

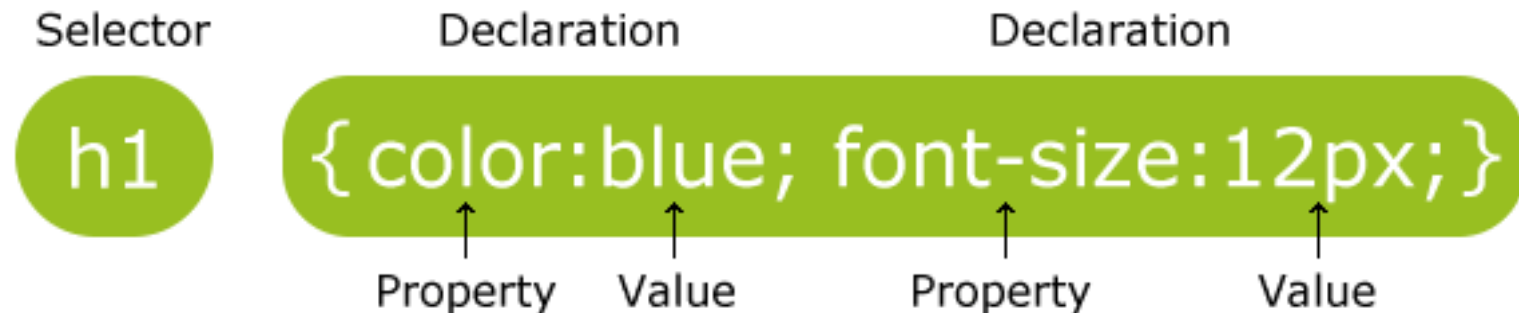
# CSS

**CSS = Cascading Style Sheets**, tập tin định kiểu theo tầng

CSS dùng để định nghĩa phong cách cho trang web

CSS tiết kiệm thời gian định nghĩa phong cách

CSS được lưu trong tập tin .CSS



# CSS Opacity

Thuộc tính **opacity** đặc tả tính trong suốt của 1 phần tử.

Thuộc tính **opacity** có giá trị từ 0.0 đến 1.0. Giá trị càng thấp, đối tượng càng mờ (trong suốt).



opacity 0.2



opacity 0.5



opacity 1  
(default)

# CSS Opacity

## Hình ảnh Transparent

```
<style>
```

```
img {
```

```
    opacity: 0.5;
```

```
    filter: alpha(opacity=50); /* For IE8 and earlier */
```

```
}
```

```
</style>
```

```
...
```

```

```



opacity 0.5

# CSS Opacity

## Hiệu ứng Transparent Hover

Thuộc tính **opacity** và selector **:hover** dùng để thay đổi độ opacity khi rê chuột lên hình.



# CSS Opacity

## Hiệu ứng Transparent Hover

```
<style>
```

```
img {
```

```
    opacity: 0.5;
```

```
    filter: alpha(opacity=50); /* For IE8 and earlier */
```

```
}
```

```
img:hover {
```

```
    opacity: 1.0;
```

```
    filter: alpha(opacity=100); /* For IE8 and earlier */
```

```
}
```

```
</style>
```

# CSS Opacity

## Hiệu ứng Transparent Hover

```

```

```

```

```

```

## Image Transparency

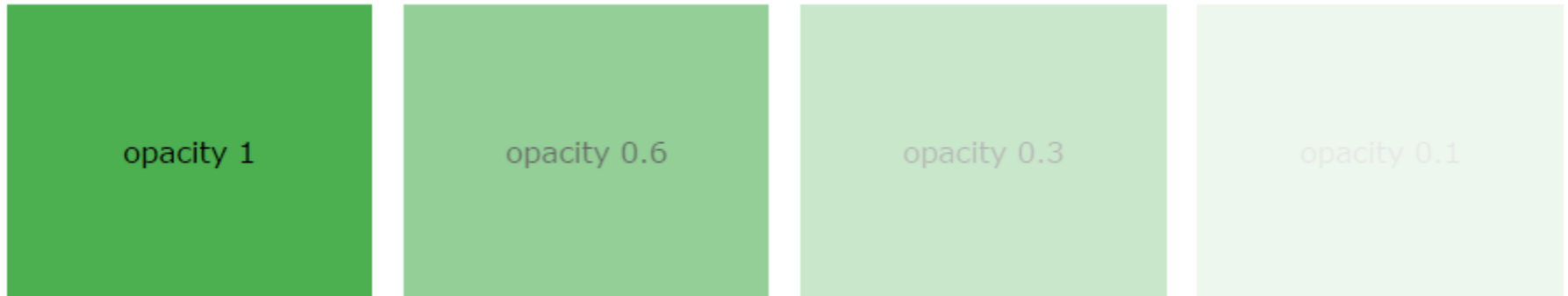
The opacity property is often used together with the :hover selector to change the opacity on mouse-over:



# CSS Opacity

## Transparent Box

Khi dùng thuộc tính opacity để làm mờ nền 1 phần tử, tất cả các phần tử con của nó cũng mờ theo.



# CSS Opacity

## Transparent Box

```
div {  
    background-color: #4CAF50; padding: 10px;  
}  
div.first {  
    opacity: 0.1; filter: alpha(opacity=10); /* For IE8 and earlier */  
}  
div.second {  
    opacity: 0.3; filter: alpha(opacity=30); /* For IE8 and earlier */  
}  
div.third {  
    opacity: 0.6; filter: alpha(opacity=60); /* For IE8 and earlier */  
}
```



# CSS Opacity

## Transparency dùng RGBA

Chúng ta có thể dùng transparent bằng các giá trị màu **RBGA**.

Một giá trị màu RBGA đặc tả bởi:

**rgba(red, green, blue, *alpha*)**



# CSS Opacity

## Transparency dùng RGBA

```
div {  
    background: rgb(76, 175, 80); padding: 10px;  
}  
div.first {  
    background: rgba(76, 175, 80, 0.1);  
}  
div.second {  
    background: rgba(76, 175, 80, 0.3);  
}  
div.third {  
    background: rgba(76, 175, 80, 0.6);  
}
```

# CSS Opacity


## Transparency dùng RGBA

```
<div class="first"><p>10% opacity</p></div>
```

```
<div class="second"><p>30% opacity</p></div>
```

```
<div class="third"><p>60% opacity</p></div>
```

```
<div><p>default</p></div>
```



10% opacity

30% opacity

60% opacity

default

# CSS Opacity

## Văn bản trong Transparent Box

Tạo 1 phần tử `<div>` (`class="background"`) với hình nền và đường viền. Sau đó tạo `<div>` (`class="transbox"`) khác bên trong thẻ `<div>` đầu tiên. Thẻ `<div class="transbox">` chứa màu nền, viền và transparent. Bên trong thẻ `<div>` transparent này, thêm 1 số văn bản.



# CSS Opacity

## Văn bản trong Transparent Box

```
div.background {  
    background: url(klematis.jpg) repeat; border: 2px solid black;  
}  
div.transbox {  
    margin: 30px;background-color: #ffffff;  
    border: 1px solid black;opacity: 0.6;  
    filter: alpha(opacity=60); /* For IE8 and earlier */  
}  
div.transbox p {  
    margin: 5%;font-weight: bold;color: #000000;  
}
```

# CSS Opacity

## Văn bản trong Transparent Box

```
<div class="background">
```

```
  <div class="transbox">
```

```
    <p>This is some text that is placed in the transparent  
box.</p>
```

```
  </div>
```

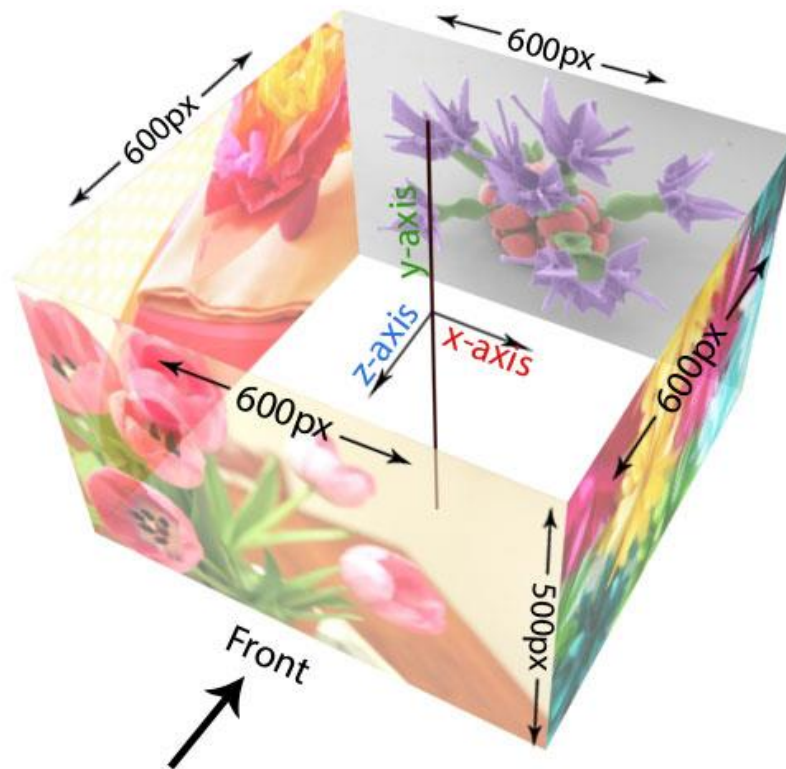
```
</div>
```

# CSS3

## Khái niệm

CSS3 là chuẩn mới nhất của CSS.

CSS3 hoàn toàn tương thích với các phiên bản CSS trở về trước.



# CSS3

## Góc bo tròn

Thuộc tính **border-radius** dùng để bo tròn các góc của 1 phần tử.

```
#rcorners1 {  
    border-radius: 25px;  
    background: #73AD21;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}
```



Rounded corners!

```
<p id="rcorners1">Rounded corners!</p>
```



# CSS3

## Góc bo tròn

Có thể bo tròn ở các góc khác nhau

## Bo tròn 4 góc

`border-radius: 15px 50px 30px 5px;`



# CSS3

## Góc bo tròn

Có thể bo tròn ở các góc khác nhau

### Bo tròn 3 góc

`border-radius: 15px 50px 30px;`



# CSS3

## Ảnh viền

Dùng thuộc tính **border-image**, chúng ta có thể dùng 1 hình ảnh làm đường viền bao quanh 1 phần tử.

Thuộc tính gồm 3 phần: hình làm đường viền, nơi để chia hình, định nghĩa phần lặp lại.



Here, the middle sections of the image are repeated to create the border.

# CSS3

## Ảnh viền

<style>

#borderimg {

border: 10px solid transparent;

padding: 15px;

-webkit-border-image: url(border.png) 30 round; /\* Safari  
3.1-5 \*/

-o-border-image: url(border.png) 30 round; /\* Opera 11-  
12.1 \*/

border-image: url(border.png) 30 round;

}

</style>

# CSS3

## Ảnh viền

<style>

**#borderimg {**

border: 10px solid transparent;

padding: 15px;

-webkit-border-image: url(border.png) 30 round; /\* Safari 3.1-5 \*/

-o-border-image: url(border.png) 30 round; /\* Opera 11-12.1 \*/  
border-image: url(border.png) 30 round;

}

...

</style>

<p id="**borderimg**">Here, the middle sections of the image are repeated to create the border.</p>

# CSS3

## Ảnh viền – Các giá trị chia ra khác nhau

`border-image: url(border.png) 50 round;`

Example 2:

`border-image: url(border.png) 20% round;`

Example 3:

`border-image: url(border.png) 30% round;`

# CSS3

## Dùng nhiều hình nền trong CSS3

CSS3 cho phép dùng nhiều hình nền cho 1 phần tử, thông qua thuộc tính **background-image**.

Cách 1:

```
#example1 {  
    background-image: url(img_flwr.gif), url(paper.gif);  
    background-position: right bottom, left top;  
    background-repeat: no-repeat, repeat;  
}
```

Cách 2:

```
#example1 {  
    background: url(img_flwr.gif) right bottom no-repeat,  
    url(paper.gif) left top repeat;  
}
```

# CSS3

## Dùng nhiều hình nền trong CSS3

CSS3 cho phép dùng nhiều hình nền cho 1 phần tử, thông qua thuộc tính **background-image**.



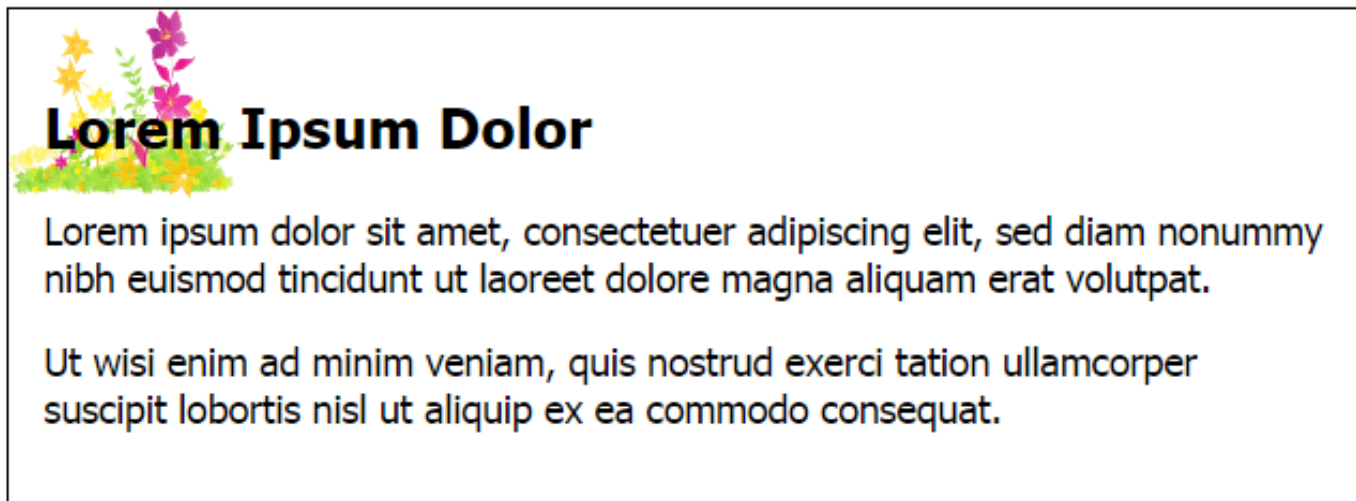


# CSS3

## Kích thước hình nền CSS3

Thuộc tính background-size cho phép điều chỉnh kích thước của các hình nền.

```
#div1 {  
    background: url(img_flower.jpg);  
    background-size: 100px 80px;  
    background-repeat: no-repeat;  
}
```



# CSS3

## Kích thước hình nền

Hai giá trị khác của thuộc tính **background-size** là **cover** và **contain**.

- Từ khóa **contain** sẽ giãn hình lớn nhất có thể (độ rộng và độ cao của hình phải nằm trong vùng nội dung)
- Từ khóa **cover** sẽ giãn hình để phần nội dung được bao phủ bởi hình nền (độ rộng và độ cao của hình sẽ bằng hoặc lớn hơn phần nội dung)

# CSS3

## Kích thước hình nền

```
#div1 {  
    background: url(img_flower.jpg);  
    background-size: contain;  
    background-repeat: no-repeat;  
}
```

```
#div2 {  
    background: url(img_flower.jpg);  
    background-size: cover;  
    background-repeat: no-repeat;  
}
```

# CSS3

## Kích thước hình nền

Using the "contain" keyword:



Using the "cover" keyword:



# CSS3

## Các hiệu ứng bóng

Với CSS3, chúng ta có thể thêm các hiệu ứng vào văn bản và các phần tử.

- **text-shadow** và **box-shadow**



Box Shadow

With CSS3 you can create  
shadow effects!

Hover over me!

# CSS3

## Hiệu ứng bóng văn bản

Dùng thuộc tính text-shadow để thêm độ bóng cho văn bản.

```
h1 { text-shadow: 2px 2px; }
```

**Text shadow effect!**

Thêm màu:

```
h1 { text-shadow: 2px 2px red; }
```

**Text shadow effect!**

# CSS3

## Hiệu ứng bóng văn bản

### Hiệu ứng bóng mờ

```
h1 {  
    text-shadow: 2px 2px 5px red;  
}
```

Text shadow effect!

# CSS3

## Hiệu ứng bóng văn bản

Ví dụ chữ trắng bóng mờ màu đen:

```
h1 {  
    color: white;  
    text-shadow: 2px 2px 4px #000000;  
}
```

Text shadow effect!



# CSS3

## Hiệu ứng đa bóng văn bản

Muốn dùng nhiều bóng, chỉ cần thêm các định nghĩa bóng sau mỗi dấu phẩy ở thuộc tính **text-shadow**.

```
h1 {  
    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;  
}
```

Text shadow effect!

```
h1 {  
    color: white;  
    text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px  
darkblue;  
}
```

Text shadow effect!

# CSS3

## Hiệu ứng đa bóng trên khối hộp

Thuộc tính **box-shadow** để định nghĩa bóng cho các phần tử (khối hộp).

```
<style>
```

```
div {
```

```
    width: 300px; height: 100px;
```

```
    padding: 15px;
```

```
    background-color: yellow;
```

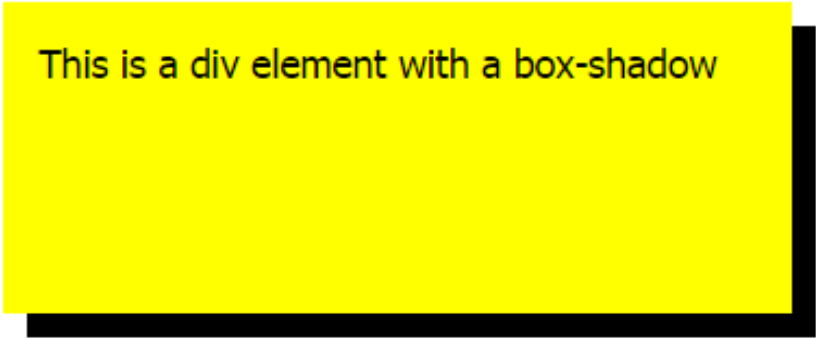
```
    box-shadow: 10px 10px;
```

```
}
```

```
</style>
```

```
...
```

```
<div>This is a div element with a box-shadow</div>
```



This is a div element with a box-shadow

# CSS3

## Hiệu ứng đa bóng trên khối hộp

Thuộc tính **box-shadow** để định nghĩa bóng cho các phần tử (khối hộp).

```
<style>
```

```
div {
```

```
    width: 300px; height: 100px;
```

```
    padding: 15px;
```

```
    background-color: yellow;
```

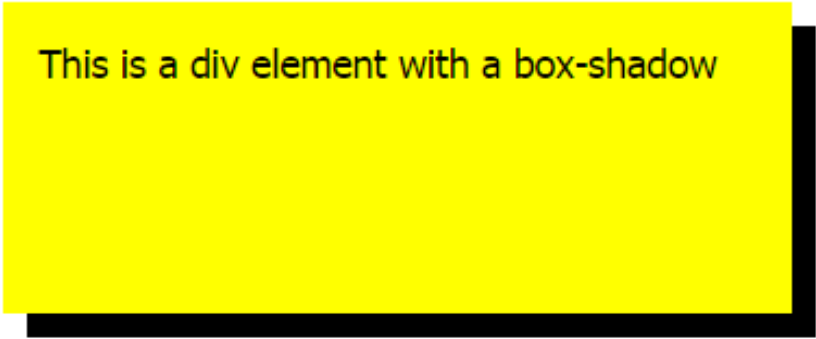
```
    box-shadow: 10px 10px;
```

```
}
```

```
</style>
```

```
...
```

```
<div>This is a div element with a box-shadow</div>
```



This is a div element with a box-shadow

# CSS3

## Hiệu ứng đa bóng trên khối hộp

### Màu cho khối hộp

```
div {  
    box-shadow: 10px 10px grey;  
}
```

This is a yellow <div> element  
with a grey box-shadow

### Hiệu ứng bóng mờ (blur)

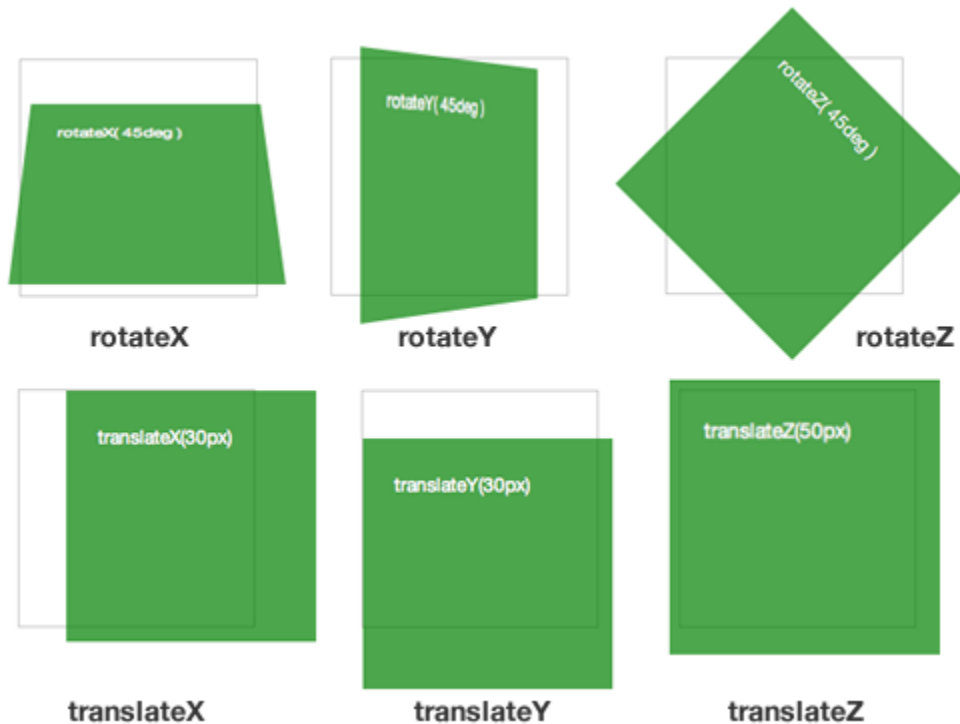
```
div {  
    box-shadow: 10px 10px 5px grey;  
}
```

This is a yellow <div> element  
with a blurred, grey box-shadow

# CSS3 2D Transforms

Các chuyển đổi CSS3 cho phép dịch, xoay, thay đổi kích thước và làm nghiêng các phần tử.

CSS3 hỗ trợ các chuyển đổi **2D** và **3D**.



# CSS3 2D Transforms

Trong phần này, chúng ta sẽ học các phương thức chuyển đổi 2D:

- `translate()`
- `rotate()`
- `scale()`
- `skewX()`
- `skewY()`
- `matrix()`

# CSS3 2D Transforms

## Phương thức translate()

Phương thức translate() cho phép di chuyển 1 phần tử từ vị trí hiện tại của nó đến 1 vị trí khác. (theo tham số trục X và trục Y cho trước)

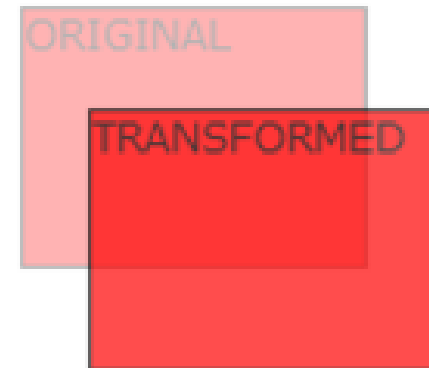
div {

-ms-transform: translate(50px, 100px); /\* IE 9 \*/

-webkit-transform: translate(50px, 100px); /\* Safari \*/

**transform: translate(50px, 100px);**

}



# CSS3 2D Transforms

## Phương thức rotate()

Phương thức này xoay 1 phần tử theo chiều kim đồng hồ hoặc ngược lại tùy theo độ đã cho.

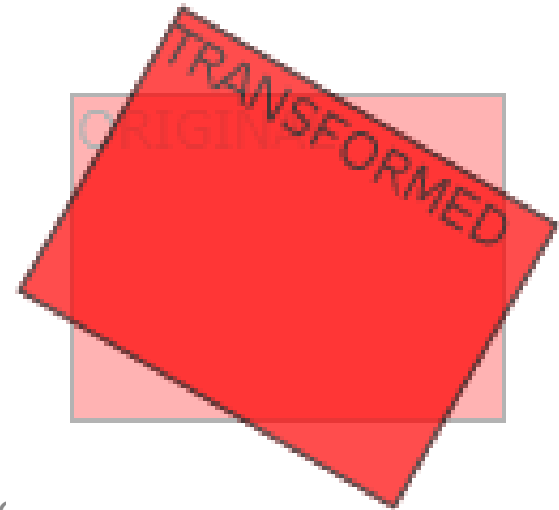
```
div {
```

```
-ms-transform: rotate(20deg); /* IE 9 */
```

```
-webkit-transform: rotate(20deg); /* Safari */
```

```
transform: rotate(20deg);
```

```
}
```





# CSS3 2D Transforms

## Phương thức scale()

Phương thức này phóng to, thu nhỏ phần tử theo kích thước cho trước.

```
div {  
  -ms-transform: scale(2, 3); /* IE 9 */  
  -webkit-transform: scale(2, 3); /* Safari */  
  transform: scale(2, 3);  
}
```

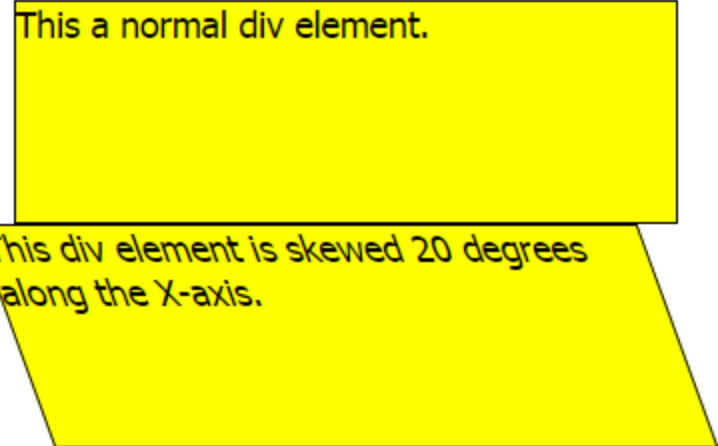


# CSS3 2D Transforms

## Phương thức skewX()

Phương thức này làm nghiêng 1 phần tử theo trục X với 1 góc cho trước.

```
div {  
  -ms-transform: skewX(20deg); /* IE 9 */  
  -webkit-transform: skewX(20deg); /* Safari */  
  transform: skewX(20deg);  
}
```



This a normal div element.

This div element is skewed 20 degrees along the X-axis.

# CSS3 2D Transforms

## Phương thức skewY()

Phương thức này làm nghiêng 1 phần tử theo trục Y với 1 góc cho trước.

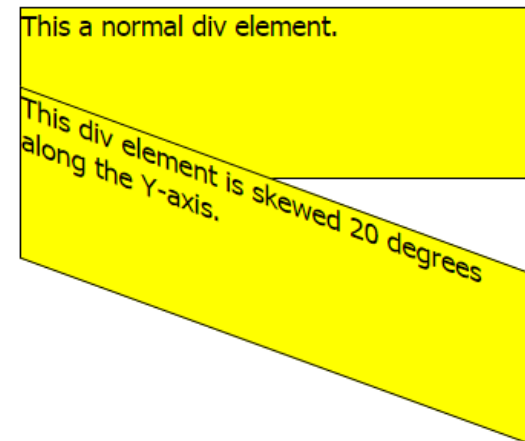
div {

-ms-transform: skewY(20deg); /\* IE 9 \*/

-webkit-transform: skewY(20deg); /\* Safari \*/

transform: skewY(20deg);

}



# CSS3 2D Transforms

## Phương thức `matrix()`

Phương thức `matrix` sẽ kết hợp tất cả các phương thức trên.

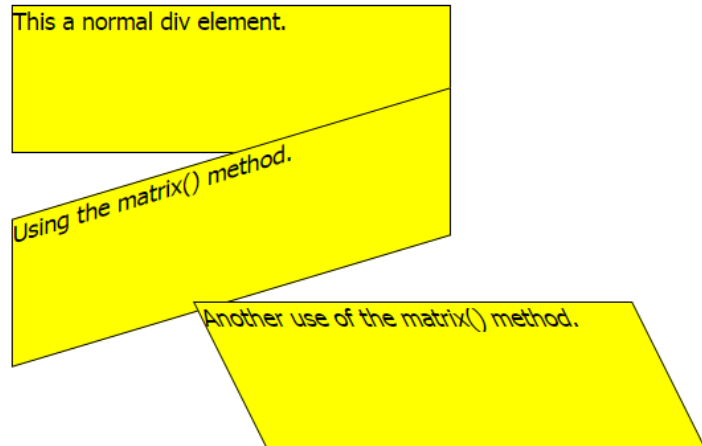
Mô tả phương thức:

`matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())`

# CSS3 2D Transforms

## Phương thức matrix()

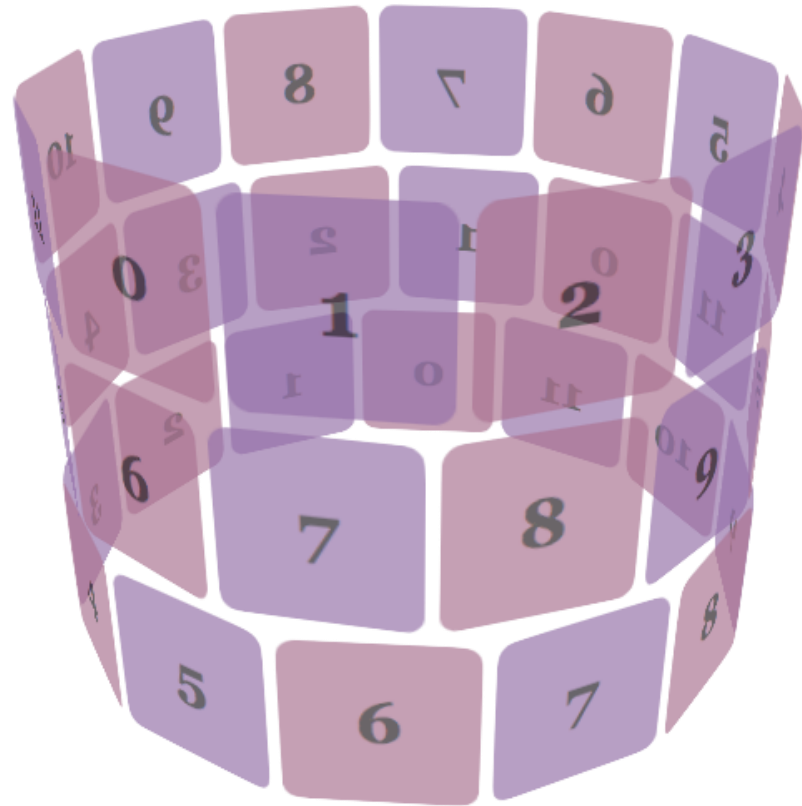
```
div {  
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */  
  -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari  
*/  
  transform: matrix(1, -0.3, 0, 1, 0, 0);  
}
```



# CSS3 3D Transforms

Trong phần này, chúng ta sẽ học các phương thức chính 3D như:

- rotateX()
- rotateY()
- rotateZ()

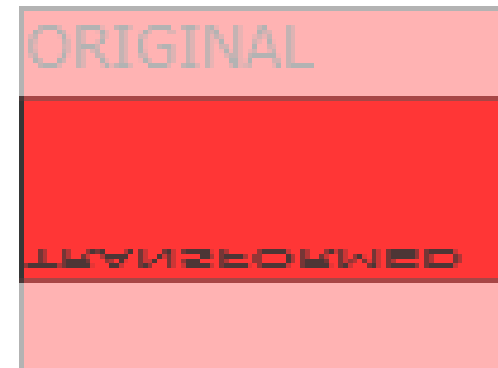


# CSS3 3D Transforms

## Phương thức rotateX()

Phương thức này sẽ dịch 1 phần tử theo trục X với góc cho trước.

```
div {  
    -webkit-transform: rotateX(150deg); /* Safari */  
    transform: rotateX(150deg);  
}
```

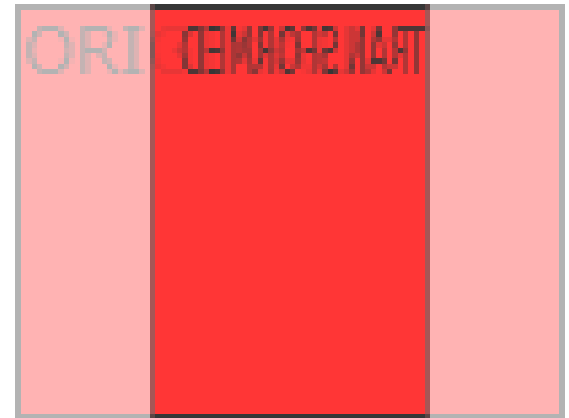


# CSS3 3D Transforms

## Phương thức rotateY()

Phương thức này sẽ dịch 1 phần tử theo trục Y với góc cho trước.

```
div {  
    -webkit-transform: rotateY(150deg); /* Safari */  
    transform: rotateY(150deg);  
}
```



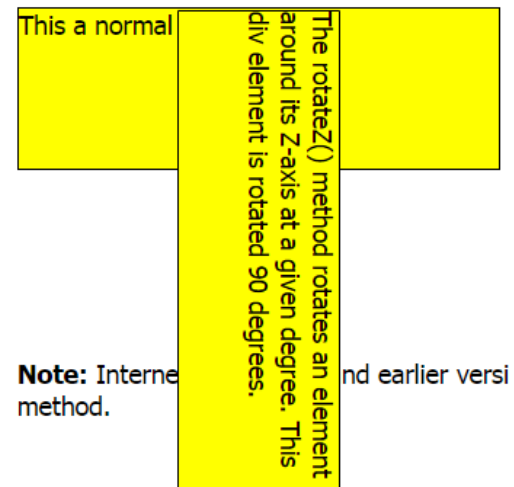


# CSS3 3D Transforms

## Phương thức rotateZ()

Phương thức này sẽ dịch 1 phần tử theo trục Z với góc cho trước.

```
div {  
    -webkit-transform: rotateZ(150deg); /* Safari */  
    transform: rotateZ(150deg);  
}
```



# CSS3 3D Transforms

Các phương thức 3D khác:

- `matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)`
- `translate3d(x,y,z)`, `translateX(x)`, `translateY(y)`, `translateZ(z)`
- `scale3d(x,y,z)`, `scaleX(x)`, `scaleY(y)`, `scaleZ(z)`
- `rotate3d(x,y,z,angle)`, `rotateX(angle)`, `rotateY(angle)`, `rotateZ(angle)`
- `perspective(n)`

# CSS3 Transitions

Các hiệu ứng chuyển tiếp trong CSS cho phép thay đổi giá trị thuộc tính 1 cách mượt mà trong khoảng thời gian cho trước.

Property					
transition	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-delay	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-duration	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-property	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-timing-function	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

# CSS3 Transitions

## Cách sử dụng CSS3 Transitions

Để tạo ra hiệu ứng, cần đặc tả 2 điều:

- Thuộc tính CSS muốn thêm hiệu ứng
- Thời gian hiệu ứng

**Lưu ý:** Nếu thời gian hiệu ứng không được mô tả, hiệu ứng sẽ không xảy ra, bởi vì giá trị mặc định thời gian hiệu ứng là 0 (s).

# CSS3 Transitions

## Cách sử dụng CSS3 Transitions

### Ví dụ:

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition: width 2s; /* Safari */  
    transition: width 2s;  
}  
div:hover {  
    width: 300px;  
}
```



# CSS3 Transitions

Thay đổi các giá trị

Ví dụ:

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition: width 2s; /* Safari */  
    transition: width 2s, height 4s;  
}  
div:hover {  
    width: 300px;  
    height: 300px;  
}
```



# CSS3 Transitions

Thay đổi các giá trị

Ví dụ:

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition: width 2s; /* Safari */  
    transition: width 2s, height 4s;  
}  
div:hover {  
    width: 300px;  
    height: 300px;  
}
```



# CSS3 Transitions

## Đặc tả tốc độ các chuyển tiếp

Thuộc tính **transition-timing-function** sẽ mô tả tốc độ của các hiệu ứng, gồm các giá trị:

- ease
- linear
- ease-in
- ease-out
- ease-in-out
- cubic-bezier(n,n,n,n)



# CSS3 Transitions

## Đặc tả tốc độ các chuyển tiếp

```
#div1 {transition-timing-function: linear;}
```

```
#div2 {transition-timing-function: ease;}
```

```
#div3 {transition-timing-function: ease-in;}
```

```
#div4 {transition-timing-function: ease-out;}
```

```
#div5 {transition-timing-function: ease-in-out;}
```

# CSS3 Transitions

## Độ trễ của hiệu ứng chuyển tiếp

Thuộc tính **transition-delay** mô tả độ trễ (giây) của 1 hiệu ứng chuyển tiếp.

```
div {  
    -webkit-transition-delay: 1s; /* Safari */  
    transition-delay: 1s;  
}
```

# CSS3 Transitions

## Transition + Transformation

**Ví dụ:** mô tả cách kết hợp hiệu ứng chuyển tiếp và chuyển đổi cùng 1 lúc.

```
div {  
    width: 100px; height: 100px; background: red;  
    -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /*  
Safari */  
    transition: width 2s, height 2s, transform 2s;  
}  
div:hover {  
    width: 300px; height: 300px;  
    -webkit-transform: rotate(180deg); /* Safari */  
    transform: rotate(180deg);  
}
```

# CSS3 Transitions

## Transition + Transformation

**Ví dụ:** mô tả cách kết hợp hiệu ứng chuyển tiếp và chuyển đổi cùng 1 lúc.

```
div {  
    width: 100px; height: 100px; background: red;  
    -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /*  
Safari */  
    transition: width 2s, height 2s, transform 2s;  
}  
div:hover {  
    width: 300px; height: 300px;  
    -webkit-transform: rotate(180deg); /* Safari */  
    transform: rotate(180deg);  
}
```

# CSS3 Image

Chúng ta cũng có thể sử dụng thuộc tính **border-radius** làm bo tròn các góc của 1 hình ảnh.

```
img {  
    border-radius: 8px;  
}
```



# CSS3 Image

Chúng ta cũng có thể sử dụng thuộc tính **border-radius** làm bo tròn các góc của 1 hình ảnh.

```
img {  
    border-radius: 50%;  
}
```



# CSS3 Image

Làm khung ảnh bo tròn

```
img {  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    padding: 5px;  
}  

```



# CSS3 Image

Làm khung ảnh bo tròn, rê chuột có bóng

```
a {  
    display: inline-block;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    padding: 5px;  
    transition: 0.3s;  
}  
a:hover {  
    box-shadow: 0 0 2px 1px rgba  
    (0, 140, 186, 0.5);  
}
```





# CSS3 Button

Sử dụng CSS3, chúng ta cũng có thể tạo ra các phong cách nút nhất khác nhau.

