ❖ Trước CSS3, người thiết kế phải sử dụng font đã được cài trên máy tính của người dùng.
- ❖ Với CSS3, người thiết kế web có thể sử dụng bất cứ font gì mình muốn
- ❖ Khi chúng ta tìm một font mong muốn, chứa tập tin font trên webserver và font này được tự tải về cho người dùng khi cần
- ❖ Các font sở hữu được định nghĩa trong quy tắc @font-face

Property	Browser Support				
<code>@font-face</code>					

- ❖ Firefox, Chrome, Safari, và Opera hỗ trợ các font loại .ttf (True Type Fonts) và .otf (OpenType Fonts).
- ❖ Internet Explorer 9+ hỗ trợ qui tắc `@font-face` mới, nhưng nó chỉ hỗ trợ các font loại .eot (Embedded OpenType).
- ❖ **Chú ý:** Internet Explorer 8 và các phiên bản trước đó không hỗ trợ qui tắc `@font-face` mới.

- ❖ Trong qui tắc @font-face mới chúng ta phải định nghĩa một tên cho font (thí dụ myFirstFont) và sau đó tham chiếu tới tập tin font.
- ❖ Để dùng font cho phần tử HTML, tham chiếu tên của font (myFirstFont) qua thuộc tính font-family






```
❖ <style>
  @font-face
  {
    font-family: myFirstFont;
    src: url('Sansation_Light.ttf'),
         url('Sansation_Light.eot'); /* IE9+ */
  }
  div
  {
    font-family:myFirstFont;
  }
</style>
```

- ❖ Phải dùng một qui tắc `@font-face` chứa bộ mô tả để làm đậm văn bản.
- ❖ Thí dụ:
 - `@font-face`
{
font-family: myFirstFont;
src: url('Sansation_Bold.ttf'),
url('Sansation_Bold.eot'); /* IE9+ */
font-weight:bold;
}
 - Tập tin "Sansation_Bold.ttf" là một tập tin font khác chứa ký tự in đậm cho font Sansation.

CSS3 2D TRANSFORM



- ❖ Với CSS3 transform, chúng ta có thể di chuyển, thay đổi kích thước, xoay và kéo dãn các phần tử.
- ❖ Một transform là một hiệu ứng để thay đổi hình dáng, kích thước và vị trí.
- ❖ Chúng ta có thể biến đổi các phần tử dùng biến đổi 2D hay 3D.

Property	Browser Support				
transform					

❖ Thí dụ:

- div
 - {
 - transform: rotate(30deg);
 - ms-transform: rotate(30deg); /* IE 9 */
 - webkit-transform: rotate(30deg); /* Safari and Chrome */
 - o-transform: rotate(30deg); /* Opera */
 - moz-transform: rotate(30deg); /* Firefox */
 - }

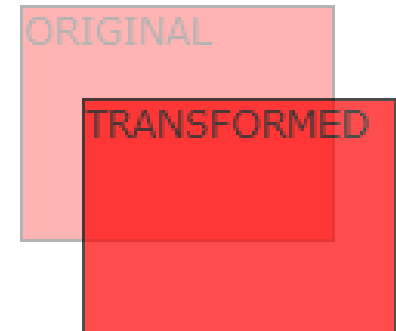
TRANSLATE()



❖ Với phương thức `translate()`, phần tử di chuyển từ vị trí hiện hành của nó, phụ thuộc các tham số được cho đối với vị trí bên trái trục-X và ở trên trục-Y.

❖ Thí dụ:

- `div`
{
 `transform: translate(50px,100px);`
 `-ms-transform: translate(50px,100px); /* IE 9 */`
 `-webkit-transform: translate(50px,100px); /* Safari and Chrome */`
 `-o-transform: translate(50px,100px); /* Opera */`
 `-moz-transform: translate(50px,100px); /* Firefox */`
}

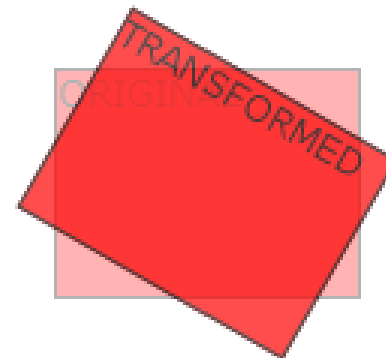


ROTATE()



- ❖ Với phương thức rotate(), phần tử quay theo chiều kim đồng hồ một góc được cho.
- ❖ Giá trị âm, phần tử quay ngược chiều kim đồng hồ
- ❖ Thí dụ:

- div
 - {
 - transform: rotate(30deg);
 - ms-transform: rotate(30deg); /* IE 9 */
 - webkit-transform: rotate(30deg); /* Safari and Chrome */
 - o-transform: rotate(30deg); /* Opera */
 - moz-transform: rotate(30deg); /* Firefox */
 - }



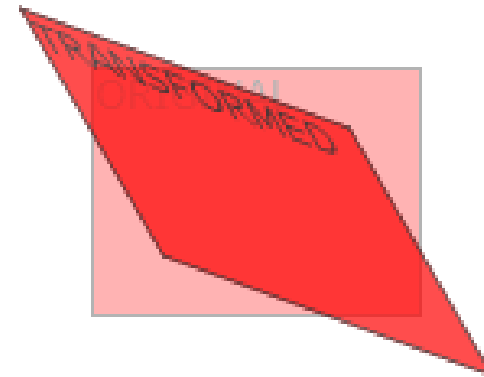
SKEW()



❖ Với phương thức `skew()`, phần tử lật một góc được cho, phụ thuộc vào tham số được cho theo trục ngang trục-X và đứng trục-Y.

❖ Thí dụ:

- `div`
{
 `transform: skew(30deg,20deg);`
 `-ms-transform: skew(30deg,20deg); /* IE 9 */`
 `-webkit-transform: skew(30deg,20deg); /* Safari and Chrome */`
 `-o-transform: skew(30deg,20deg); /* Opera */`
 `-moz-transform: skew(30deg,20deg); /* Firefox */`
}



MATRIX()



- ❖ Phương thức `matrix()` kết hợp tất cả phương thức biến đổi 2D thành một phương thức.
- ❖ Có 6 tham số, chứa các hàm toán, cho phép: xoay, thay đổi kích thước, di chuyển và lật.
- ❖ Thí dụ: quay 30 độ

```
div
{
transform:matrix(0.866,0.5,-0.5,0.866,0,0);
-ms-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* IE 9 */
-moz-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* Firefox */
-webkit-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* Safari and
Chrome */
-o-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* Opera */
}
```



3D TRANSFORMS



- ❖ CSS3 cho phép định dạng các phần tử dùng biến đổi 3D.
- ❖ rotateX()
- ❖ rotateY()

2D
rotate

3D
rotate

Property	Browser Support				
<code>transform</code>					

ROTATEX()



❖ Với rotateX(), phần tử quay xung quanh trục X một góc được cho.

❖ Thí dụ:

❖ div

```
{  
transform: rotateX(120deg);  
-webkit-transform: rotateX(120deg); /* Safari and  
Chrome */  
-moz-transform: rotateX(120deg); /* Firefox */  
}
```



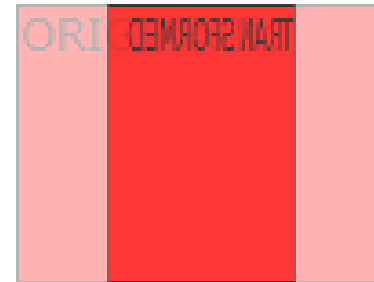
ROTATEY()





❖ Với rotateY(), phần tử quay xung quanh trục Y một góc được cho.

❖ Thí dụ:

- div
{
transform: rotateY(130deg);
-webkit-transform: rotateY(130deg); /* Safari and Chrome */
-moz-transform: rotateY(130deg); /* Firefox */
}



- ❖ Với CSS3, chúng ta có thể thêm hiệu ứng khi thay đổi một kiểu này tới một kiểu khác mà không dùng Flash hay Javascript.

Property	Browser Support				
<code>transition</code>					

- ❖ CSS3 transitions là những hiệu ứng để một phần tử thay đổi dần từ một kiểu này tới một kiểu khác.
- ❖ Để thực hiện, chúng ta phải xác định hai yếu tố:
 - Xác định thuộc tính CSS muốn thêm hiệu ứng tới
 - Xác định khoảng thời gian của hiệu ứng.
- ❖ Thí dụ: hiệu ứng chuyển trên width, thời gian 2s
 - ```
div
{
transition: width 2s;
-moz-transition: width 2s; /* Firefox 4 */
-webkit-transition: width 2s; /* Safari and Chrome */
-o-transition: width 2s; /* Opera */
}
```











### ❖ Thí dụ 1: xác định hover cho phần tử div

- div:hover  
{  
width:300px;  
}

### ❖ Thí dụ 2: thực hiện nhiều thay đổi

- div  
{  
**transition: width 2s, height 2s, transform 2s;**  
-moz-transition: width 2s, height 2s, -moz-transform 2s;  
-webkit-transition: width 2s, height 2s, -webkit-transform 2s;  
-o-transition: width 2s, height 2s, -o-transform 2s;  
}

- ❖ Với CSS3, chúng ta có thể tạo hoạt hình, để thay thế các hình động, hoạt hình Flash và Javascript.
- ❖ Để tạo hoạt hình trong CSS3, chúng ta dùng qui tắc @keyframes.
- ❖ Xác định một CSS style bên trong qui tắc @keyframes và hoạt hình sẽ thay đổi dần dần từ kiểu hiện hành tới một kiểu mới.

| Property   | Browser Support                                                                     |                                                                                     |                                                                                       |                                                                                       |                                                                                       |
|------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| @keyframes |  |  |  |  |  |
| animation  |  |  |  |  |  |

❖ @keyframes myfirst

```
{
 from {background: red;}
 to {background: yellow;}
}
```

@-moz-keyframes myfirst /\* Firefox \*/

```
{
 from {background: red;}
 to {background: yellow;}
}
```

```
❖ @-webkit-keyframes myfirst /* Safari and Chrome */
{
 from {background: red;}
 to {background: yellow;}
}

@-o-keyframes myfirst /* Opera */
{
 from {background: red;}
 to {background: yellow;}
}
```

- ❖ Khi hoạt hình được tạo trong @keyframe, liên kết tới một bộ chọn, ngược lại hoạt hình sẽ không có tác dụng.
- ❖ Liên kết hoạt hình tới bộ chọn bằng cách xác định ít nhất hai thuộc tính CSS animation:
  - Xác định tên của hoạt hình
  - Xác định khoảng thời gian
- ❖ Thí dụ: liên kết hoạt hình myfirst tới div, thời gian 5 s
  - div

```
{
animation: myfirst 5s;
-moz-animation: myfirst 5s; /* Firefox */
-webkit-animation: myfirst 5s; /* Safari and Chrome */
-o-animation: myfirst 5s; /* Opera */
}
```



# ANIMATION LÀ GÌ?


















- ❖ Hoạt hình là một hiệu ứng để thay đổi dần từ một kiểu này tới một kiểu khác.
- ❖ Có thể thay đổi nhiều kiểu nhiều lần.
- ❖ Xác định khi nào thay đổi xảy ra bằng % hay từ khóa “from” và “to”, giống như 0% và 100%.
- ❖ 0% là bắt đầu hoạt hình và 100% là khi hoạt hình hoàn thành.
- ❖ Để hỗ trợ trình duyệt tốt nhất luôn định nghĩa cả hai bộ chọn 0% và 100%.

- ❖ Thí dụ: thay đổi màu nền khi animation là 25%, 50%, và khi animation kết thúc hoàn toàn 100%:
- ❖ @keyframes myfirst

```
{
0% {background: red;}
25% {background: yellow;}
50% {background: blue;}
100% {background: green;}
}
```

- ❖ Với CSS3, có thể tạo nhiều cột để định dạng văn bản như trong báo!
- ❖ Có thuộc tính tạo nhiều cột :
  - column-count
  - column-gap
  - column-rule

| Property     | Browser Support                                                                     |                                                                                     |                                                                                     |                                                                                       |                                                                                       |
|--------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| column-count |   |   |   |   |   |
| column-gap   |  |  |  |  |  |
| column-rule  |  |  |  |  |  |

- ❖ Thuộc tính column-count xác định số lượng cột mà một phần tử muốn được chia
- ❖ Thí dụ: để tạo 3 cột
  - `div`
    - {
    - `-moz-column-count:3; /* Firefox */`
    - `-webkit-column-count:3; /* Safari and Chrome */`
    - `column-count:3;`
    - }

- ❖ Thuộc tính column-gap xác định khoảng trống giữa các cột:
- ❖ Thí dụ: xác định 40 pixels giữa các cột
  - `div`
    - {
    - `-moz-column-gap:40px; /* Firefox */`
    - `-webkit-column-gap:40px; /* Safari and Chrome */`
    - `column-gap:40px;`
    - }

- ❖ Thuộc tính column-rule thiết lập độ rộng, kiểu và màu của qui tắc giữa các cột.
- ❖ Thí dụ:
  - `div`
    - {
    - `-moz-column-rule:3px outset #ff00ff; /* Firefox */`
    - `-webkit-column-rule:3px outset #ff00ff; /* Safari and Chrome */`
    - `column-rule:3px outset #ff00ff;`
    - }