Introduction

The development of **style sheet** was to make the markup language more impressive. It was discovered around 1980s in the beginning of the **SGML**.

The third level of **CSS** was started to develop around 1998. And till 2009, it was under development. The first working draft of **CSS3** came in 19-01-2001. And since the first introduction still it is under construction.

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".

Some of the most important CSS3 modules are:

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

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

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 border-radius and box-shadow.

Firefox supports all of the new border properties.

Chrome and Safari support border-radius and box-shadow, but require the prefix -webkit- for border-image.

Opera supports border-radius and box-shadow, but requires the prefix -o- for border-image.

CSS3 Rounded Corners

Adding rounded corners in CSS2 was tricky.

We had to use different images for each corner.

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

Example

```
<!DOCTYPE html>
<html>
        <head>
                <style>
                        div
                        {
                                border:2px solid #a1a1a1;
                                padding:10px 40px;
                                background:#dddddd;
                                width:300px;
                                border-radius:25px;
                                -moz-border-radius:25px;  /* Old Firefox */
                        }
                </style>
        </head>
        <body>

                <h1>border Radious</h1>
                <div>The border-radius property allows you to add rounded
                corners to elements.</div>
        </body>
</html>
```

CSS3 Box Shadow

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

With inset and rgba

Box-shadow:inset Horizontal-Length Vertical-Length Blur-Radius Spread rgba(r,g,b,opacity);

Example

```
-webkit-box-shadow: inset 15px 15px 100px 100px rgba(15, 15, 15, .5);

box-shadow: inset 15px 15px 100px 100px rgba(15, 15, 15, .5);
```

Without inset and hex

box-shadow:Horizontal-Length Vertical-Length Blur-Radius Spread #rrggbb;

Example

```
-webkit-box-shadow: 15px 15px 100px 100px #b5b5b5;

box-shadow: 15px 15px 100px 100px #b5b5b5;
```

Example

```
<!DOCTYPE html>
<html>
	<head>
		<style>
			div
			{
				width:300px;
				height:100px;
				background-color:yellow;
				-webkit-box-shadow: 15px 15px 50px 50px
				#b5b5b5;
				box-shadow: 15px 15px 50px 50px #b5b5b5;

			}
		</style>
	</head>
	<body>

		<div>Marlabs</div>
```

```
            </body>
        </html>
```

CSS3 Border Image

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
        div
        {
        border:15px solid transparent;
        width:250px;
        padding:10px 20px;
        }

        #round
        {
        -moz-border-image:url(border1.jpg) 30 30 round; /* Old Firefox */
        -webkit-border-image:url(border1.jpg) 30 30 round; /* Safari and Chrome

        */
        -o-border-image:url(border1.jpg) 30 30 round; /* Opera */
        border-image:url(border1.jpg) 30 30 round;
        }

        #stretch
        {
        -moz-border-image:url(border1.jpg) 30 30 stretch; /* Old Firefox */
        -webkit-border-image:url(border1.jpg) 30 30 stretch; /* Safari and Chrome

        */
        -o-border-image:url(border1.jpg) 30 30 stretch; /* Opera */
        border-image:url(border1.jpg) 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="border1.jpg">

</body>
</html>
```

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

Browser Support

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.

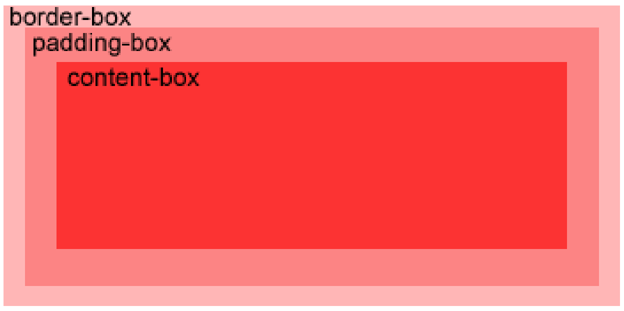
If you specify the size as a percentage, the size is relative to the width and height of the parent element.

```
<!DOCTYPE html>
<html>
<head>
<style>
        body
        {
        background:url(border1.jpg);
        background-size:80px 60px;
        -moz-background-size:80px 60px; /* Old Firefox */
        background-repeat:no-repeat;
        padding-top:40px;
        }
</style>
</head>
<body>
        <p>
        marlabs
        </p>

        <p>Original image: <img src="border1.jpg" alt="Flowers"
        width="224"

        height="162"></p>

</body>
</html>
```

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

```
<!DOCTYPE html>
<html>
<head>
<style>
        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. </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. </div>

</body>
</html>
```

CSS3 Multiple Background Images

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

```
body
{
        background-image:url(img_flwr.gif),url(img_tree.gif);
}
```

CSS3 Text Effects

CSS3 contains several new text features.

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

- text-shadow
- word-wrap

Browser Support

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

```
<!DOCTYPE html>
<html>
<head>
        <style>
                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>
```

CSS3 Word Wrapping

If a word is too long to fit within an area, it expands outside:

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:

```
<!DOCTYPE html>
<html>
<head>
<style>
        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>
```

CSS3 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).

Example

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

div
{
font-family:myFirstFont;
}
</style>
```

Using Bold Text

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

```
     @font-face
     {
     font-family: myFirstFont;
     src: url('Sansation_Bold.ttf'),
          url('Sansation_Bold.eot'); /* IE9+ */
     font-weight:bold;
     }
```

```
<!DOCTYPE html>

<html>

    <head>
     <title>Title Name will go here</title>
          <link
href='http://fonts.googleapis.com/css?family=Cherry+Cream
+Soda|Ropa+Sans|Butcherman|Rochester|Open+Sans+Condensed:
300' rel='stylesheet' type='text/css'/>
          <style>

               .font1
               {
                    font-family: 'Cherry Cream Soda',
cursive;
                    font-size: 14px;
                    color: yellow;
                    line-height: 1.3em;
               }

               .font2
               {
                    font-family: 'Ropa Sans', sans-serif;
                    font-size: 14px;
                    color: green;
                    line-height: 1.3em;
               }

               .font3
               {
                    font-family: 'Butcherman', cursive;
                    font-size: 14px;
                    color: red;
                    line-height: 1.3em;
               }
```

```
                .font4
                {
                     font-family: 'Rochester', cursive;
                     font-size: 14px;
                     color: blue;
                     line-height: 1.3em;
                }

                .font5
                {
                     font-family: 'Open Sans Condensed',
sans-serif;
                     font-size: 14px;
                     color: pink;
                     line-height: 1.3em;
                }

          </style>

    </head>
    <body>

          <p class="font1">THIS LINE IS CONTAINING THE
DIFFERENT FONT</p>
          <p class="font2">THIS LINE IS CONTAINING THE
DIFFERENT FONT</p>
          <p class="font3">THIS LINE IS CONTAINING THE
DIFFERENT FONT</p>
          <p class="font4">THIS LINE IS CONTAINING THE
DIFFERENT FONT</p>
          <p class="font5">THIS LINE IS CONTAINING THE
DIFFERENT FONT</p>

    </body>

</html>
```

CSS3 2D Transforms

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

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()

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

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:

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
        div
        {
                width:100px;
                height:200px;
                border:2px solid blue;
                word-wrap:break-word;
                position:absolute;
                top:50px;
                left:50px;
        }
```

```
div:hover
{
transform: translate(100px,100px);
-ms-transform: translate(100px,100px); /* IE 9 */
-webkit-transform: translate(100px,100px); /* Safari and

Chrome */
-o-transform: translate(100px,100px); /* Opera */
-moz-transform: translate(100px,100px); /* Firefox */
}

</style>
</head>
<body>
        <div>
         Marlabs
        </div>
</body>
</html>
```

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.

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
            div
            {
                width:100px;
                height:200px;
                border:2px solid blue;
                word-wrap:break-word;
                position:absolute;
                top:50px;
                left:50px;
            }

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


</style>
</head>
<body>
            <div>
             Marlabs Inc
            </div>
</body>
</html>
```

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):

```
div
{
transform: scale(2,4);
-ms-transform: scale(2,4); /* IE 9 */
-webkit-transform: scale(2,4); /* Safari and Chrome
*/
-o-transform: scale(2,4); /* Opera */
-moz-transform: scale(2,4); /* Firefox */
}
```

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
        div
        {
                width:100px;
                height:200px;
                border:2px solid blue;
                word-wrap:break-word;
                position:absolute;
                top:50px;
                left:50px;
        }

div:hover
{
transform: scale(2,4);
-ms-transform: scale(2,4); /* IE 9 */
-webkit-transform: scale(2,4); /* Safari and Chrome */
-o-transform: scale(2,4); /* Opera */
-moz-transform: scale(2,4); /* Firefox */
}
</style>
</head>
<body>
        <div>
         Marlabs
SoftwareSoftwareSoftwareSoftwareSoftwareSoftware
        </div>
</body>
</html>
```

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:

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
        div
        {
                width:100px;
                height:200px;
                border:2px solid blue;
                word-wrap:break-word;
                position:absolute;
                top:50px;
                left:50px;
        }
```

```
div:hover
{
transform: skew(30deg,20deg);
-ms-transform: skew(30deg,20deg); /* IE 9 */
-webkit-transform: skew(30deg,20deg); /* Safari and Chrome

*/
-o-transform: skew(30deg,20deg); /* Opera */
-moz-transform: skew(30deg,20deg); /* Firefox */
}
</style>
</head>
<body>
        <div>
         Marlabs
        </div>
</body>
</html>
```

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.

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

CSS3 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.

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

The rotateY() Method

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

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

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.

```
div
{
transition: width 2s;
-moz-transition: width 2s; /* Firefox 4 */
-webkit-transition: width 2s; /* Safari and Chrome
*/
-o-transition: width 2s; /* Opera */
}
```

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:

Specify :hover for <div> elements:

```
div:hover
{
width:300px;
}
```

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
	div
	{
		width:100px;
		height:200px;
		border:2px solid blue;
		word-wrap:break-word;
		position:absolute;
		top:50px;
		left:50px;
	}


div
{
transition: width 2s;
-moz-transition: width 5s; /* Firefox 4 */
-webkit-transition: width 2s; /* Safari and Chrome */
-o-transition: width 2s; /* Opera */
}
div:hover
{
	width:50px;
}
</style>
</head>
<body>
	<div>
	 Marlabs
	</div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
    <head>
       <title>Title Name will go here</title>
              <style>
                     div.swapnil
                     {
                            width: 20px;
                            height: 20px;
                            margin: 20px auto;
                     }
                     div.swapnil div.raja img
                     {
                            width: 20px;
                            height: 20px;
                            color: #fff;
                            padding: 10px;
                            border-radius: 5px;
                            margin-left: 0;
                            -webkit-transition: 3s linear;
                            -moz-transition: 3s linear;
                            -o-transition: 3s linear;
                            -ms-transition: 3s linear;
                            transition: 3s linear;
                     }

                     div.swapnil:hover div.raja img
                     {
                            width: 240px;
                            height: 220px;
                            -webkit-transform:rotate(360deg);
                            -moz-transform:rotate(360deg);
                            -o-transform:rotate(360deg);
                            -ms-transform:rotate(360deg);
                            transform:rotate(360deg);
                     }
              </style>
    </head>
    <body>
              <p>Hover on object to see it in action</p>
              <div class="swapnil">
                     <div class="raja">
                     <img src="media/star.png" width="20" height="20"/>
                     </div>
              </div>
    </body>
</html>
```

Multiple changes

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

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

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
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>
```

```
<!DOCTYPE html>

<html>

    <head>

       <title>Title Name will go here</title>

              <style>

                     #example
                     {
                            position:relative;
                            width:530px;
                            height:530px;
                            margin:0 auto 10px;
                            padding:10px;
                     }

                     .childbox
                     {
                            font-size:12px;
                            position:relative;
                            width:60px;
                            height:60px;
                            margin-bottom:10px;
                            background-color:#ccc;
                     }

                     .childbox p
                     {
                            text-align:center;
                            padding-top:10px;
                     }

                     #ease.childbox
                     {
                            -webkit-transition: all 4s ease;
                            -moz-transition: all 4s ease;
                            -o-transition: all 4s ease;
                            transition: all 4s ease;
                            border:1px solid #ff0000;
                     }

                     #ease_in.childbox
                     {
                            -webkit-transition: all 4s ease-in;
                            -moz-transition: all 4s ease-in;
                            -o-transition: all 4s ease-in;
                            transition: all 4s ease-in;
                            border:1px solid #00ffff;
                     }

                     #ease_out.childbox
                     {
                            -webkit-transition: all 4s ease-out;
                            -moz-transition: all 4s ease-out;
                            -o-transition: all 4s ease-out;
                            transition: all 4s ease-out;
                            border:1px solid #f5f5f5;
```

```
                    }

                    #ease_in_out.childbox
                    {
                           -webkit-transition: all 4s ease-in-out;
                           -moz-transition: all 4s ease-in-out;
                           -o-transition: all 4s ease-in-out;
                           transition: all 4s ease-in-out;
                           border:1px solid #f209f3;
                    }

                    #linear.childbox
                    {
                           -webkit-transition: all 4s linear;
                           -moz-transition: all 4s linear;
                           -o-transition: all 4s linear;
                           transition: all 4s linear;
                           border:1px solid #ddddff;
                    }

                    #custom.childbox
                    {
                           -webkit-transition: all 4s cubic-bezier(1.000, 
0.835, 0.000, 0.945);
                           -moz-transition: all 4s cubic-bezier(1.000, 0.835, 
0.000, 0.945);
                           -o-transition: all 4s cubic-bezier(1.000, 0.835, 
0.000, 0.945);
                           transition: all 4s cubic-bezier(1.000, 0.835, 
0.000, 0.945);
                           border:1px solid #cfdf44;
                    }

                    #negative.childbox
                    {
                           -webkit-transition: all 4s cubic-bezier(1.000, -
0.530, 0.405, 1.425);
                           -moz-transition: all 4s cubic-bezier(1.000, -0.530, 
0.405, 1.425);
                           -o-transition: all 4s cubic-bezier(1.000, -0.530, 
0.405, 1.425);
                           transition: all 4s cubic-bezier(1.000, -0.530, 
0.405, 1.425);
                           border:1px solid #000;
                    }

                    #example:hover .childbox, #example.hover_effect .childbox
                    {
                           -webkit-border-radius:30px;
                           -moz-border-radius:30px;
                           border-radius:30px;
                           -webkit-transform: rotate(720deg);
                           -moz-transform: rotate(720deg);
                           -o-transform: rotate(720deg);
                           -ms-transform: rotate(720deg);
                           transform: rotate(720deg);
                           margin-left:420px;
                           background-color:#fff;
                    }

             </style>
```

```html
    </head>
    <body>

              <p>Hover on object to see it in action</p>
              <div id="example">
                <div id="ease" class="childbox"><p>CSS3</p></div>
                <div id="ease_in" class="childbox"><p>CSS3</p></div>
                <div id="ease_out" class="childbox"><p>CSS3</p></div>
                <div id="ease_in_out" class="childbox"><p>CSS3</p></div>
                <div id="linear" class="childbox"><p>CSS3</p></div>
                <div id="custom" class="childbox"><p>CSS3</p></div>
                <div id="negative" class="childbox"><p>CSS3</p></div>
              </div>

    </body>

</html>
```

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

```
div
{
transition-property: width;
transition-duration: 1s;
transition-timing-function: linear;
transition-delay: 2s;
/* Firefox 4 */
-moz-transition-property:width;
-moz-transition-duration:1s;
-moz-transition-timing-function:linear;
-moz-transition-delay:2s;
/* Safari and Chrome */
-webkit-transition-property:width;
-webkit-transition-duration:1s;
-webkit-transition-timing-function:linear;
-webkit-transition-delay:2s;
/* Opera */
-o-transition-property:width;
-o-transition-duration:1s;
-o-transition-timing-function:linear;
-o-transition-delay:2s;
}
```

CSS3 Animations

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

CSS3 @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

Supported Browser

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-.

```
@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;}
}
```

CSS3 animation

When the animation is created in the @keyframes, 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

```
div

{

animation: myfirst 5s;

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

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

-o-animation: myfirst 5s; /* Opera */

}
```

Example

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

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.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
width:100px;
height:100px;
background:red;
position:relative;
animation:myfirst 5s;
-moz-animation:myfirst 5s; /* Firefox */
-webkit-animation:myfirst 5s; /* Safari and Chrome */
-o-animation:myfirst 5s; /* Opera */
}

@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>
```

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

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
width:100px;
height:100px;
background:red;
position:relative;
animation-name:myfirst;
animation-duration:5s;
animation-timing-function:linear;
animation-delay:2s;
animation-iteration-count:infinite;
animation-direction:alternate;
animation-play-state:running;
/* Firefox: */
-moz-animation-name:myfirst;
-moz-animation-duration:5s;
-moz-animation-timing-function:linear;
-moz-animation-delay:2s;
-moz-animation-iteration-count:infinite;
-moz-animation-direction:alternate;
-moz-animation-play-state:running;
/* Safari and Chrome: */
-webkit-animation-name:myfirst;
-webkit-animation-duration:5s;
-webkit-animation-timing-function:linear;
-webkit-animation-delay:2s;
-webkit-animation-iteration-count:infinite;
-webkit-animation-direction:alternate;
-webkit-animation-play-state:running;
/* Opera: */
-o-animation-name:myfirst;
-o-animation-duration:5s;
-o-animation-timing-function:linear;
-o-animation-delay:2s;
-o-animation-iteration-count:infinite;
-o-animation-direction:alternate;
-o-animation-play-state:running;
}

@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>
```

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-.

```
div
{
-moz-column-count:3; /* Firefox */
-webkit-column-count:3; /* Safari and Chrome */
column-count:3;
}
```

CSS3 Specify the Gap between Columns

The column-gap property specifies the gap between the columns:

```
div
{
-moz-column-gap:40px; /* Firefox */
-webkit-column-gap:40px; /* Safari and Chrome */
column-gap:40px;
}
```

CSS3 Column Rules

The column-rule property sets the width, style, and color of the rule between columns.

```
div
{
-moz-column-rule:3px outset #ff00ff; /* Firefox */
-webkit-column-rule:3px outset #ff00ff; /* Safari
and Chrome */
column-rule:3px outset #ff00ff;
}
```

New Multiple Columns Properties

Property	Description	CSS
column-count	Specifies the number of columns an element should be divided into	3
column-fill	Specifies how to fill columns	3
column-gap	Specifies the gap between the columns	3
column-rule	A shorthand property for setting all the column-rule-* properties	3
column-rule-color	Specifies the color of the rule between columns	3
column-rule-style	Specifies the style of the rule between columns	3
column-rule-width	Specifies the width of the rule between columns	3
column-span	Specifies how many columns an element should span across	3
column-width	Specifies the width of the columns	3
columns	A shorthand property for setting column-width and column-count	3

CSS3 User Interface

In CSS3, some of the new user interface features are resizing elements, box sizing, and outlining.

In this chapter you will learn about the following user interface properties:

- resize
- box-sizing
- outline-offset

Browser Support

The resize property is supported in Firefox 4+, Chrome, and Safari.

The box-sizing is supported in Internet Explorer, Chrome, and Opera. Firefox requires the prefix -moz-. Safari requires the prefix -webkit-.

The outline property is supported in all major browsers, except Internet Explorer.

CSS3 Resizing

In CSS3, the resize property specifies whether or not an element should be resizable by the user.

```
div
{
resize:both;
overflow:auto;
}
```

CSS3 Box Sizing

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

```
<!DOCTYPE html>
<html>
<head>
<style>
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>
```

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

```
div
{
border:2px solid black;
outline:2px solid red;
outline-offset:15px;
}
```

Overview of CSS 3 selector syntax

Selector type	Pattern	Description
Substring matching attribute selector	E[att^=”val”]	Matches any E element whose `att` attribute value begins with “val”.
Substring matching attribute selector	E[att$=”val”]	Matches any E element whose `att` attribute value ends with “val”.
Substring matching attribute selector	E[att*=”val”]	Matches any E element whose `att` attribute value contains the substring “val”.
Structural pseudo-class	E:root	Matches the document’s root element. In HTML, the root element is always the HTML element.
Structural pseudo-class	E:nth-child(n)	Matches any E element that is the n-th child of its parent.
Structural pseudo-class	E:nth-last-child(n)	Matches any E element that is the n-th child of its parent, counting from the last child.
Structural pseudo-class	E:nth-of-type(n)	Matches any E element that is the n-th sibling of its type.
Structural pseudo-class	E:nth-last-of-type(n)	Matches any E element that is the n-th sibling of its type, counting from the last sibling.
Structural pseudo-class	E:last-child	Matches any E element that is the last child of its parent.
Structural pseudo-class	E:first-of-type	Matches any E element that is the first sibling of its type.
Structural pseudo-class	E:last-of-type	Matches any E element that is the last sibling of its type.
Structural pseudo-class	E:only-child	Matches any E element that is the only child of its parent.
Structural pseudo-class	E:only-of-type	Matches any E element that is the only sibling of its type.
Structural pseudo-class	E:empty	Matches any E element that has no children (including text nodes).
Target pseudo-class	E:target	Matches an E element that is the target of the referring URL.
UI element states pseudo-class	E:enabled	Matches any user interface element (form control) E that is enabled.
UI element states pseudo-class	E:disabled	Matches any user interface element (form control) E that is disabled.
UI element states	E:checked	Matches any user interface element (form

Selector type	Pattern	Description
pseudo-class		control) E that is checked.
UI element fragments pseudo-element	E::selection	Matches the portion of an element E that is currently selected or highlighted by the user.
Negation pseudo-class	E:not(s)	Matches any E element that does not match the simple selector s.
General sibling combinator	E ~ F	Matches any F element that is preceded by an E element.

Overview of CSS 2.1 selector syntax

Selector type	Pattern	Description
Universal	*	Matches any element.
Type	E	Matches any E element.
Class	.info	Matches any element whose `class` attribute contains the value `info`.
ID	#footer	Matches any element with an `id` equal to `footer`.
Descendant	E F	Matches any F element that is a descendant of an E element.
Child	E > F	Matches any F element that is a child of an E element.
Adjacent	E + F	Matches any F element immediately preceded by a sibling element E.
Attribute	E[att]	Matches any E element that has an `att` attribute, regardless of its value.
Attribute	E[att=val]	Matches any E element whose `att` attribute value is exactly equal to `val`.
Attribute	E[att~=val]	Matches any E element whose `att` attribute value is a list of space-separated values, one of which is exactly equal to `val`.
Attribute	E[att|=val]	Matches any E element whose `att` attribute has a hyphen-separated list of values beginning with `val`.
The :first-child pseudo-class	E:first-child	Matches element E when E is the first child of its parent.
The link pseudo-classes	E:link E:visited	Matches not yet visited (:link) or already visited (:visited) links.
The dynamic	E:active	Matches E during certain user actions.

Overview of CSS 2.1 selector syntax

Selector type	Pattern	Description
pseudo-classes	E:hover E:focus	
The :language pseudo-class	E:lang(c)	Matches elements of type E that are in language c.
The :first-line pseudo-element	E:first-line	Matches the contents of the first formatted line of element E.
The :first-letter pseudo-element	E:first-letter	Matches the first letter of element E.
The :before and :after pseudo-elements	E:before E:after	Used to insert generated content before or after an element's content.

descendant	An element that is the child, grandchild or later descendant of an element in the document tree.
ancestor	An element that is the parent, grandparent or earlier ancestor of an element in the document tree.
child	The direct descendant of an element. No other elements may come between the two in the document tree.
parent	The direct ancestor of an element. No other element may come between the two in the document tree.
sibling	An element that has the same parent as the current element.