GlobeTechnologies

CSS

CSS3 Introduction

CSS3 is completely backwards compatible, so you will not have to change existing designs. Browsers will always support CSS2.

CSS3 Modules

CSS3 is split up into "modules". The old specification has been split into smaller pieces, and new ones are also added.

Some of the most important CSS3 modules are:

- Selectors
- Box Model
- Backgrounds and Borders
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

CSS3 Recommendation

The CSS3 specification is still under development by W3C.

However, many of the new CSS3 properties have been implemented in modern browsers.

CSS3 Borders

CSS3 Borders

With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border - without using a design program, like Photoshop.

In this chapter you will learn about the following border properties:

- border-radius
- box-shadow
- border-image

Browser Support

Internet Explorer 9 supports two of the new border properties.

Firefox requires the prefix -moz- for border-image.

Chrome and Safari requires the prefix -webkit- for border-image.

Opera requires the prefix -o- for border-image.

Opera supports the new border properties.

CSS3 Rounded Corners

Adding rounded corners in CSS2 was tricky. We had to use different images for each corner.

In CSS3, creating rounded corners is easy.

In CSS3, the border-radius property is used to create rounded corners:

This box has rounded corners!

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

border:2px solid #a1a1a1;

padding:10px 40px;

background: #f48935;

width:300px;

border-radius:25px;

-moz-border-radius:25px; /* Firefox 3.6 and earlier */

}

</style>

</head>

<body>

<div>The border-radius property allows you to add rounded corners to elements.</div>

</body>

</html>
```

Result

The border-radius property allows you to add rounded corners to elements.

CSS3 Box Shadow

In CSS3, the box-shadow property is used to add shadow to boxes:

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:300px;

height:100px;

background-color:blue;

-moz-box-shadow: 10px 10px 5px #888888; /* Firefox 3.6 and earlier */

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

}

</style>

</head>
```

```
<body>

<div></div>

</body>

</html>
```

Result

CSS3 Border Image

With the CSS3 border-image property you can use an image to create a border:

The border-image property allows you to specify an image as a border!

The original image used to create the border above:

Example

```
<!DOCTYPE html>

<html>

<head>

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

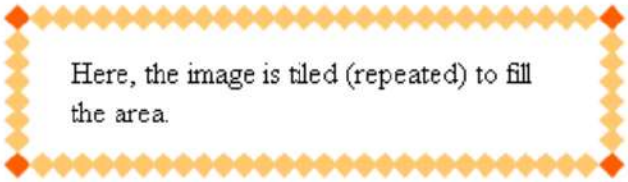
```
div
{
border-width:15px;
width:250px;
padding:10px 20px;
}
#round
{
-moz-border-image:url(border.png) 30 30 round; /* Firefox */
-webkit-border-image:url(border.png) 30 30 round; /* Safari and Chrome */
-o-border-image:url(border.png) 30 30 round; /* Opera */
border-image:url(border.png) 30 30 round;
}
#stretch
{
-moz-border-image:url(border.png) 30 30 stretch; /* Firefox */
```

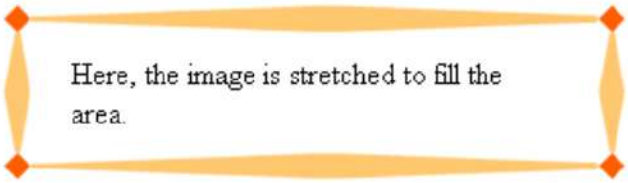
```
-webkit-border-image:url(border.png) 30 30 stretch; /* Safari and Chrome */

-o-border-image:url(border.png) 30 30 stretch; /* Opera */

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

}

</style>

</head>

<body>

<p><b>Note:</b> Internet Explorer does not support the border-image property.</p>

<p>The border-image property specifies an image to be used as a border.</p>

<div id="round">Here, the image is tiled (repeated) to fill the area.</div>

<br />

<div id="stretch">Here, the image is stretched to fill the area.</div>

<p>Here is the image used:</p>

<img src="border.png" />

</body>

</html>
```

Result

Note: Internet Explorer does not support the border-image property.

The border-image property specifies an image to be used as a border.





Here is the image used:

CSS3 Backgrounds

CSS3 Backgrounds

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

In this chapter you will learn about the following background properties:

- background-size
- background-origin

You will also learn how to use multiple background images.

Firefox 3.6 and earlier does not support the background-origin property, and requires the prefix -moz- to support the background-size property.

Safari 4 requires the prefix -webkit- to support the new background properties.

Internet Explorer 9, Firefox 4, Chrome, Safari 5 and Opera support the new background properties.

CSS3 The background-size Property

The background-size property specifies the size of the background image.

Before CSS3, the background image size was determined by the actual size of the image. In CSS3 it is possible to specify the size of the background image, which allows us to re-use background images in different contexts.

You can specify the size in pixels or in percentages. If you specify the size as a percentage, the size is relative to the width and height of the parent element.

Example 1 Resize a background image:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

body

{

background:url(img_flwr.gif);

background-size:80px 60px;


-moz-background-size:80px 60px; /* Firefox 3.6 */

background-repeat:no-repeat;

padding-top:40px;

}

</style>

</head>

<body>
```

```
<p>

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.

</p>

<p>Original image: <img src="img_flwr.gif" alt="Flowers" width="224" height="162" /></p>

</body>

</html>
```

Result

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.

Original image: 

Example 2 Stretch the background image to completely fill the content area:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

background:url(img_flwr.gif);

background-size:100% 100%;

-moz-background-size:100% 100%; /* Firefox 3.6 */

background-repeat:no-repeat;

}

</style>

</head>

<body>

<div>

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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

</div>
```

```
</body>

</html>
```

Result

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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

CSS3 The background-origin Property

The background-origin property specifies the positioning area of the background images.

The background image can be placed within the content-box, padding-box, or border-box area.

Example

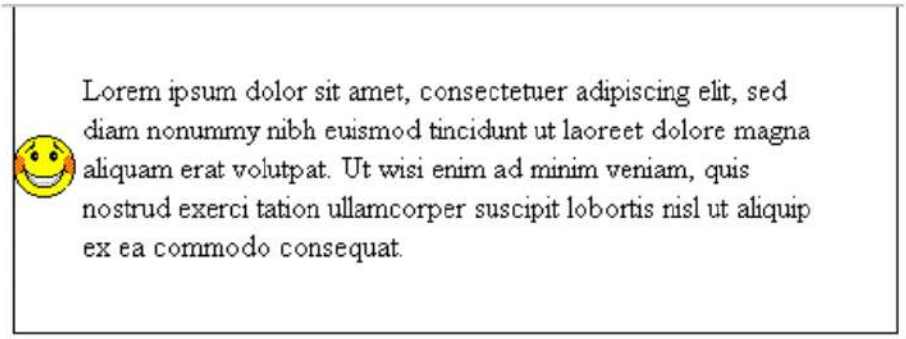
Position the background image within the content-box:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
border:1px solid black;
padding:35px;
background-image:url('smiley.gif');
background-repeat:no-repeat;
background-position:left;
}
#div1
{
background-origin:border-box;
}
#div2
{
background-origin:content-box;
}

</style>
</head>
```

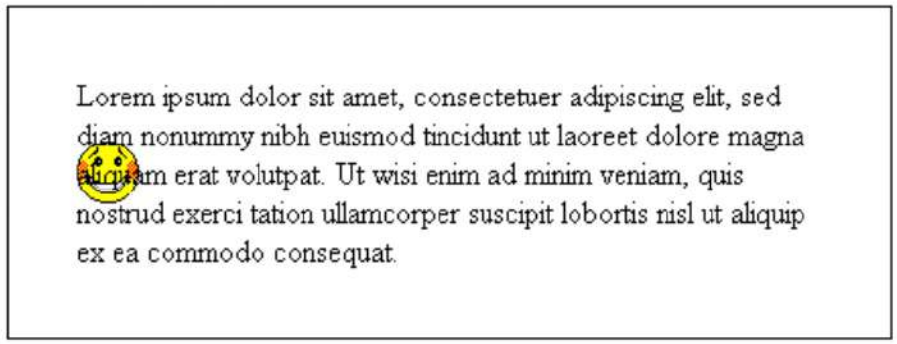
```
<body>

<p>background-origin:border-box:</p>

<div id="div1">

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.

</div>

<p>background-origin:content-box:</p>

<div id="div2">

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.

</div>

</body>

</html>
```

Result

background-origin:border-box:



background-origin:content-box:



CSS3 Multiple Background Images

CSS3 allows you to use several background images for an element.

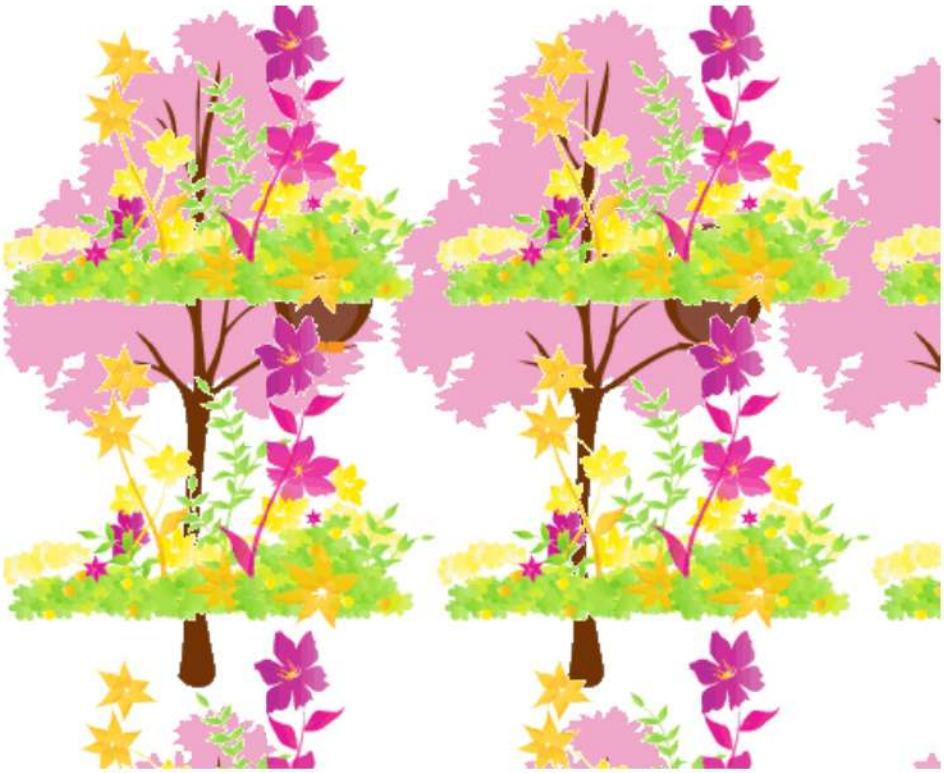
Example

Set two background images for the body element:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
body
{
background-image:url(img_flwr.gif),url(img_tree.gif);
```

```
}
</style>
</head>
<body>
</body>
</html>
```

Result



CSS3 Text Effects

CSS3 contains several new text features.

In this chapter you will learn about the following text properties:

- text-shadow
- word-wrap

Internet Explorer does not yet support the text-shadow property.

Firefox, Chrome, Safari, and Opera support the text-shadow property.

All major browsers support the word-wrap property.

CSS3 Text Shadow

In CSS3, the text-shadow property applies shadow to text.

You specify the horizontal shadow, the vertical shadow, the blur distance, and the color of the shadow:

Example

Add a shadow to a header:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

h1

{
```

```
text-shadow: 5px 5px 5px #FF0000;

}

</style>

</head>

<body>

<h1>Text-shadow effect!</h1>

<p><b>Note:</b> Internet Explorer does not support the text-shadow property.</p>

</body>

</html>
```

Result

Note: Internet Explorer does not support the text-shadow property.

CSS3 Word Wrapping

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.

In CSS3, the word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word:

Example

Allow long words to be able to break and wrap onto the next line:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

p.test

{

width:11em;

border:1px solid #000000;

word-wrap:break-word;

}

</style>

</head>

<body>

<p class="test"> This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to
the next line.</p>

</body>

</html>
```

Result

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

CSS3 Fonts

With CSS3, web designers are no longer forced to use only "web-safe" fonts.

The CSS3 @font-face Rule

Before CSS3, web designers had to use fonts that were already installed on the user's computer.

With CSS3, web designers can use whatever font he/she likes.

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

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

Browser Support

Firefox, Chrome, Safari, and Opera support fonts of type .ttf (True Type Fonts) and .otf (OpenType Fonts).

Internet Explorer 9+ supports the new @font-face rule, but it only supports fonts of type .eot (Embedded OpenType).

Note: Internet Explorer 8 and earlier versions, do not support the new @font-face rule.

Using The Font You Want

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

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

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

@font-face

{

font-family: myFirstFont;

src: url('Sansation_Light.ttf')

   ,url('Sansation_Light.eot'); /* IE9+ */

}

div

{

font-family:myFirstFont;

}

</style>

</head>

<body>
```

** **

With CSS3, websites can finally use fonts other than the pre-selected "web-safe" fonts.

** **

** Note: Internet Explorer 9+ only supports fonts of type .eot. Internet Explorer 8 and earlier, do not support the new @font-face rule. **

</body>

</html>

Result

With CSS3, websites can finally use fonts other than the pre-selected "web-safe" fonts.

Note: Internet Explorer 9+ only supports fonts of type .eot. Internet Explorer 8 and earlier, do not support the new @font-face rule.

Using Bold Text

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

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

@font-face

{

font-family: myFirstFont;
```

```
src: url('Sansation_Light.ttf')

 ,url('Sansation_Light.eot'); /* IE9+ */

}

@font-face

{

font-family: myFirstFont;

src: url('Sansation_Bold.ttf')

   ,url('Sansation_Bold.eot'); /* IE9+ */

font-weight:bold;

}

div

{

font-family:myFirstFont;

}

</style>

</head>

<body>

<div>

With CSS3, websites can <b>finally</b> use fonts other than the pre-selected "web-safe"
fonts.

</div>
```

** Note: Internet Explorer 9+ only supports fonts of type .eot. Internet Explorer 8 and earlier, do not support the new @font-face rule. **

</body>

</html>

Result

With CSS3, websites can **finally** use fonts other than the pre-selected "web-safe" fonts.

Note: Internet Explorer 9+ only supports fonts of type .eot. Internet Explorer 8 and earlier, do not support the new @font-face rule.

The file "Sansation_Bold.ttf" 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.

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	Optional. Defines how the font should be stretched. Default is "normal"

	ultra-expanded	
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 2D Transforms

CSS3 Transforms

With CSS3 transform, we can move, scale, turn, spin, and stretch elements.

How Does it Work?

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

You can transform your elements using 2D or 3D transformation.

Browser Support

Internet Explorer 9 requires the prefix -ms-.

Firefox requires the prefix -moz-.

Chrome and Safari requires the prefix -webkit- Opera requires the prefix -o-.

2D Transforms

In this chapter you will learn about the 2d transform methods:

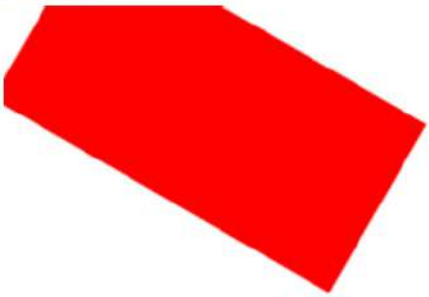
- translate()
- rotate()
- scale()
- skew()
- matrix()

Example

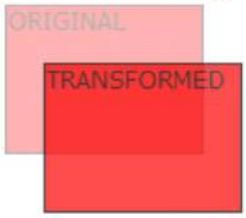
```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:200px;

height:100px;

background-color:red;

/* Rotate div */

transform:rotate(30deg);


-ms-transform:rotate(30deg); /* IE 9 */

-moz-transform:rotate(30deg); /* Firefox */

-webkit-transform:rotate(30deg); /* Safari and Chrome */

-o-transform:rotate(30deg); /* Opera */

}
```

```
</style>

</head>

<body>

<div>Hello</div>

</body>

</html>
```

Result



The translate() Method



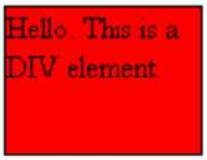
With the translate() method, the element moves from its current position, depending on the parameters given for the left (X-axis) and the top (Y-axis) position:

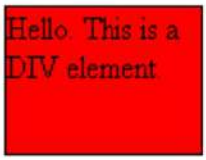
Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div
```

```
{
width:100px;
height:75px;
background-color:red;
border:1px solid black;
}
div#div2
{
transform:translate(50px,100px);
-ms-transform:translate(50px,100px); /* IE 9 */
-moz-transform:translate(50px,100px); /* Firefox */
-webkit-transform:translate(50px,100px); /* Safari and Chrome */
-o-transform:translate(50px,100px); /* Opera */
}
</style>
</head>

<body>
<div>Hello. This is a DIV element.</div>
<div id="div2">Hello. This is a DIV element.</div>
</body>
</html>
```

Result





The value translate(50px,100px) moves the element 50 pixels from the left, and 100 pixels from the top.

The rotate() Method

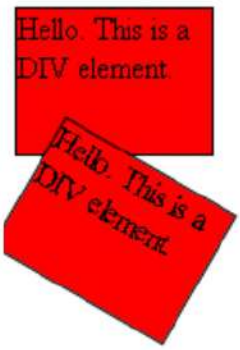
With the rotate() method, the element rotates clockwise at a given degree. Negative values are allowed and rotates the element counter-clockwise.

Example

```
<!DOCTYPE html>

<html>

<head>

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

```
div
{
width:100px;
height:75px;
background-color:red;
border:1px solid black;
}
div#div2
{
transform:rotate(30deg);
-ms-transform:rotate(30deg); /* IE 9 */
-moz-transform:rotate(30deg); /* Firefox */
-webkit-transform:rotate(30deg); /* Safari and Chrome */
-o-transform:rotate(30deg); /* Opera */
}
</style>
</head>
<body>
<div>Hello. This is a DIV element.</div>
<div id="div2">Hello. This is a DIV element.</div>
</body>
</html>
```

Result



The value rotate(30deg) rotates the element clockwise 30 degrees.

The scale() Method

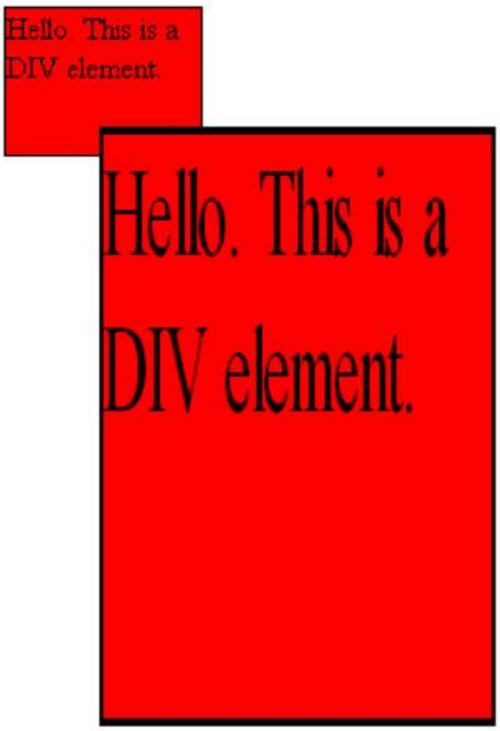
With the scale() method, the element increases or decreases the size, depending on the parameters given for the width (X-axis) and the height (Y-axis):

Example

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:75px;
background-color:red;
border:1px solid black;
}
div#div2
{
margin:100px;
transform:scale(2,4);
-ms-transform:scale(2,4); /* IE 9 */
```

```
-moz-transform:scale(2,4); /* Firefox */
-webkit-transform:scale(2,4); /* Safari and Chrome */
-o-transform:scale(2,4); /* Opera */
}
</style>
</head>
<body>

<div>Hello. This is a DIV element.</div>

<div id="div2">Hello. This is a DIV element.</div>

</body>
</html>
```

Result



The value scale(2,4) transforms the width to be twice its original size, and the height 4 times its original size.

The skew() Method

With the skew() method, the element turns in a given angle, depending on the parameters given for the horizontal (X-axis) and the vertical (Y-axis) lines:

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

}

div#div2

{

transform:skew(30deg,20deg);

-ms-transform:skew(30deg,20deg); /* IE 9 */

-moz-transform:skew(30deg,20deg); /* Firefox */
```

```
-webkit-transform:skew(30deg,20deg); /* Safari and Chrome */

-o-transform:skew(30deg,20deg); /* Opera */

}

</style>

</head>

<body>

<div>Hello. This is a DIV element.</div>

<div id="div2">Hello. This is a DIV element.</div>

</body>

</html>
```

Result

The value skew(30deg,20deg) turns the element 30 degrees around the X-axis, and 20 degrees around the Y-axis.

The matrix() Method

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

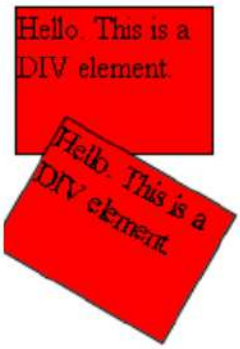
The matrix method take six parameters, containing mathematic functions, which allows you to: rotate, scale, move (translate), and skew elements.

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

}

div#div2

{

transform:matrix(0.866,0.5,-0.5,0.866,0,0);
```

```
-ms-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* IE 9 */

-moz-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* Firefox */

-webkit-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* Safari and Chrome */

-o-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /* Opera */

}

</style>

</head>

<body>

<div>Hello. This is a DIV element.</div>

<div id="div2">Hello. This is a DIV element.</div>

</body>

</html>
```

Result



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

3D Transforms

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

In this chapter you will learn about some of the 3D transform methods:

- rotateX()
- rotateY()

Browser Support

Internet Explorer and Opera does not yet support 3D transforms (They support only 2D transforms).

Firefox requires the prefix -moz-.

Chrome and Safari requires the prefix -webkit-.

The rotateX() Method

With the rotateX() method, the element rotates around its X-axis at a given degree.

Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

}
```

```
div#div2
{
transform:rotateX(120deg);
-webkit-transform:rotateX(120deg); /* Safari and Chrome */
-moz-transform:rotateX(120deg); /* Firefox */
}
</style>
</head>
<body>
<p><b>Note:</b> Internet Explorer and Opera does not support the rotateX method.</p>
<div>Hello. This is a DIV element.</div>
<div id="div2">Hello. This is a DIV element.</div>
</body>
</html>
```

Result

Note: Internet Explorer and Opera does not support the rotateX method.

Hello. This is a DIV element.

The rotateY() Method

With the rotateY() method, the element rotates around its Y-axis at a given degree.

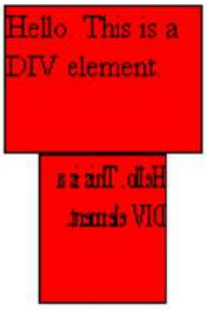
Example

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

}

div#div2

{

transform:rotateY(130deg);

-webkit-transform:rotateY(130deg); /* Safari and Chrome */

-moz-transform:rotateY(130deg); /* Firefox */
```

```
}

</style>

</head>

<body>

<p><b>Note:</b> Internet Explorer and Opera does not support the rotateY method.</p>

<div>Hello. This is a DIV element.</div>

<div id="div2">Hello. This is a DIV element.</div>

</body>

</html>
```

Result

Note: Internet Explorer and Opera does not support the rotateY method.



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

With CSS3, we can add an effect when changing from one style to another, without using Flash animations or JavaScripts.

Browser Support

Internet Explorer does not yet support the transition property.

Firefox 4 requires the prefix -moz-.

Chrome and Safari requires the prefix -webkit-.

Opera requires the prefix -o-.

How does it work?

CSS3 transitions are effects that let an element gradually change from one style to another.

To do this, you must specify two things:

- Specify the CSS property you want to add an effect to
- Specify the duration of the effect.

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

The effect will start when the specified CSS property changes value. A typical CSS property change would be when a user mouse-over an element:

Example

Specify :hover for elements:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:100px;

height:100px;

background:red;

transition:width 2s;

-moz-transition:width 2s; /* Firefox 4 */
```

```
-webkit-transition:width 2s; /* Safari and Chrome */

-o-transition:width 2s; /* Opera */

}

div:hover

{

width:300px;

}

</style>

</head>

<body>

<p><b>Note:</b> This example does not work in Internet Explorer.</p>

<div></div>

<p>Hover over the div element above, to see the transition effect.</p>

</body>

</html>
```

Result

Note: This example does not work in Internet Explorer.

Before Mouse Hover

Hover over the div element above, to see the transition effect.

Note: This example does not work in Internet Explorer.

Hover over the div element above, to see the transition effect.

On Mouse Hover

Note: When the cursor mouse out of the element, it gradually changes back to it's original style.

Multiple changes

To add a transitional effect for more than one style, add more properties, separated by commas:

Example

Add effects on the width, height, and the transformation:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:red;
transition:width 2s, height 2s;
```

```
-moz-transition:width 2s, height 2s, -moz-transform 2s; /* Firefox 4 */
-webkit-transition:width 2s, height 2s, -webkit-transform 2s; /* Safari and Chrome */
-o-transition:width 2s, height 2s, -o-transform 2s; /* Opera */
}
div:hover
{
width:200px;
height:200px;
transform:rotate(180deg);
-moz-transform:rotate(180deg); /* Firefox 4 */
-webkit-transform:rotate(180deg); /* Safari and Chrome */
-o-transform:rotate(180deg); /* Opera */
}
</style>
</head>
<body>
<p><b>Note:</b> This example does not work in Internet Explorer.</p>
<div>Hover over me to see the transition effect!</div>
</body>
</html>
```

Result

Note: This example does not work in Internet Explorer.

Hover over me to see the transition effect!

Before Mouse Hover

Note: This example does not work in Internet Explorer.

On Mouse Hover

Transition Properties

The following table lists all the transition properties:

Property	Description	CSS
transition	A shorthand property for setting the four transition properties into a single property	3
transition-property	Specifies the name of the CSS property to which the transition is applied	3
transition-duration	Defines the length of time that a transition takes. Default 0	3
transition-timing-function	Describes how the speed during a transition will be calculated. Default "ease"	3

transition-delay	Defines when the transition will start. Default 0	3

Example

Use all transition properties in one example:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:red;
transition-property:width 1s linear 2s;
/* Firefox 4 */
-moz-transition:width 1s linear 2s;
/* Safari and Chrome */
-webkit-transition:width 1s linear 2s;
/* Opera */
-o-transition:width 1s linear 2s;
}
```

```
div:hover
{
width:200px;
}
</style>
</head>
<body>
<p><b>Note:</b> This example does not work in Internet Explorer.</p>
<div></div>
<p>Hover over the div element above, to see the transition effect.</p>
<p><b>Note:</b> The transition effect will wait 2 seconds before starting.</p>
</body>
</html>
```

Result

Note: This example does not work in Internet Explorer.

Before Mouse Hover

Hover over the div element above, to see the transition effect.

Note: The transition effect will wait 2 seconds before starting.

Note: This example does not work in Internet Explorer.

On Mouse Hover

Hover over the div element above, to see the transition effect.

Note: The transition effect will wait 2 seconds before starting.

Example

The same transition effects as above, using the shorthand transition property:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:red;
transition-property:width 1s linear 2s;
/* Firefox 4 */
-moz-transition:width 1s linear 2s;
/* Safari and Chrome */
-webkit-transition:width 1s linear 2s;
```

```
/* Opera */
-o-transition:width 1s linear 2s;
}
div:hover
{
width:200px;
}
</style>
</head>
<body>
<p><b>Note:</b> This example does not work in Internet Explorer.</p>
<div></div>
<p>Hover over the div element above, to see the transition effect.</p>
<p><b>Note:</b> The transition effect will wait 2 seconds before starting.</p>
</body>
</html>
```

Result

Note: This example does not work in Internet Explorer.

Before Mouse Hover

Hover over the div element above, to see the transition effect.

Note: The transition effect will wait 2 seconds before starting.

Note: This example does not work in Internet Explorer.

On Mouse Hover

Hover over the div element above, to see the transition effect.

Note: The transition effect will wait 2 seconds before starting.

CSS3 Animations

With CSS3, we can create animations, which can replace animated images, Flash animations, and JavaScripts in many web pages.

CSS3 @keyframes Rule

To create animations in CSS3, you will have to learn about the @keyframes rule.

The @keyframes rule is where the animation is created. Specify a CSS style inside the @keyframes rule and the animation will gradually change from the current style to the new style.

Browser Support

Internet Explorer does not yet support the @keyframes rule or the animation property.

Firefox requires the prefix -moz-, Chrome and Safari require the prefix -webkit-, and Opera require the prefix -o-.

CSS3 animation

When the animation is created in the @keyframe, bind it to a selector, otherwise the animation will have no effect.

Bind the animation to a selector by specifying at least these two CSS3 animation properties:

- Specify the name of the animation
- Specify the duration of the animation

Example

Binding the "myfirst" animation to a div element, duration: 5 seconds:

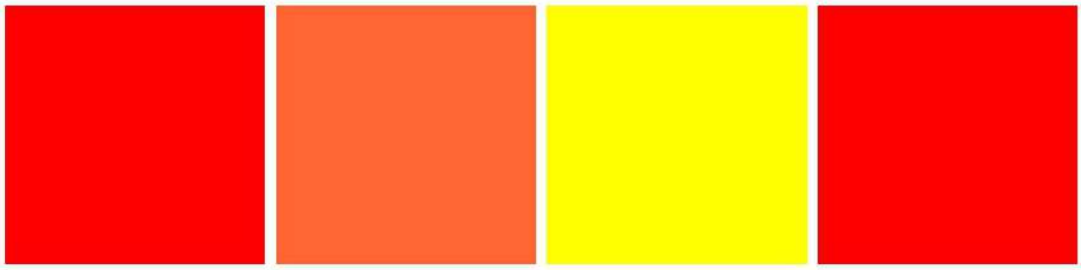
```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

width:100px;

height:100px;

background:red;

animation:myfirst 5s;

-moz-animation:myfirst 5s; /* Firefox */

-webkit-animation:myfirst 5s; /* Safari and Chrome */
```

```
-o-animation:myfirst 5s; /* Opera */
}
@keyframes myfirst
{
from {background:red;}
to {background:yellow;}
}
@-moz-keyframes myfirst /* Firefox */
{
from {background:red;}
to {background:yellow;}
}
@-webkit-keyframes myfirst /* Safari and Chrome */
{
from {background:red;}
to {background:yellow;}
}
@-o-keyframes myfirst /* Opera */
{
from {background:red;}
to {background:yellow;}
}
</style>
```

```
</head>

<body>

<p><b>Note:</b> This example does not work in Internet Explorer.</p>

<div></div>

</body>

</html>
```

Result

It will Animate and change color from Red to Yellow in 5 seconds



Note: You must define the name and the duration of the animation. If duration is omitted, the animation will not run, because the default value is 0.

What are Animations in CSS3?

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

You can change as many styles you want, as many times you want.

Specify when the change will happen in percent, or the keywords "from" and "to", which is the same as 0% and 100%.

0% is the beginning of the animation, 100% is when the animation is complete.

For best browser support, you should always define both the 0% and the 100% selectors.

CSS3 Animation Properties

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

Property	Description	CSS
@keyframes	Specifies the animation	3
animation	A shorthand property for all the the animation properties, except the animation-play-state property	3
animation-name	Specifies the name of the @keyframes animation	3
animation-duration	Specifies how many seconds or milliseconds an animation takes to complete one cycle. Default 0	3
animation-timing-function	Describes how the animation will progress over one cycle of its duration. Default "ease"	3
animation-delay	Specifies when the animation will start. Default 0	3
animation-iteration-count	Specifies the number of times an animation is played. Default 1	3
animation-direction	Specifies whether or not the animation should play in reverse on alternate cycles. Default "normal"	3
animation-play-state	Specifies whether the animation is running or paused. Default "running"	3

Example

Run an animation called myfirst, with all the animation properties set:

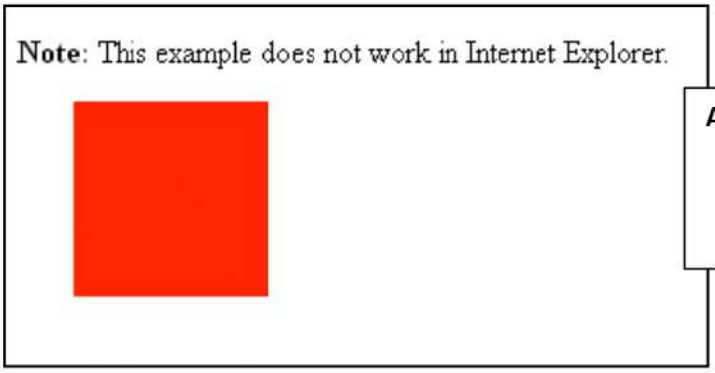
```
<!DOCTYPE html>

<html>

<head>

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

```
div
{
width:100px;
height:100px;
background:red;
position:relative;
animation:myfirst 5s linear 2s infinite alternate;
/* Firefox: */
-moz-animation:myfirst 5s linear 2s infinite alternate;
/* Safari and Chrome: */
-webkit-animation:myfirst 5s linear 2s infinite alternate;
/* Opera: */
-o-animation:myfirst 5s linear 2s infinite alternate;
}
@keyframes myfirst
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:200px; top:0px;}
50%  {background:blue; left:200px; top:200px;}
75%  {background:green; left:0px; top:200px;}
100% {background:red; left:0px; top:0px;}
}
```

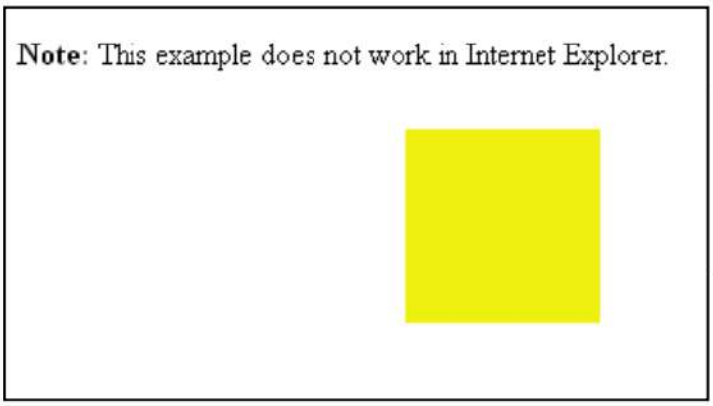
```
@-moz-keyframes myfirst /* Firefox */
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:200px; top:0px;}
50%  {background:blue; left:200px; top:200px;}
75%  {background:green; left:0px; top:200px;}
100% {background:red; left:0px; top:0px;}
}
@-webkit-keyframes myfirst /* Safari and Chrome */
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:200px; top:0px;}
50%  {background:blue; left:200px; top:200px;}
75%  {background:green; left:0px; top:200px;}
100% {background:red; left:0px; top:0px;}
}
@-o-keyframes myfirst /* Opera */
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:200px; top:0px;}
50%  {background:blue; left:200px; top:200px;}
75%  {background:green; left:0px; top:200px;}
100% {background:red; left:0px; top:0px;}
```

```
}

</style>

</head>

<body>

<p><b>Note:</b> This example does not work in Internet Explorer.</p>

<div></div>

</body>

</html>
```

Result



At the Beginning i.e, at 0% at left:0px; top:0px;



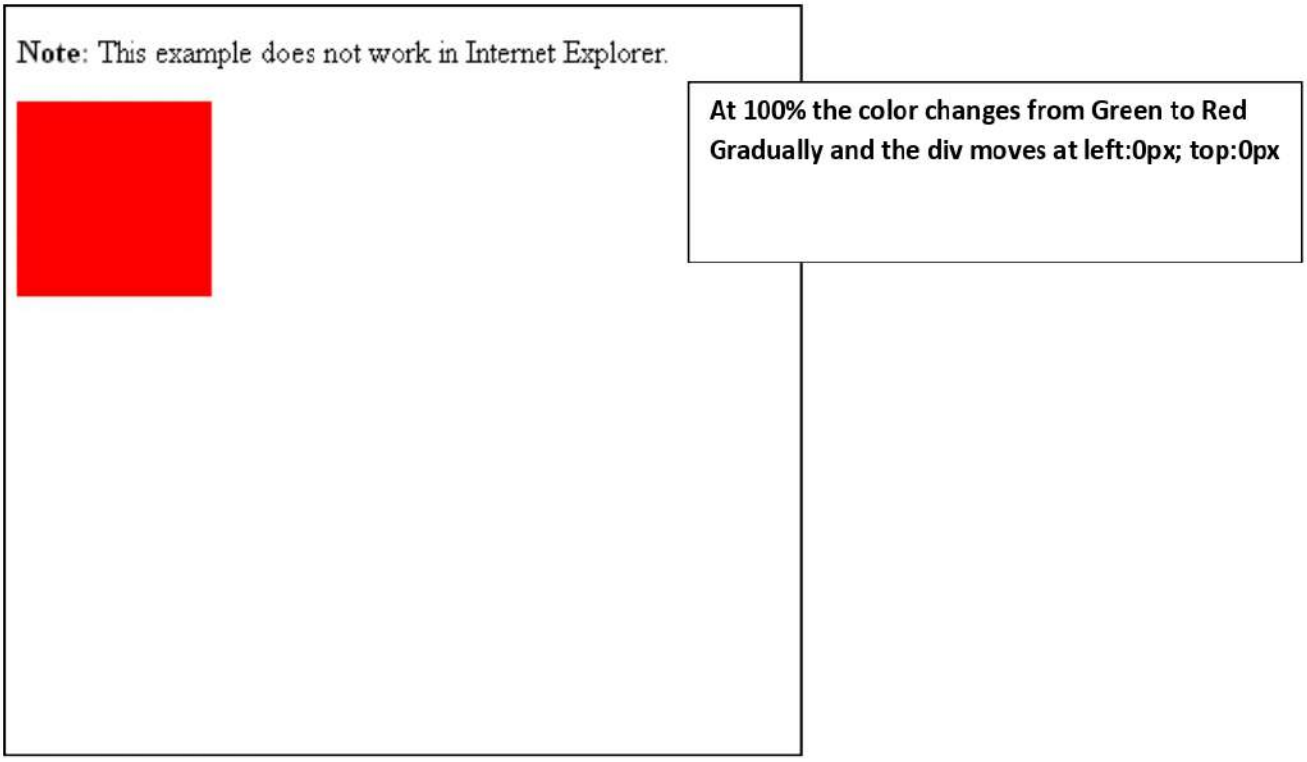
At 25% the color changes from Red to Yellow Gradually and the div moves at left:200px; top:0px;

Note: This example does not work in Internet Explorer.

At 50% the color changes from Yellow to Blue Gradually and the div moves at left:200px; top:200px

Note: This example does not work in Internet Explorer.

At 75% the color changes from Blue to Green Gradually and the div moves at left:0px; top:200px



CSS3 Multiple Columns

With CSS3, you can create multiple columns for laying out text - like in newspapers!

In this chapter you will learn about the following multiple column properties:

- column-count
- column-gap
- column-rule

Browser Support

Internet Explorer does not yet support the multiple columns properties.

Firefox requires the prefix -moz-.

Chrome and Safari require the prefix -webkit-.

CSS3 Create Multiple Columns

The column-count property specifies the number of columns an element should be divided into:

Example

Divide the text in a div element into three columns:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
.newspaper
{
-moz-column-count:3; /* Firefox */
-webkit-column-count:3; /* Safari and Chrome */
column-count:3;
}
</style>
</head>
<body>
<p><b>Note:</b> Internet Explorer does not support the column-count property.</p>
```

```
<div class="newspaper">

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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

</div>

</body>

</html>
```

Result

Note: Internet Explorer does not support the column-count property.

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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

```
<div>The resize property specifies whether or not an element is resizable by the user.</div>

</body>

</html>
```

Result

Note: Firefox 4+, Safari, and Chrome support the resize property.

The resize property specifies whether or not an element is resizable by the user.

This Div Can Be Resized By the User in Browser.

CSS3 Box Sizing

The box-sizing property allows you to define certain elements to fit an area in a certain way:

Example

Specify two bordered boxes side by side:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div.container

{

width:30em;

border:1em solid;
```

```
}
div.box
{
box-sizing:border-box;
-moz-box-sizing:border-box; /* Firefox */
-webkit-box-sizing:border-box; /* Safari */
width:50%;
border:1em solid red;
float:left;
}
</style>
</head>
<body>
<div class="container">
<div class="box">This div occupies the left half.</div>
<div class="box">This div occupies the right half.</div>
</div>
</body>
</html>
```

Result

CSS3 Outline Offset

The outline-offset property offsets an outline, and draws it beyond the border edge.

Outlines differ from borders in two ways:

- Outlines do not take up space
- Outlines may be non-rectangular

Example

Specify an outline 15px outside the border edge:

```
<!DOCTYPE html>

<html>

<head>

<style type="text/css">

div

{

margin:20px;

width:150px;

padding:10px;

height:70px;

border:2px solid black;
```

```
outline:2px solid red;

outline-offset:15px;

}

</style>

</head>

<body>

<p><b>Note:</b> Internet Explorer and Opera does not support the outline-offset property.</p>

<div>This div has an outline border 15px outside the border edge.</div>

</body>

</html>
```

Result

Note: Internet Explorer and Opera does not support the outline-offset property.

This div has an outline border 15px outside the border edge.