CSS3 Introduction

CSS3 is the latest standard for CSS.

CSS3 is completely backwards-compatible with earlier versions of CSS.

## CSS3 Modules

CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added.

Some of the most important CSS3 modules are:

* Selectors
* Box Model
* Backgrounds and Borders
* Image Values and Replaced Content
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface

Most of the new CSS3 properties are implemented in modern browsers.

# CSS3 Rounded Corners

## CSS3 Rounded Corners

With the CSS3 border-radius property, you can give any element "rounded corners".

## Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| border-radius | 5.0 4.0 -webkit- | 12.0 | 9.0 | 4.0 3.0 -moz- | 5.0 3.1 -webkit- | 10.5 |

## CSS3 border-radius Property

With CSS3, you can give any element "rounded corners", by using the border-radius property.

Here are three examples:

1. Rounded corners for an element with a specified background color:

Rounded corners!

2. Rounded corners for an element with a border:

Rounded corners!

3. Rounded corners for an element with a background image:

Rounded corners!

Here is the code:

### Example

#rcorners1 {  
    border-radius: 25px;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}  
  
#rcorners2 {  
    border-radius: 25px;  
    border: 2px solid #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}  
  
#rcorners3 {  
    border-radius: 25px;  
    background: url(paper.gif);  
    background-position: left top;  
    background-repeat: repeat;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_border-radius)

|  |  |
| --- | --- |
| **Note** | **Tip:** The border-radius property is actually a shorthand property for the border-top-left-radius,border-top-right-radius, border-bottom-right-radius and border-bottom-left-radiusproperties. |

## CSS3 border-radius - Specify Each Corner

If you specify only one value for the border-radius property, this radius will be applied to all 4 corners.

However, you can specify each corner separately if you wish. Here are the rules:

* **Four values:** first value applies to top-left, second value applies to top-right, third value applies to bottom-right, and fourth value applies to bottom-left corner
* **Three values:** first value applies to top-left, second value applies to top-right and bottom-left, and third value applies to bottom-right
* **Two values:** first value applies to top-left and bottom-right corner, and the second value applies to top-right and bottom-left corner
* **One value:** all four corners are rounded equally

Here are three examples:

1. Four values - border-radius: 15px 50px 30px 5px:

2. Three values - border-radius: 15px 50px 30px:

3. Two values - border-radius: 15px 50px:

Here is the code:

### Example

#rcorners4 {  
    border-radius: 15px 50px 30px 5px;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}  
  
#rcorners5 {  
    border-radius: 15px 50px 30px;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}  
  
#rcorners6 {  
    border-radius: 15px 50px;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}

You could also create elliptical corners:

### Example

#rcorners7 {  
    border-radius: 50px/15px;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}  
  
#rcorners8 {  
    border-radius: 15px/50px;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;   
}  
  
#rcorners9 {  
    border-radius: 50%;  
    background: #73AD21;  
    padding: 20px;   
    width: 200px;  
    height: 150px;  
}

## CSS3 Rounded Corners Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [border-radius](http://www.w3schools.com/cssref/css3_pr_border-radius.asp) | A shorthand property for setting all the four border-\*-\*-radius properties |
| [border-top-left-radius](http://www.w3schools.com/cssref/css3_pr_border-top-left-radius.asp) | Defines the shape of the border of the top-left corner |
| [border-top-right-radius](http://www.w3schools.com/cssref/css3_pr_border-top-right-radius.asp) | Defines the shape of the border of the top-right corner |
| [border-bottom-right-radius](http://www.w3schools.com/cssref/css3_pr_border-bottom-right-radius.asp) | Defines the shape of the border of the bottom-right corner |
| [border-bottom-left-radius](http://www.w3schools.com/cssref/css3_pr_border-bottom-left-radius.asp) | Defines the shape of the border of the bottom-left corner |

# CSS3 Border Images

## CSS3 Border Images

With the CSS3 border-image property, you can set an image to be used as the border around an element.

## Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| border-image | 16.0 4.0 -webkit- | 12.0 | 11.0 | 15.0 3.5 -moz- | 6.0 3.1 -webkit- | 15.0 11.0 -o- |

## CSS3 border-image Property

The CSS3 border-image property allows you to specify an image to be used instead of the normal border around an element.

The property has three parts:

1. The image to use as the border
2. Where to slice the image
3. Define whether the middle sections should be repeated or stretched

We will use the following image (called "border.png"):



The border-image property takes the image and slices it into nine sections, like a tic-tac-toe board. It then places the corners at the corners, and the middle sections are repeated or stretched as you specify.

**Note:** For border-image to work, the element also needs the border property set!

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

An image as a border!

Here is the code:

### Example

#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;  
}

Here, the middle sections of the image are stretched to create the border:

An image as a border!

Here is the code:

### Example

#borderimg {  
    border: 10px solid transparent;  
    padding: 15px;  
    -webkit-border-image: url(border.png) 30 stretch; /\* Safari 3.1-5 \*/  
    -o-border-image: url(border.png) 30 stretch; /\* Opera 11-12.1 \*/  
    border-image: url(border.png) 30 stretch;  
}

|  |  |
| --- | --- |
| **Note** | **Tip:** The border-image property is actually a shorthand property for the border-image-source,border-image-slice, border-image-width, border-image-outset and border-image-repeatproperties. |

## CSS3 border-image - Different Slice Values

Different slice values completely changes the look of the border:

Example 1:

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;

Here is the code:

### Example

#borderimg1 {  
    border: 10px solid transparent;  
    padding: 15px;  
    -webkit-border-image: url(border.png) 50 round; /\* Safari 3.1-5 \*/  
    -o-border-image: url(border.png) 50 round; /\* Opera 11-12.1 \*/  
    border-image: url(border.png) 50 round;  
}  
  
#borderimg2 {  
    border: 10px solid transparent;  
    padding: 15px;  
    -webkit-border-image: url(border.png) 20% round; /\* Safari 3.1-5 \*/  
    -o-border-image: url(border.png) 20% round; /\* Opera 11-12.1 \*/  
    border-image: url(border.png) 20% round;  
}  
  
#borderimg3 {  
    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;  
}

## CSS3 Border Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [border-image](http://www.w3schools.com/cssref/css3_pr_border-image.asp) | A shorthand property for setting all the border-image-\* properties |
| [border-image-source](http://www.w3schools.com/cssref/css3_pr_border-image-source.asp) | Specifies the path to the image to be used as a border |
| [border-image-slice](http://www.w3schools.com/cssref/css3_pr_border-image-slice.asp) | Specifies how to slice the border image |
| [border-image-width](http://www.w3schools.com/cssref/css3_pr_border-image-width.asp) | Specifies the widths of the border image |
| [border-image-outset](http://www.w3schools.com/cssref/css3_pr_border-image-outset.asp) | Specifies the amount by which the border image area extends beyond the border box |
| [border-image-repeat](http://www.w3schools.com/cssref/css3_pr_border-image-repeat.asp) | Specifies whether the border image should be repeated, rounded or stretched |

# CSS3 Backgrounds

## CSS3 Backgrounds

CSS3 contains a few new background properties, which allow greater control of the background element.

In this chapter you will learn how to add multiple background images to one element.

You will also learn about the following new CSS3 properties:

* background-size
* background-origin
* background-clip

## Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| background-image (with multiple backgrounds) | 4.0 | 12.0 | 9.0 | 3.6 | 3.1 | 11.5 |
| background-size | 4.0 1.0 -webkit- | 12.0 | 9.0 | 4.0 3.6 -moz- | 4.1 3.0 -webkit- | 10.5 10.0 -o- |
| background-origin | 1.0 | 12.0 | 9.0 | 4.0 | 3.0 | 10.5 |
| background-clip | 4.0 | 12.0 | 9.0 | 4.0 | 3.0 | 10.5 |

## CSS3 Multiple Backgrounds

CSS3 allows you to add multiple background images for an element, through the background-image property.

The different background images are separated by commas, and the images are stacked on top of each other, where the first image is closest to the viewer.

The following example has two background images, the first image is a flower (aligned to the bottom and right) and the second image is a paper background (aligned to the top-left corner):

### Example

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

Multiple background images can be specified using either the individual background properties (as above) or thebackground shorthand property.

The following example uses the background shorthand property (same result as example above):

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_background_multiple2)

## CSS3 Background Size

The CSS3 background-size property allows you to specify the size of background images.

Before CSS3, the size of a background image was the actual size of the image. CSS3 allows us to re-use background images in different contexts.

The size can be specified in lengths, percentages, or by using one of the two keywords: contain or cover.

The following example resizes a background image to much smaller than the original image (using pixels):

Original background image:

## Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Resized background image:

## Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Here is the code:

### Example

#div1 {  
    background: url(img\_flower.jpg);  
    background-size: 100px 80px;  
    background-repeat: no-repeat;  
}

The two other possible values for background-size are contain and cover.

The contain keyword scales the background image to be as large as possible (but both its width and its height must fit inside the content area). As such, depending on the proportions of the background image and the background positioning area, there may be some areas of the background which are not covered by the background image.

The cover keyword scales the background image so that the content area is completely covered by the background image (both its width and height are equal to or exceed the content area). As such, some parts of the background image may not be visible in the background positioning area.

The following example illustrates the use of contain and cover:

### Example

#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;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_background-size_contain)

## Define Sizes of Multiple Background Images

The background-size property also accepts multiple values for background size (using a comma-separated list), when working with multiple backgrounds.

The following example has three background images specified, with different background-size value for each image:

### Example

#example1 {  
    background: url(img\_flwr.gif) left top no-repeat, url(img\_flwr.gif) right bottom no-repeat, url(paper.gif) left top repeat;  
    background-size: 50px, 130px, auto;  
}

## Full Size Background Image

Now we want to have a background image on a website that covers the entire browser window at all times.

The requirements are as follows:

* Fill the entire page with the image (no white space)
* Scale image as needed
* Center image on page
* Do not cause scrollbars

The following example shows how to do it; Use the html element (the html element is always at least the height of the browser window). Then set a fixed and centered background on it. Then adjust its size with the background-size property:

### Example

html {  
    background: url(img\_flower.jpg) no-repeat center fixed;   
    background-size: cover;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_background_full)

## CSS3 background-origin Property

The CSS3 background-origin property specifies where the background image is positioned.

The property takes three different values:

* border-box - the background image starts from the upper left corner of the border
* padding-box - (default) the background image starts from the upper left corner of the padding edge
* content-box - the background image starts from the upper left corner of the content

The following example illustrates the background-origin property:

### Example

#example1 {  
    border: 10px solid black;  
    padding: 3px;  
    background: url(img\_flwr.gif);  
    background-repeat: no-repeat;  
    background-origin: content-box;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_background-origin)

## CSS3 background-clip Property

The CSS3 background-clip property specifies the painting area of the background.

The property takes three different values:

* border-box - (default) the background is painted to the outside edge of the border
* padding-box - the background is painted to the outside edge of the padding
* content-box - the background is painted within the content box

The following example illustrates the background-clip property:

### Example

#example1 {  
    border: 10px dotted black;  
    padding: 35px;  
    background: yellow;  
    background-clip: content-box;  
}

## CSS3 Background Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [background](http://www.w3schools.com/cssref/css3_pr_background.asp) | A shorthand property for setting all the background properties in one declaration |
| [background-clip](http://www.w3schools.com/cssref/css3_pr_background-clip.asp) | Specifies the painting area of the background |
| [background-image](http://www.w3schools.com/cssref/pr_background-image.asp) | Specifies one or more background images for an element |
| [background-origin](http://www.w3schools.com/cssref/css3_pr_background-origin.asp) | Specifies where the background image(s) is/are positioned |
| [background-size](http://www.w3schools.com/cssref/css3_pr_background-size.asp) | Specifies the size of the background image(s) |

# CSS3 Colors

## CSS3 Colors

CSS supports color names, hexadecimal and RGB colors.

In addition, CSS3 also introduces:

* RGBA colors
* HSL colors
* HSLA colors
* opacity

## Browser Support

The numbers in the table specify the first browser version that fully supports CSS3 color values/property.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| RGBA, HSL, and HSLA | 4.0 | 12.0 | 9.0 | 3.0 | 3.1 | 10.1 |
| Opacity | 4.0 | 12.0 | 9.0 | 2.0 | 3.1 | 10.1 |

## RGBA Colors

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with: rgba(red, green, blue, alpha). The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

rgba(255, 0, 0, 0.2);

rgba(255, 0, 0, 0.4);

rgba(255, 0, 0, 0.6);

rgba(255, 0, 0, 0.8);

The following example defines different RGBA colors:

### Example

#p1 {background-color: rgba(255, 0, 0, 0.3);}  /\* red with opacity \*/  
#p2 {background-color: rgba(0, 255, 0, 0.3);}  /\* green with opacity \*/  
#p3 {background-color: rgba(0, 0, 255, 0.3);}  /\* blue with opacity \*/

## HSL Colors

HSL stands for Hue, Saturation and Lightness.

An HSL color value is specified with: hsl(hue, saturation, lightness).

1. Hue is a degree on the color wheel (from 0 to 360):
   * 0 (or 360) is red
   * 120 is green
   * 240 is blue
2. Saturation is a percentage value: 100% is the full color.
3. Lightness is also a percentage; 0% is dark (black) and 100% is white.

hsl(0, 100%, 30%);

hsl(0, 100%, 50%);

hsl(0, 100%, 70%);

hsl(0, 100%, 90%);

The following example defines different HSL colors:

### Example

#p1 {background-color: hsl(120, 100%, 50%);}  /\* green \*/  
#p2 {background-color: hsl(120, 100%, 75%);}  /\* light green \*/  
#p3 {background-color: hsl(120, 100%, 25%);}  /\* dark green \*/  
#p4 {background-color: hsl(120, 60%, 70%);}   /\* pastel green \*/

## HSLA Colors

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha), where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

hsla(0, 100%, 30%, 0.3);

hsla(0, 100%, 50%, 0.3);

hsla(0, 100%, 70%, 0.3);

hsla(0, 100%, 90%, 0.3);

The following example defines different HSLA colors:

### Example

#p1 {background-color: hsla(120, 100%, 50%, 0.3);}  /\* green with opacity \*/  
#p2 {background-color: hsla(120, 100%, 75%, 0.3);}  /\* light green with opacity \*/  
#p3 {background-color: hsla(120, 100%, 25%, 0.3);}  /\* dark green with opacity \*/  
#p4 {background-color: hsla(120, 60%, 70%, 0.3);}   /\* pastel green with opacity \*/

## Opacity

The CSS3 opacity property sets the opacity for a specified RGB value.

The opacity property value must be a number between 0.0 (fully transparent) and 1.0 (fully opaque).

rgb(255, 0, 0);opacity:0.2;

rgb(255, 0, 0);opacity:0.4;

rgb(255, 0, 0);opacity:0.6;

rgb(255, 0, 0);opacity:0.8;

Notice that the text above will also be opaque.

The following example shows different RGB values with opacity:

### Example

#p1 {background-color:rgb(255,0,0);opacity:0.6;}  /\* red with opacity \*/  
#p2 {background-color:rgb(0,255,0);opacity:0.6;}  /\* green with opacity \*/  
#p3 {background-color:rgb(0,0,255);opacity:0.6;}  /\* blue with opacity \*/

CSS3 Gradients

**Gradient Background**

CSS3 gradients let you display smooth transitions between two or more specified colors.

Earlier, you had to use images for these effects. However, by using CSS3 gradients you can reduce download time and bandwidth usage. In addition, elements with gradients look better when zoomed, because the gradient is generated by the browser.

CSS3 defines two types of gradients:

* **Linear Gradients (goes down/up/left/right/diagonally)**
* **Radial Gradients (defined by their center)**

## Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| linear-gradient | 26.0 10.0 -webkit- | 12.0 | 10.0 | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.1 -o- |
| radial-gradient | 26.0 10.0 -webkit- | 12.0 | 10.0 | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.6 -o- |
| repeating-linear-gradient | 26.0 10.0 -webkit- | 12.0 | 10.0 | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.1 -o- |
| repeating-radial-gradient | 26.0 10.0 -webkit- | 12.0 | 10.0 | 16.0 3.6 -moz- | 6.1 5.1 -webkit- | 12.1 11.6 -o- |

## CSS3 Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

### Syntax

background: linear-gradient(direction, color-stop1, color-stop2, ...);

**Linear Gradient - Top to Bottom (this is default)**

The following example shows a linear gradient that starts at the top. It starts red, transitioning to yellow:

### Example

#grad {  
    background: red; /\* For browsers that do not support gradients \*/  
    background: -webkit-linear-gradient(red, yellow); /\* For Safari 5.1 to 6.0 \*/  
    background: -o-linear-gradient(red, yellow); /\* For Opera 11.1 to 12.0 \*/  
    background: -moz-linear-gradient(red, yellow); /\* For Firefox 3.6 to 15 \*/  
    background: linear-gradient(red, yellow); /\* Standard syntax \*/  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear)

**Linear Gradient - Left to Right**

The following example shows a linear gradient that starts from the left. It starts red, transitioning to yellow:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-linear-gradient(left, red , yellow); /\* For Safari 5.1 to 6.0 \*/  
  background: -o-linear-gradient(right, red, yellow); /\* For Opera 11.1 to 12.0 \*/  
  background: -moz-linear-gradient(right, red, yellow); /\* For Firefox 3.6 to 15 \*/  
  background: linear-gradient(to right, red , yellow); /\* Standard syntax \*/  
}

**Linear Gradient - Diagonal**

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to yellow:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-linear-gradient(left top, red, yellow); /\* For Safari 5.1 to 6.0 \*/  
  background: -o-linear-gradient(bottom right, red, yellow); /\* For Opera 11.1 to 12.0 \*/  
  background: -moz-linear-gradient(bottom right, red, yellow); /\* For Firefox 3.6 to 15 \*/  
  background: linear-gradient(to bottom right, red, yellow); /\* Standard syntax \*/  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear_diagonal)

## Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right, to left, to bottom right, etc.).

### Syntax

background: linear-gradient(angle, color-stop1, color-stop2);

The angle is specified as an angle between a horizontal line and the gradient line.

The following example shows how to use angles on linear gradients:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-linear-gradient(-90deg, red, yellow); /\* For Safari 5.1 to 6.0 \*/  
  background: -o-linear-gradient(-90deg, red, yellow); /\* For Opera 11.1 to 12.0 \*/  
  background: -moz-linear-gradient(-90deg, red, yellow); /\* For Firefox 3.6 to 15 \*/  
  background: linear-gradient(-90deg, red, yellow); /\* Standard syntax \*/  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear_angles)

## Using Multiple Color Stops

The following example shows a linear gradient (from top to bottom) with multiple color stops:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-linear-gradient(red, yellow, green); /\* For Safari 5.1 to 6.0 \*/  
  background: -o-linear-gradient(red, yellow, green); /\* For Opera 11.1 to 12.0 \*/  
  background: -moz-linear-gradient(red, yellow, green); /\* For Firefox 3.6 to 15 \*/  
  background: linear-gradient(red, yellow, green); /\* Standard syntax \*/  
}

The following example shows how to create a linear gradient (from left to right) with the color of the rainbow and some text:

**Gradient Background**

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  /\* For Safari 5.1 to 6.0 \*/  
  background: -webkit-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet);  
  /\* For Opera 11.1 to 12.0 \*/  
  background: -o-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet);  
  /\* For Fx 3.6 to 15 \*/  
  background: -moz-linear-gradient(left,red,orange,yellow,green,blue,indigo,violet);  
  /\* Standard syntax \*/  
  background: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet);   
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear_rainbow)

## Using Transparency

CSS3 gradients also support transparency, which can be used to create fading effects.

To add transparency, we use the rgba() function to define the color stops. The last parameter in the rgba() function can be a value from 0 to 1, and it defines the transparency of the color: 0 indicates full transparency, 1 indicates full color (no transparency).

The following example shows a linear gradient that starts from the left. It starts fully transparent, transitioning to full color red:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-linear-gradient(left,rgba(255,0,0,0),rgba(255,0,0,1)); /\*Safari 5.1-6\*/  
  background: -o-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /\*Opera 11.1-12\*/  
  background: -moz-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /\*Fx 3.6-15\*/  
  background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /\*Standard\*/  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear_trans)

## Repeating a linear-gradient

The repeating-linear-gradient() function is used to repeat linear gradients:

### Example

A repeating linear gradient:

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  /\* Safari 5.1 to 6.0 \*/  
  background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /\* Opera 11.1 to 12.0 \*/  
  background: -o-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /\* Firefox 3.6 to 15 \*/  
  background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%);  
  /\* Standard syntax \*/  
  background: repeating-linear-gradient(red, yellow 10%, green 20%);  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear_repeating)

## CSS3 Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

### Syntax

background: radial-gradient(shape sizeatposition, start-color, ..., last-color);

By default, shape is ellipse, size is farthest-corner, and position is center.

**Radial Gradient - Evenly Spaced Color Stops (this is default)**

The following example shows a radial gradient with evenly spaced color stops:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-radial-gradient(red, yellow, green); /\* Safari 5.1 to 6.0 \*/  
  background: -o-radial-gradient(red, yellow, green); /\* For Opera 11.6 to 12.0 \*/  
  background: -moz-radial-gradient(red, yellow, green); /\* For Firefox 3.6 to 15 \*/  
  background: radial-gradient(red, yellow, green); /\* Standard syntax \*/  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-radial)

**Radial Gradient - Differently Spaced Color Stops**

The following example shows a radial gradient with differently spaced color stops:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-radial-gradient(red 5%, yellow 15%, green 60%); /\* Safari 5.1-6.0 \*/  
  background: -o-radial-gradient(red 5%, yellow 15%, green 60%); /\* For Opera 11.6-12.0 \*/  
  background: -moz-radial-gradient(red 5%, yellow 15%, green 60%); /\* For Firefox 3.6-15 \*/  
  background: radial-gradient(red 5%, yellow 15%, green 60%); /\* Standard syntax \*/  
6666}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-radial2)

## Set Shape

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

The following example shows a radial gradient with the shape of a circle:

### Example

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  background: -webkit-radial-gradient(circle, red, yellow, green); /\* Safari \*/  
  background: -o-radial-gradient(circle, red, yellow, green); /\* Opera 11.6 to 12.0 \*/  
  background: -moz-radial-gradient(circle, red, yellow, green); /\* Firefox 3.6 to 15 \*/  
  background: radial-gradient(circle, red, yellow, green); /\* Standard syntax \*/  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-radial_shape)

## Use of Different Size Keywords

The size parameter defines the size of the gradient. It can take four values:

* **closest-side**
* **farthest-side**
* **closest-corner**
* **farthest-corner**

### Example

A radial gradient with different size keywords:

#grad1 {  
  background: red; /\* For browsers that do not support gradients \*/  
  /\* Safari 5.1 to 6.0 \*/  
  background: -webkit-radial-gradient(60% 55%, closest-side, red, yellow, black);   
  /\* For Opera 11.6 to 12.0 \*/  
  background: -o-radial-gradient(60% 55%, closest-side, red, yellow, black);  
  /\* For Firefox 3.6 to 15 \*/  
  background: -moz-radial-gradient(60% 55%, closest-side, red, yellow, black);  
  /\* Standard syntax \*/  
  background: radial-gradient(closest-side at 60% 55%, red, yellow, black);  
}  
  
#grad2 {  
  /\* Safari 5.1 to 6.0 \*/  
  background: -webkit-radial-gradient(60% 55%, farthest-side, red, yellow, black);  
  /\* Opera 11.6 to 12.0 \*/   
  background: -o-radial-gradient(60% 55%, farthest-side, red, yellow, black);  
  /\* For Firefox 3.6 to 15 \*/  
  background: -moz-radial-gradient(60% 55%, farthest-side, red, yellow, black);  
  /\* Standard syntax \*/  
  background: radial-gradient(farthest-side at 60% 55%, red, yellow, black);  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-radial_size)

## Repeating a radial-gradient

The repeating-radial-gradient() function is used to repeat radial gradients:

### Example

A repeating radial gradient:

#grad {  
  background: red; /\* For browsers that do not support gradients \*/  
  /\* For Safari 5.1 to 6.0 \*/  
  background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /\* For Opera 11.6 to 12.0 \*/  
  background: -o-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /\* For Firefox 3.6 to 15 \*/  
  background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%);  
  /\* Standard syntax \*/  
  background: repeating-radial-gradient(red, yellow 10%, green 15%);  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-radial_repeating)

CSS3 Shadow Effects

## CSS3 Shadow Effects

With CSS3 you can add shadow to text and to elements.

In this chapter you will learn about the following properties:

* text-shadow
* box-shadow

## Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specifies the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| text-shadow | 4.0 | 12.0 | 10.0 | 3.5 | 4.0 | 9.5 |
| box-shadow | 10.0 4.0 -webkit- | 12.0 | 9.0 | 4.0 3.5 -moz- | 5.1 3.1 -webkit- | 10.5 |

## CSS3 Text Shadow

The CSS3 text-shadow property applies shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

## Text shadow effect!

### Example

h1 {  
    text-shadow: 2px 2px;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow1)

Next, add a color to the shadow:

## Text shadow effect!

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow2)

Then, add a blur effect to the shadow:

## Text shadow effect!

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow3)

The following example shows a white text with black shadow:

## Text shadow effect!

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow4)

The following example shows a red neon glow shadow:

## Text shadow effect!

### Example

h1 {  
    text-shadow: 0 0 3px #FF0000;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow5)

## Multiple Shadows

To add more than one shadow to the text, you can add a comma-separated list of shadows.

The following example shows a red and blue neon glow shadow:

## Text shadow effect!

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow6)

The following example shows a white text with black, blue, and darkblue shadow:

## Text shadow effect!

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-shadow7)

## CSS3 box-shadow Property

The CSS3 box-shadow property applies shadow to elements.

In its simplest use, you only specify the horizontal shadow and the vertical shadow:

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

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_box-shadow)

Next, add a color to the shadow:

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

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_box-shadow2)

Next, add a blur effect to the shadow:

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

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_box-shadow3)

You can also add shadows to the ::before and ::after pseudo-classes, to create an interesting effect:

### Example

#boxshadow {  
    position: relative;  
    box-shadow: 1px 2px 4px rgba(0, 0, 0, .5);  
    padding: 10px;  
    background: white;  
}  
  
#boxshadow img {  
    width: 100%;  
    border: 1px solid #8a4419;  
    border-style: inset;  
}  
  
#boxshadow::after {  
    content: '';  
    position: absolute;  
    z-index: -1; /\* hide shadow behind image \*/  
    box-shadow: 0 15px 20px rgba(0, 0, 0, 0.3);   
    width: 70%;   
    left: 15%; /\* one half of the remaining 30% \*/  
    height: 100px;  
    bottom: 0;  
}

## Cards

An example of using the box-shadow property to create paper-like cards:

# 1

January 1, 2016



Hardanger, Norway

### Example

div.card {  
    width: 250px;  
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);  
    text-align: center;  
}

## CSS3 Shadow Properties

The following table lists the CSS3 shadow properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [box-shadow](http://www.w3schools.com/cssref/css3_pr_box-shadow.asp) | Adds one or more shadows to an element |
| [text-shadow](http://www.w3schools.com/cssref/css3_pr_text-shadow.asp) | Adds one or more shadows to a text  CSS3 Text  CSS3 contains several new text features.  In this chapter you will learn about the following text properties:   * text-overflow * word-wrap * word-break  Browser Support The numbers in the table specify the first browser version that fully supports the property.  Numbers followed by -o- specify the first version that worked with a prefix.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Property |  |  |  |  |  |  | | text-overflow | 4.0 | 12.0 | 6.0 | 7.0 | 3.1 | 11.0 9.0 -o- | | word-wrap | 23.0 | 12.0 | 5.5 | 3.5 | 6.1 | 12.1 | | word-break | 4.0 | 12.0 | 5.5 | 15.0 | 3.1 | 15.0 |  CSS3 Text Overflow The CSS3 text-overflow property specifies how overflowed content that is not displayed should be signaled to the user.  It can be clipped:  This is some long text that will not fit in the box  or it can be rendered as an ellipsis (...):  This is some long text that will not fit in the box  The CSS code is as follows: Example p.test1 {     white-space: nowrap;      width: 200px;      border: 1px solid #000000;     overflow: hidden;     text-overflow: clip;  }  p.test2 {     white-space: nowrap;      width: 200px;      border: 1px solid #000000;     overflow: hidden;     text-overflow: ellipsis;  }  [Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-overflow)  The following example shows how you can display the overflowed content when hovering over the element: Example div.test:hover {     text-overflow: inherit;     overflow: visible; }  [Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_text-overflow_hover) CSS3 Word Wrapping The CSS3 word-wrap property allows long words to be able to be broken and  wrap onto the next line.  If a word is too long to fit within an area, it expands outside:  This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.  The word-wrap property allows you to force the text to wrap - even if it means split  ing it in the middle of a word:  This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.  The CSS code is as follows: Example Allow long words to be able to be broken and wrap onto the next line:  p {     word-wrap: break-word; }  [Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_word-wrap) CSS3 Word Breaking The CSS3 word-break property specifies line breaking rules.  This paragraph contains some text. This line will-break-at-hyphens.  This paragraph contains some text. The lines will break at any character.  The CSS code is as follows: Example p.test1 {     word-break: keep-all; }  p.test2 {     word-break: break-all; }  [Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_word-break) CSS3 Text Properties The following table lists the new CSS3 text properties:   |  |  | | --- | --- | | **Property** | **Description** | | [text-align-last](http://www.w3schools.com/cssref/css3_pr_text-align-last.asp) | Specifies how to align the last line of a text | | [text-emphasis](http://www.w3schools.com/cssref/css3_pr_text-emphasis.asp) | A shorthand for setting text-emphasis-style and text-emphasis-color in one declaration | | [text-justify](http://www.w3schools.com/cssref/css3_pr_text-justify.asp) | Specifies how justified text should be aligned and spaced | | [text-overflow](http://www.w3schools.com/cssref/css3_pr_text-overflow.asp) | Specifies how overflowed content that is not displayed should be signaled to the user | | [word-break](http://www.w3schools.com/cssref/css3_pr_word-break.asp) | Specifies line breaking rules for non-CJK scripts | | [word-wrap](http://www.w3schools.com/cssref/css3_pr_word-wrap.asp) | Allows long words to be able to be broken and wrap onto the next line | |

CSS3 Web Fonts

## CSS3 Web Fonts - The @font-face Rule

Web fonts allow Web designers to use fonts that are not installed on the user's computer.

When you have found/bought the font you wish to use, just include the font file on your web server, and it will be automatically downloaded to the user when needed.

Your "own" fonts are defined within the CSS3 @font-face rule.

## Browser Support

The numbers in the table specify the first browser version that fully supports the property.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| @font-face | 4.0 | 12.0 | 9.0 | 3.5 | 3.2 | 10.0 |

## Different Font Formats

**TrueType Fonts (TTF)**

TrueType is a font standard developed in the late 1980s, by Apple and Microsoft. TrueType is the most common font format for both the Mac OS and Microsoft Windows operating systems.

**OpenType Fonts (OTF)**

OpenType is a format for scalable computer fonts. It was built on TrueType, and is a registered trademark of Microsoft. OpenType fonts are used commonly today on the major computer platforms.

**The Web Open Font Format (WOFF)**

WOFF is a font format for use in web pages. It was developed in 2009, and is now a W3C Recommendation. WOFF is essentially OpenType or TrueType with compression and additional metadata. The goal is to support font distribution from a server to a client over a network with bandwidth constraints.

**The Web Open Font Format (WOFF 2.0)**

TrueType/OpenType font that provides better compression than WOFF 1.0.

**SVG Fonts/Shapes**

SVG fonts allow SVG to be used as glyphs when displaying text. The SVG 1.1 specification define a font module that allows the creation of fonts within an SVG document. You can also apply CSS to SVG documents, and the @font-face rule can be applied to text in SVG documents.

**Embedded OpenType Fonts (EOT)**

EOT fonts are a compact form of OpenType fonts designed by Microsoft for use as embedded fonts on web pages.

## Browser Support for Font Formats

The numbers in the table specifies the first browser version that fully supports the font format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Font format |  |  |  |  |  |
| TTF/OTF | 9.0\* | 4.0 | 3.5 | 3.1 | 10.0 |
| WOFF | 9.0 | 5.0 | 3.6 | 5.1 | 11.1 |
| WOFF2 | Not supported | 36.0 | 35.0\* | Not supported | 26.0 |
| SVG | Not supported | 4.0 | Not supported | 3.2 | 9.0 |
| EOT | 6.0 | Not supported | Not supported | Not supported | Not supported |

\*IE: The font format only works when set to be "installable".

\*Firefox: Not supported by default, but can be enabled (need to set a flag to "true" to use WOFF2).

## Using The Font You Want

In the CSS3 @font-face rule you must first define a name for the font (e.g. myFirstFont), and then point to the font file.

|  |  |
| --- | --- |
| **Note** | **Tip:** Always use lowercase letters for the font URL. Uppercase letters can give unexpected results in IE. |

To use the font for an HTML element, refer to the name of the font (myFirstFont) through the font-familyproperty:

### Example

@font-face {  
    font-family: myFirstFont;  
    src: url(sansation\_light.woff);  
}  
  
div {  
    font-family: myFirstFont;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_font-face_rule)

## Using Bold Text

You must add another @font-face rule containing descriptors for bold text:

### Example

@font-face {  
    font-family: myFirstFont;  
    src: url(sansation\_bold.woff);  
    font-weight: bold;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_font-face_rule_bold)

The file "sansation\_bold.woff" is another font file, that contains the bold characters for the Sansation font.

Browsers will use this whenever a piece of text with the font-family "myFirstFont" should render as bold.

This way you can have many @font-face rules for the same font.

## Test Yourself with Exercises!

[Exercise 1 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_fonts1)  [Exercise 2 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_fonts2)

## CSS3 Font Descriptors

The following table lists all the font descriptors that can be defined inside the @font-face rule:

|  |  |  |
| --- | --- | --- |
| **Descriptor** | **Values** | **Description** |
| font-family | *Name* | Required. Defines a name for the font |
| Src | *URL* | Required. Defines the URL of the font file |
| font-stretch | normal condensed ultra-condensed extra-condensed semi-condensed expanded semi-expanded extra-expanded ultra-expanded | Optional. Defines how the font should be stretched. Default is "normal" |
| font-style | normal italic oblique | Optional. Defines how the font should be styled. Default is "normal" |
| font-weight | normal bold 100 200 300 400 500 600 700 800 900 | Optional. Defines the boldness of the font. Default is "normal" |
| unicode-range | *unicode-range* | Optional. Defines the range of UNICODE characters the font supports. Default is "U+0-10FFFF" |

## CSS3 Transforms

CSS3 transforms allow you to translate, rotate, scale, and skew elements.

A transformation is an effect that lets an element change shape, size and position.

CSS3 supports 2D and 3D transformations.

Mouse over the elements below to see the difference between a 2D and a 3D transformation:

**2D rotate**

**3D rotate**

## Browser Support for 2D Transforms

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -ms-, -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| Transform | 36.0 4.0 -webkit- | 12.0 | 10.0 9.0 -ms- | 16.0 3.5 -moz- | 3.2 -webkit- | 23.0 15.0 -webkit- 12.1 10.5 -o- |
| transform-origin (two-value syntax) | 36.0 4.0 -webkit- | 12.0 | 10.0 9.0 -ms- | 16.0 3.5 -moz- | 3.2 -webkit- | 23.0 15.0 -webkit- 12.1 10.5 -o- |

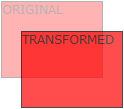
## CSS3 2D Transforms

In this chapter you will learn about the following 2D transformation methods:

* translate()
* rotate()
* scale()
* skewX()
* skewY()
* matrix()

|  |  |
| --- | --- |
| **Note** | **Tip:** You will learn about 3D transformations in the next chapter. |

## The translate() Method



The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

The following example moves the <div> element 50 pixels to the right, and 100 pixels down from its current position:

### Example

div {  
    -ms-transform: translate(50px,100px); /\* IE 9 \*/  
    -webkit-transform: translate(50px,100px); /\* Safari \*/  
    transform: translate(50px,100px);  
}

## The rotate() Method



The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

The following example rotates the <div> element clockwise with 20 degrees:

### Example

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

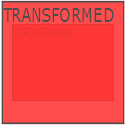
Using negative values will rotate the element counter-clockwise.

The following example rotates the <div> element counter-clockwise with 20 degrees:

### Example

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

## The scale() Method



The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).

The following example increases the <div> element to be two times of its original width, and three times of its original height:

### Example

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

The following example decreases the <div> element to be half of its original width and height:

### Example

div {  
    -ms-transform: scale(0.5,0.5); /\* IE 9 \*/  
    -webkit-transform: scale(0.5,0.5); /\* Safari \*/  
    transform: scale(0.5,0.5);  
}

## The skewX() Method

The skewX() method skews an element along the X-axis by the given angle.

The following example skews the <div> element 20 degrees along the X-axis:

### Example

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

## The skewY() Method

The skewY() method skews an element along the Y-axis by the given angle.

The following example skews the <div> element 20 degrees along the Y-axis:

### Example

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

## The skew() Method

The skew() method skews an element along the X and Y-axis by the given angles.

The following example skews the <div> element 20 degrees along the X-axis, and 10 degrees along the Y-axis:

### Example

div {  
    -ms-transform: skew(20deg, 10deg); /\* IE 9 \*/  
    -webkit-transform: skew(20deg, 10deg); /\* Safari \*/  
    transform: skew(20deg, 10deg);  
}

If the second parameter is not specified, it has a zero value. So, the following example skews the <div> element 20 degrees along the X-axis:

### Example

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

## The matrix() Method



The matrix() method combines all the 2D transform methods into one.

The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.

The parameters are as follow: matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY()):

### Example

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 Transform Properties

The following table lists all the 2D transform properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [transform](http://www.w3schools.com/cssref/css3_pr_transform.asp) | Applies a 2D or 3D transformation to an element |
| [transform-origin](http://www.w3schools.com/cssref/css3_pr_transform-origin.asp) | Allows you to change the position on transformed elements |

## 2D Transform Methods

|  |  |
| --- | --- |
| **Function** | **Description** |
| matrix(*n,n,n,n,n,n*) | Defines a 2D transformation, using a matrix of six values |
| translate(*x,y*) | Defines a 2D translation, moving the element along the X- and the Y-axis |
| translateX(*n*) | Defines a 2D translation, moving the element along the X-axis |
| translateY(*n*) | Defines a 2D translation, moving the element along the Y-axis |
| scale(*x,y*) | Defines a 2D scale transformation, changing the elements width and height |
| scaleX(*n*) | Defines a 2D scale transformation, changing the element's width |
| scaleY(*n*) | Defines a 2D scale transformation, changing the element's height |
| rotate(*angle*) | Defines a 2D rotation, the angle is specified in the parameter |
| skew(*x-angle,y-angle*) | Defines a 2D skew transformation along the X- and the Y-axis |
| skewX(*angle*) | Defines a 2D skew transformation along the X-axis |
| skewY(*angle*) | Defines a 2D skew transformation along the Y-axis |

CSS3 3D Transforms

## CSS3 3D Transforms

CSS3 allows you to format your elements using 3D transformations.

Mouse over the elements below to see the difference between a 2D and a 3D transformation:

**2D rotate**

**3D rotate**

## Browser Support for 3D Transforms

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| Transform | 36.0 12.0 -webkit- | 12.0 | 10.0 | 16.0 10.0 -moz- | 4.0 -webkit- | 23.0 15.0 -webkit- |
| transform-origin (three-value syntax) | 36.0 12.0 -webkit- | 12.0 | 10.0 | 16.0 10.0 -moz- | 4.0 -webkit- | 23.0 15.0 -webkit- |
| transform-style | 36.0 12.0 -webkit- | 12.0 | 11.0 | 16.0 10.0 -moz- | 4.0 -webkit- | 23.0 15.0 -webkit- |
| Perspective | 36.0 12.0 -webkit- | 12.0 | 10.0 | 16.0 10.0 -moz- | 4.0 -webkit- | 23.0 15.0 -webkit- |
| perspective-origin | 36.0 12.0 -webkit- | 12.0 | 10.0 | 16.0 10.0 -moz- | 4.0 -webkit- | 23.0 15.0 -webkit- |
| backface-visibility | 36.0 12.0 -webkit- | 12.0 | 10.0 | 16.0 10.0 -moz- | 4.0 -webkit- | 23.0 15.0 -webkit- |

## CSS3 3D Transforms

In this chapter you will learn about the following 3D transformation methods:

* rotateX()
* rotateY()
* rotateZ()

## The rotateX() Method

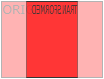


The rotateX() method rotates an element around its X-axis at a given degree:

### Example

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

## The rotateY() Method



The rotateY() method rotates an element around its Y-axis at a given degree:

### Example

div {  
    -webkit-transform: rotateY(130deg); /\* Safari \*/  
    transform: rotateY(130deg);  
}

## The rotateZ() Method

The rotateZ() method rotates an element around its Z-axis at a given degree:

### Example

div {  
    -webkit-transform: rotateZ(90deg); /\* Safari \*/  
    transform: rotateZ(90deg);  
}

## CSS3 Transform Properties

The following table lists all the 3D transform properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [transform](http://www.w3schools.com/cssref/css3_pr_transform.asp) | Applies a 2D or 3D transformation to an element |
| [transform-origin](http://www.w3schools.com/cssref/css3_pr_transform-origin.asp) | Allows you to change the position on transformed elements |
| [transform-style](http://www.w3schools.com/cssref/css3_pr_transform-style.asp) | Specifies how nested elements are rendered in 3D space |
| [perspective](http://www.w3schools.com/cssref/css3_pr_perspective.asp) | Specifies the perspective on how 3D elements are viewed |
| [perspective-origin](http://www.w3schools.com/cssref/css3_pr_perspective-origin.asp) | Specifies the bottom position of 3D elements |
| [backface-visibility](http://www.w3schools.com/cssref/css3_pr_backface-visibility.asp) | Defines whether or not an element should be visible when not facing the screen |

## 3D Transform Methods

|  |  |
| --- | --- |
| **Function** | **Description** |
| matrix3d (*n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n*) | Defines a 3D transformation, using a 4x4 matrix of 16 values |
| translate3d(*x,y,z*) | Defines a 3D translation |
| translateX(*x*) | Defines a 3D translation, using only the value for the X-axis |
| translateY(*y*) | Defines a 3D translation, using only the value for the Y-axis |
| translateZ(*z*) | Defines a 3D translation, using only the value for the Z-axis |
| scale3d(*x,y,z*) | Defines a 3D scale transformation |
| scaleX(*x*) | Defines a 3D scale transformation by giving a value for the X-axis |
| scaleY(*y*) | Defines a 3D scale transformation by giving a value for the Y-axis |
| scaleZ(*z*) | Defines a 3D scale transformation by giving a value for the Z-axis |
| rotate3d(*x,y,z,angle*) | Defines a 3D rotation |
| rotateX(*angle*) | Defines a 3D rotation along the X-axis |
| rotateY(*angle*) | Defines a 3D rotation along the Y-axis |
| rotateZ(*angle*) | Defines a 3D rotation along the Z-axis |
| perspective(*n*) | Defines a perspective view for a 3D transformed element |

## CSS3 Transitions

CSS3 transitions allows you to change property values smoothly (from one value to another), over a given duration.

**Example:** Mouse over the element below to see a CSS3 transition effect:

**CSS3**

## Browser Support for Transitions

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| Transition | 26.0 4.0 -webkit- | 12.0 | 10.0 | 16.0 4.0 -moz- | 6.1 3.1 -webkit- | 12.1 10.5 -o- |
| transition-delay | 26.0 4.0 -webkit- | 12.0 | 10.0 | 16.0 4.0 -moz- | 6.1 3.1 -webkit- | 12.1 10.5 -o- |
| transition-duration | 26.0 4.0 -webkit- | 12.0 | 10.0 | 16.0 4.0 -moz- | 6.1 3.1 -webkit- | 12.1 10.5 -o- |
| transition-property | 26.0 4.0 -webkit- | 12.0 | 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- | 12.0 | 10.0 | 16.0 4.0 -moz- | 6.1 3.1 -webkit- | 12.1 10.5 -o- |

## How to Use CSS3 Transitions?

To create a transition effect, you must specify two things:

* the CSS property you want to add an effect to
* the duration of the effect

**Note:** If the duration part is not specified, the transition will have no effect, because the default value is 0.

The following example shows a 100px \* 100px red <div> element. The <div> element has also specified a transition effect for the width property, with a duration of 2 seconds:

### Example

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

The transition effect will start when the specified CSS property (width) changes value.

Now, let us specify a new value for the width property when a user mouses over the <div> element:

### Example

div:hover {  
    width: 300px;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_transition1)

Notice that when the cursor mouses out of the element, it will gradually change back to its original style.

## Change Several Property Values

The following example adds a transition effect for both the width and height property, with a duration of 2 seconds for the width and 4 seconds for the height:

### Example

div {  
    -webkit-transition: width 2s, height 4s; /\* Safari \*/  
    transition: width 2s, height 4s;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_transition2)

## Specify the Speed Curve of the Transition

The transition-timing-function property specifies the speed curve of the transition effect.

The transition-timing-function property can have the following values:

* ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
* linear - specifies a transition effect with the same speed from start to end
* ease-in - specifies a transition effect with a slow start
* ease-out - specifies a transition effect with a slow end
* ease-in-out - specifies a transition effect with a slow start and end
* cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

The following example shows the some of the different speed curves that can be used:

### Example

#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;}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_transition_speed)

## Delay the Transition Effect

The transition-delay property specifies a delay (in seconds) for the transition effect.

The following example has a 1 second delay before starting:

### Example

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

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_transition_delay)

## Transition + Transformation

The following example also adds a transformation to the transition effect:

### Example

div {  
    -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /\* Safari \*/  
    transition: width 2s, height 2s, transform 2s;  
}

## More Transition Examples

The CSS3 transition properties can be specified one by one, like this:

### Example

div {  
    transition-property: width;  
    transition-duration: 2s;  
    transition-timing-function: linear;  
    transition-delay: 1s;  
}

or by using the shorthand property transition:

### Example

div {  
    transition: width 2s linear 1s;  
}

## CSS3 Transition Properties

The following table lists all the transition properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [transition](http://www.w3schools.com/cssref/css3_pr_transition.asp) | A shorthand property for setting the four transition  properties into a single property |
| [transition-delay](http://www.w3schools.com/cssref/css3_pr_transition-delay.asp) | Specifies a delay (in seconds) for the transition effect |
| [transition-duration](http://www.w3schools.com/cssref/css3_pr_transition-duration.asp) | Specifies how many seconds or milliseconds a transition effect  takes to complete |
| [transition-property](http://www.w3schools.com/cssref/css3_pr_transition-property.asp) | Specifies the name of the CSS property the transition effect is  For |
| [transition-timing-function](http://www.w3schools.com/cssref/css3_pr_transition-timing-function.asp) | Specifies the speed curve of the transition effect |

CSS3 Animations

## CSS3 Animations

CSS3 animations allows animation of most HTML elements without using JavaScript or Flash!

**CSS3  
Animation**

## Browser Support for Animations

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |  |
| @keyframes | 43.0 4.0 -webkit- | 12.0 | 10.0 | 16.0 5.0 -moz- | 9.0 4.0 -webkit- | 30.0 15.0 -webkit- 12.0 -o- |
| animation | 43.0 4.0 -webkit- | 12.0 | 10.0 | 16.0 5.0 -moz- | 9.0 4.0 -webkit- | 30.0 15.0 -webkit- 12.0 -o- |

## What are CSS3 Animations?

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times you want.

To use CSS3 animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

## The @keyframes Rule

When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

The following example binds the "example" animation to the <div> element. The animation will lasts for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow":

### Example

/\* The animation code \*/  
@keyframes example {  
    from {background-color: red;}  
    to {background-color: yellow;}  
}  
  
/\* The element to apply the animation to \*/  
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1)

**Note:** If the animation-duration property is not specified, the animation will have no effect, because the default value is 0.

In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

It is also possible to use percent. By using percent, you can add as many style changes as you like.

The following example will change the background-color of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

### Example

/\* The animation code \*/  
@keyframes example {  
    0%   {background-color: red;}  
    25%  {background-color: yellow;}  
    50%  {background-color: blue;}  
    100% {background-color: green;}  
}  
  
/\* The element to apply the animation to \*/  
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation2)

The following example will change both the background-color and the position of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

### Example

/\* The animation code \*/  
@keyframes example {  
    0%   {background-color: red; left:0px; top:0px;}  
    25%  {background-color: yellow; left:200px; top:0px;}  
    50%  {background-color: blue; left:200px; top:200px;}  
    75%  {background-color: green; left:0px; top:200px;}  
    100% {background-color: red; left:0px; top:0px;}  
}  
  
/\* The element to apply the animation to \*/  
div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation3)

## Delay an Animation

The animation-delay property specifies a delay for the start of an animation.

The following example has a 2 seconds delay before starting the animation:

### Example

div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
    animation-delay: 2s;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_delay)

## Set How Many Times an Animation Should Run

The animation-iteration-count property specifies the number of times an animation should run.

The following example will run the animation 3 times before it stops:

### Example

div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
    animation-iteration-count: 3;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_count)

The following example uses the value "infinite" to make the animation continue for ever:

### Example

div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
    animation-iteration-count: infinite;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_count2)

## Run Animation in Reverse Direction or Alternate Cycles

The animation-direction property is used to let an animation run in reverse direction or alternate cycles.

The following example will run the animation in reverse direction:

### Example

div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
    animation-iteration-count: 3;  
    animation-direction: reverse;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_direction)

The following example uses the value "alternate" to make the animation first run forward, then backward, then forward:

### Example

div {  
    width: 100px;  
    height: 100px;  
    position: relative;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
    animation-iteration-count: 3;  
    animation-direction: alternate;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_direction2)

## Specify the Speed Curve of the Animation

The animation-timing-function property specifies the speed curve of the animation.

The animation-timing-function property can have the following values:

* ease - specifies an animation with a slow start, then fast, then end slowly (this is default)
* linear - specifies an animation with the same speed from start to end
* ease-in - specifies an animation with a slow start
* ease-out - specifies an animation with a slow end
* ease-in-out - specifies an animation with a slow start and end
* cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

The following example shows the some of the different speed curves that can be used:

### Example

#div1 {animation-timing-function: linear;}  
#div2 {animation-timing-function: ease;}  
#div3 {animation-timing-function: ease-in;}  
#div4 {animation-timing-function: ease-out;}  
#div5 {animation-timing-function: ease-in-out;}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation_speed)

## Animation Shorthand Property

The example below uses six of the animation properties:

### Example

div {  
    animation-name: example;  
    animation-duration: 5s;  
    animation-timing-function: linear;  
    animation-delay: 2s;  
    animation-iteration-count: infinite;  
    animation-direction: alternate;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation4)

The same animation effect as above can be achieved by using the shorthand animation property:

### Example

div {  
    animation: example 5s linear 2s infinite alternate;  
}

[Try it yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation5)

## Test Yourself with Exercises!

[Exercise 1 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations1)  [Exercise 2 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations2)  [Exercise 3 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations3)  [Exercise 4 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations4)  [Exercise 5 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations5)  [Exercise 6 »](http://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations6)

## CSS3 Animation Properties

The following table lists the @keyframes rule and all the animation properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [@keyframes](http://www.w3schools.com/cssref/css3_pr_animation-keyframes.asp) | Specifies the animation code |
| [animation](http://www.w3schools.com/cssref/css3_pr_animation.asp) | A shorthand property for setting all the animation properties |
| [animation-delay](http://www.w3schools.com/cssref/css3_pr_animation-delay.asp) | Specifies a delay for the start of an animation |
| [animation-direction](http://www.w3schools.com/cssref/css3_pr_animation-direction.asp) | Specifies whether an animation should play in reverse direction or alternate cycles |
| [animation-duration](http://www.w3schools.com/cssref/css3_pr_animation-duration.asp) | Specifies how many seconds or milliseconds an animation takes to complete one cycle |
| [animation-fill-mode](http://www.w3schools.com/cssref/css3_pr_animation-fill-mode.asp) | Specifies a style for the element when the animation is not playing (when it is finished, or when it has a delay) |
| [animation-iteration-count](http://www.w3schools.com/cssref/css3_pr_animation-iteration-count.asp) | Specifies the number of times an animation should be played |
| [animation-name](http://www.w3schools.com/cssref/css3_pr_animation-name.asp) | Specifies the name of the @keyframes animation |
| [animation-play-state](http://www.w3schools.com/cssref/css3_pr_animation-play-state.asp) | Specifies whether the animation is running or paused |
| [animation-timing-function](http://www.w3schools.com/cssref/css3_pr_animation-timing-function.asp) | Specifies the speed curve of the animation |

Learn how to style images using CSS.

## Rounded Images

Use the border-radius property to create rounded images:



### Example

Rounded Image:

img {  
    border-radius: 8px;  
}

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_ex_images_round)



### Example

Circled Image:

img {  
    border-radius: 50%;  
}

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_ex_images_circle)

## Thumbnail Images

Use the border property to create thumbnail images.

Thumbnail Image:



### Example

img {  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    padding: 5px;  
}  
  
<img src="paris.jpg" alt="Paris">

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_ex_images_thumbnail)

Thumbnail Image as Link:

[](http://www.w3schools.com/css/paris.jpg)

### Example

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);  
}  
  
<a href="paris.jpg">  
  <img src="paris.jpg" alt="Paris">  
</a>

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_ex_images_thumbnail_link)

## Responsive Images

Responsive images will automatically adjust to fit the size of the screen.

Resize the browser window to see the effect:



If you want an image to scale down if it has to, but never scale up to be larger than its original size, add the following:

### Example

img {  
    max-width: 100%;  
    height: auto;  
}

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_ex_images_responsive)

**Tip:** Read more about Responsive Web Design in our [CSS RWD Tutorial](http://www.w3schools.com/css/css_rwd_intro.asp).

## Image Text

How to position text in an image:

### Example



Bottom Left

Top Left

Top Right

Bottom Right

Centered

## Polaroid Images / Cards



The Troll's tongue in Hardanger, Norway



Northern Lights in Norway

### Example

div.polaroid {  
    width: 80%;  
    background-color: white;  
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);  
}  
  
img {width: 100%}  
  
div.container {  
    text-align: center;  
    padding: 10px 20px;  
}

## Image Filters

The CSS filter property adds visual effects (like blur and saturation) to an element.

**Note:** This property is not supported in Internet Explorer.

### Example

Change the color of all images to black and white (100% gray):

img {  
    -webkit-filter: grayscale(100%); /\* Chrome, Safari, Opera \*/  
    filter: grayscale(100%);  
}

**Note:** The filter property is not supported in Internet Explorer.



Original image



grayscale(100%)

## Responsive Image Gallery

CSS can be used to create image galleries. This example use media queries to re-arrange the images on different screen sizes. Resize the browser window to see the effect:

[](http://www.w3schools.com/css/img_fjords.jpg)

Add a description of the image here

[](http://www.w3schools.com/css/img_forest.jpg)

Add a description of the image here

[](http://www.w3schools.com/css/img_lights.jpg)

Add a description of the image here

[](http://www.w3schools.com/css/img_mountains.jpg)

Add a description of the image here

### Example

.responsive {  
    padding: 0 6px;  
    float: left;  
    width: 24.99999%;  
}  
  
@media only screen and (max-width: 700px){  
    .responsive {  
        width: 49.99999%;  
        margin: 6px 0;  
    }  
}  
  
@media only screen and (max-width: 500px){  
    .responsive {  
        width: 100%;  
    }  
}

## Image Modal (Advanced)

This is an example to demonstrate how CSS and JavaScript can work together.

First, use CSS to create a modal window (dialog box), and hide it by default.

Then, use a JavaScript to show the modal window and to display the image inside the modal, when a user clicks on the image:



### Example

// Get the modal  
var modal = document.getElementById('myModal');  
  
// Get the image and insert it inside the modal - use its "alt" text as a caption  
var img = document.getElementById('myImg');  
var modalImg = document.getElementById("img01");  
var captionText = document.getElementById("caption");  
img.onclick = function(){  
    modal.style.display = "block";  
    modalImg.src = this.src;  
    modalImg.alt = this.alt;  
    captionText.innerHTML = this.alt;  
}  
  
// Get the <span> element that closes the modal  
var span = document.getElementsByClassName("close")[0];  
  
// When the user clicks on <span> (x), close the modal  
span.onclick = function() {   
    modal.style.display = "none";  
}

CSS3 Buttons

## Basic Button Styling

Default Button CSS Button

### Example

.button {  
    background-color: #4CAF50; /\* Green \*/  
    border: none;  
    color: white;  
    padding: 15px 32px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
    font-size: 16px;  
}

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_basic)

## Button Colors

Green Blue Red Gray Black

Use the background-color property to change the background color of a button:

### Example

.button1 {background-color: #4CAF50;} /\* Green \*/  
.button2 {background-color: #008CBA;} /\* Blue \*/  
.button3 {background-color: #f44336;} /\* Red \*/   
.button4 {background-color: #e7e7e7; color: black;} /\* Gray \*/   
.button5 {background-color: #555555;} /\* Black \*/

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_color)

## Button Sizes

10px 12px 16px 20px 24px

Use the font-size property to change the size of a button:

### Example

.button1 {font-size: 10px;}  
.button2 {font-size: 12px;}  
.button3 {font-size: 16px;}  
.button4 {font-size: 20px;}  
.button5 {font-size: 24px;}

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_font)

## Rounded Buttons

2px 4px 8px 12px 50%

Use the border-radius property to add rounded corners to a button:

### Example

.button1 {border-radius: 2px;}  
.button2 {border-radius: 4px;}  
.button3 {border-radius: 8px;}  
.button4 {border-radius: 12px;}  
.button5 {border-radius: 50%;}

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_round)

## Colored Button Borders

Green Blue Red Gray Black

Use the border property to add a colored border to a button:

### Example

.button1 {  
    background-color: white;  
    color: black;  
    border: 2px solid #4CAF50; /\* Green \*/  
}  
...

[Try it Yourself »](http://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_border)

## Hoverable Buttons

Green Blue Red Grey Black   
Green Blue Red Grey Black

Use the :hover selector to change the style of a button when you move the mouse over it.

**Tip:** Use the transition-duration property to determine the speed of the "hover" effect:

### Example

.button {  
    -webkit-transition-duration: 0.4s; /\* Safari \*/  
    transition-duration: 0.4s;  
}  
  
.button:hover {  
    background-color: #4CAF50; /\* Green \*/  
    color: white;  
}  
...

## Shadow Buttons

Shadow Button Shadow on hover

Use the box-shadow property to add shadows to a button:

### Example

.button1 {  
    box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);  
}  
  
.button2:hover {  
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);  
}

## Disabled Buttons

Normal Button Disabled Button

Use the opacity property to add transparency to a button (creates a "disabled" look).

**Tip:** You can also add the cursor property with a value of "not-allowed", which will display a "no parking sign" when you mouse over the button:

### Example

.disabled {  
    opacity: 0.6;  
    cursor: not-allowed;  
}

## Button Width

250px  
50% 100%

By default, the size of the button is determined by its text content (as wide as its content). Use the widthproperty to change the width of a button:

### Example

.button1 {width: 250px;}  
.button2 {width: 50%;}  
.button3 {width: 100%;}

## Button Groups

ButtonButtonButtonButton

Remove margins and add float:left to each button to create a button group:

### Example

.button {  
    float: left;  
}

## Bordered Button Groups

ButtonButtonButtonButton

Use the border property to create a bordered button group:

### Example

.button {  
    float: left;  
    border: 1px solid green  
}

## Animated Buttons

### Example

Add an arrow on hover:

Hover