1. CSS3 Introduction

CSS3 is the latest standard for CSS.

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

This section teaches you about the new features in CSS3!

1.1. 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

1.2. CSS3 Recommendation

The CSS3 specification is still under development by W3C.

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

2. 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

2.1. Browser Support

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

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

Property					
border-radius	9.0	5.0 4.0 -webkit-	4.0 3.0 -moz-	5.0 3.1 -webkit-	10.5
box-shadow	9.0	10.0 4.0 -webkit-	4.0 3.5 -moz-	5.1 3.1 -webkit-	10.5
border-image	11.0	16.0 4.0 -webkit-	15.0 3.5 -moz-	6.0 3.1 -webkit-	15.0 10.5 -o-

2.2. CSS3 The border-radius Property - 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:

Example

Add rounded corners to a div element:

```
div {
border:2px solid;
border-radius:25px;
}
```

2.3. CSS3 The box-shadow Property

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

Example

Add a box-shadow to a div element:

```
div {
box-shadow: 10px 10px 5px #888888;
}
```

2.4. CSS3 The border-image Property

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

Use an image to create a border around a div element:

```
div {
-webkit-border-image:url(border.png) 30 30 round; /* Safari 5 */
-o-border-image:url(border.png) 30 30 round; /* Opera 10.5-12.1 */
border-image:url(border.png) 30 30 round;
}
```

2.5. CSS3 Border Properties

Property	Description	CSS
border-image	A shorthand property for setting all the border-image-* properties	3
border-radius	A shorthand property for setting all the four border-*-radius properties	3
box-shadow	Attaches one or more drop-shadows to the box	3

3. CSS3 Backgrounds

3.1.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.

3.2. Browser Support

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

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

Property					
background-size	9.0	3.0 1.0 -webkit-	4.0 3.6 -moz-	4.1 3.0 -webkit-	10.0 9.5 -o-
background-origin	9.0	1.0	4.0	3.0	10.5

3.3. 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:

```
div
{
background:url(img_flwr.gif);
background-size:80px 60px;
background-repeat:no-repeat;
}
```

Example 2

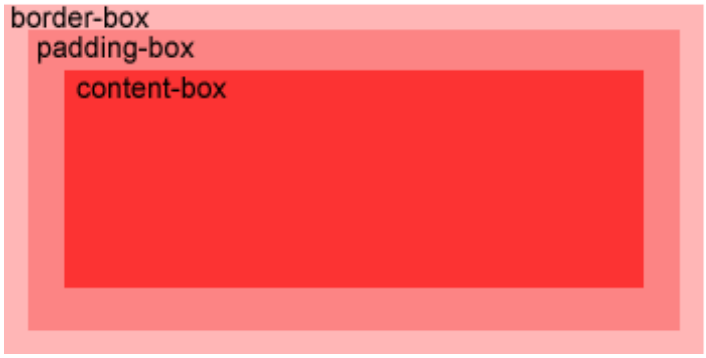
Stretch the background image to completely fill the content area:

```
div
{
background:url(img_flwr.gif);
background-size:100% 100%;
background-repeat:no-repeat;
}
```

3.4. 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:

```
div
{
background:url(img_flwr.gif);
background-repeat:no-repeat;
background-size:100% 100%;
background-origin:content-box;
}
```

3.5. CSS3 Multiple Background Images

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

Example

Set two background images for the body element:

```
body
{
background:url(img_tree.gif),url(img_flwr.gif);
background-size:100% 100%;
background-repeat:no-repeat;
}
```

3.6. CSS3 Background Properties

Property	Description	CSS
background-clip	Specifies the painting area of the background images	3
background-origin	Specifies the positioning area of the background images	3
background-size	Specifies the size of the background images	3

4. CSS3 Gradients

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)**

4.1. Browser Support

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

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

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

4.2. 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.

Example of Linear Gradient:

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 blue:

Example

A linear gradient from top to bottom:

```
#grad
{
background: -webkit-linear-gradient(red, blue); /* For Safari 5.1
to 6.0 */
background: -o-linear-gradient(red, blue); /* For Opera 11.1 to
12.0 */
background: -moz-linear-gradient(red, blue); /* For Firefox 3.6 to
15 */
background: linear-gradient(red, blue); /* Standard syntax */
}
```

Linear Gradient - Left to Right

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

Example

A linear gradient from left to right:

```
#grad
{
background: -webkit-linear-gradient(left, red , blue); /* For Safari 5.1 to 6.0 */
background: -o-linear-gradient(right, red, blue); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(right, red, blue); /* For Firefox 3.6 to 15 */
background: linear-gradient(to right, red , blue); /* 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 blue:

Example

A linear gradient that starts at top left (and goes to bottom right):

```
#grad
{
background: -webkit-linear-gradient(left top, red , blue); /* For Safari 5.1 to 6.0 */
background: -o-linear-gradient(bottom right, red, blue); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(bottom right, red, blue); /* For Firefox 3.6 to 15 */
background: linear-gradient(to bottom right, red , blue); /* Standard syntax */
}
```

4.3. 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, going counter-clockwise. In other words, 0deg creates a bottom to top gradient, while 90deg generates a left to right gradient.

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

Example

A linear gradient with a specified angle:

```
#grad
{
background: -webkit-linear-gradient(180deg, red, blue); /* For Safari 5.1 to 6.0 */
background: -o-linear-gradient(180deg, red, blue); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(180deg, red, blue); /* For Firefox 3.6 to 15 */
background: linear-gradient(180deg, red, blue); /* Standard syntax */
}
```

4.4. Using Multiple Color Stops

The following example shows how to set multiple color stops:

Example

A linear gradient from top to bottom with multiple color stops:

```
#grad
{
background: -webkit-linear-gradient(red, green, blue); /* For Safari 5.1 to 6.0 */
background: -o-linear-gradient(red, green, blue); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(red, green, blue); /* For Firefox 3.6 to 15 */
background: linear-gradient(red, green, blue); /* Standard syntax */
}
```

The following example shows how to create a linear gradient with the color of the rainbow and some text:

Example

```
#grad
{
/* 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);
}
```

4.5. 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

A linear gradient from left to right, with transparency:

```
#grad
{
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*/
}
```

4.6. Repeating a linear-gradient

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

Example

A repeating linear gradient:

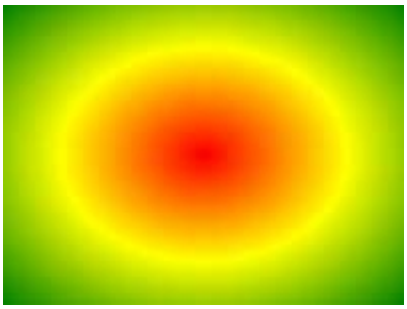
```
#grad
{
/* Safari 5.1 to 6.0 */
background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);
/* Opera 11.1 to 12.0 */
background: -o-repeating-linear-gradient(red, yellow 10%, green 20%);
/* Firefox 3.6 to 15 */
background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%);
/* Standard syntax */
background: repeating-linear-gradient(red, yellow 10%, green 20%);
}
```

4.7. 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. You can also specify the gradient's center, shape (circle or ellipse) as well as its size. By default, center is center, shape is ellipse, and size is farthest-corner.

Example of Radial Gradient:



Syntax

```
background: radial-gradient(center, shape size, start-color, ..., last-color);
```

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

Example

A radial gradient with evenly spaced color stops:

```
#grad
{
background: -webkit-radial-gradient(red, green, blue); /* Safari 5.1 to 6.0 */
background: -o-radial-gradient(red, green, blue); /* For Opera 11.6 to 12.0 */
```

```
background: -moz-radial-gradient(red, green, blue); /* For Firefox 3.6 to 15 */
background: radial-gradient(red, green, blue); /* Standard syntax */
}
```

Radial Gradient - Differently Spaced Color Stops

Example

A radial gradient with differently spaced color stops:

```
#grad
{
background: -webkit-radial-gradient(red 5%, green 15%, blue 60%);
/* Safari 5.1-6.0 */
background: -o-radial-gradient(red 5%, green 15%, blue 60%);
/* For Opera 11.6-12.0 */
background: -moz-radial-gradient(red 5%, green 15%, blue 60%);
/* For Firefox 3.6-15 */
background: radial-gradient(red 5%, green 15%, blue 60%);
/* Standard syntax */
}
```

4.8. Set Shape

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

Example

A radial gradient with the shape of a circle:

```
#grad
{
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 */
}
```

4.9. 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
{
/* Safari 5.1 to 6.0 */
background: -webkit-radial-gradient(60% 55%, closest-side,blue,green,yellow,black);
/* For Opera 11.6 to 12.0 */
background: -o-radial-gradient(60% 55%, closest-side,blue,green,yellow,black);
/* For Firefox 3.6 to 15 */
background: -moz-radial-gradient(60% 55%, closest-side,blue,green,yellow,black);
/* Standard syntax */
background: radial-gradient(60% 55%, closest-side,blue,green,yellow,black);
}

#grad2
{
/* Safari 5.1 to 6.0 */
background: -webkit-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
/* Opera 11.6 to 12.0 */
background: -o-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
/* For Firefox 3.6 to 15 */
background: -moz-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
/* Standard syntax */
background: radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
}
```

4.10. Repeating a radial-gradient

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

Example

A repeating radial gradient:

```
#grad
{
/* 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%);
}
```

5. CSS3 Text Effects

5.1. CSS3 Text Effects

CSS3 contains several new text features.

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

- text-shadow
- word-wrap

5.2. Browser Support

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

Property					
text-shadow	10.0	4.0	3.5	4.0	9.5
word-wrap	5.5	23.0	3.5	6.1	12.1

5.3. 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:

```
h1
{
text-shadow: 5px 5px 5px #FF0000;
}
```

5.4. 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:

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 break and wrap onto the next line:

```
p {word-wrap:break-word;}
```

5.5. CSS3 Text Properties

Property	Description	CSS
hanging-punctuation	Specifies whether a punctuation character may be placed outside the line box	3
punctuation-trim	Specifies whether a punctuation character should be trimmed	3
text-align-last	Describes how the last line of a block or a line right before a forced line break is aligned when text-align is "justify"	3
text-emphasis	Applies emphasis marks, and the foreground color of the emphasis marks, to the element's text	3
text-justify	Specifies the justification method used when text-align is "justify"	3
text-outline	Specifies a text outline	3
text-overflow	Specifies what should happen when text overflows the containing element	3
text-shadow	Adds shadow to text	3
text-wrap	Specifies line breaking rules for text	3
word-break	Specifies line breaking rules for non-CJK scripts	3
word-wrap	Allows long, unbreakable words to be broken and wrap to the next line	3

6. CSS3 Web Fonts

6.1. 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.

6.2. Browser Support

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

Property					
@font-face	9.0	4.0	3.5	3.2	10.0

6.3. 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.
- **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.

6.4. 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 fonts	9.0*	4.0	3.5	3.1	10.0
WOFF fonts	9.0	5.0	3.6	5.1	11.1
SVG fonts	Not supported	4.0	Not supported	3.2	9.0
EOT fonts	6.0	Not supported	Not supported	Not supported	Not supported

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

6.5. 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.

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-family property:

Example

```
<style>
@font-face
{
font-family: myFirstFont;
src: url(sansation_light.woff);
}

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

6.6. 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;
}
```

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.

6.7. 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"

7. CSS3 2D Transforms

7.1. CSS3 Transforms

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

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

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

7.2. Browser Support for 2D Transforms

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

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

Property					
transform	10.0 9.0 -ms-	12.0 -webkit-	16.0 3.5 -moz-	3.1 -webkit-	15.0 -webkit- 12.1 10.5 -o-
transform-origin (two-value syntax)	10.0 9.0 -ms-	12.0 -webkit-	16.0 3.5 -moz-	3.1 -webkit-	15.0 -webkit- 12.1 10.5 -o-

7.3. CSS3 2D Transforms

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

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

You will learn about 3D transforms in the next chapter.

Example

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

7.4. 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

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

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

7.5. 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

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

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

7.6. The scale() Method

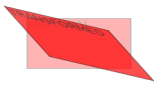
With the scale() method, the element increases or decreases the size, depending on the parameters given for the width (X-axis) and the height (Y-axis):

Example

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

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

7.7. 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

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

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

7.8. 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

How to rotate a div element 30 degrees, using the matrix method:

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

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

7.9. CSS3 Transform Properties

The following table lists all the transform properties:

Property	Description
transform	Applies a 2D or 3D transformation to an element
transform-origin	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

8. CSS3 3D Transforms

8.1. 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()

8.2. Browser Support for 3D Transforms

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

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

Property					
transform	10.0	12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	15.0 -webkit-
transform-origin (three-value syntax)	10.0	12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	15.0 -webkit-
transform-style	Not supported	12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	15.0 -webkit-
perspective	10.0	12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	15.0 -webkit-
perspective-origin	10.0	12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	15.0 -webkit-
backface-visibility	10.0	12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	15.0 -webkit-

8.3. The rotateX() Method

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

Example

```
div
{
-webkit-transform: rotateX(120deg); /* Chrome, Safari, Opera */
```

```
    transform: rotateX(120deg);
    }
```

8.4. The rotateY() Method

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

Example

```
    div
    {
    -webkit-transform: rotateY(130deg); /* Chrome, Safari, Opera */
    transform: rotateY(130deg);
    }
```

8.5. CSS3 Transform Properties

The following table lists all the transform properties:

Property	Description
transform	Applies a 2D or 3D transformation to an element
transform-origin	Allows you to change the position on transformed elements
transform-style	Specifies how nested elements are rendered in 3D space
perspective	Specifies the perspective on how 3D elements are viewed
perspective-origin	Specifies the bottom position of 3D elements
backface-visibility	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

9. CSS3 Transitions

9.1. CSS3 Transitions

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

9.2. Browser Support

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

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

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

9.3. What Are CSS3 Transitions?

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

To do this, you must specify two things:

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

Example

Add a transition effect on the width property, with a duration of 2 seconds:

```
div
{
-webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
transition: width 2s;
}
```

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

The transition 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:

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

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

9.4. Multiple Changes

To add transition effects for more than one CSS property, separate the properties with a comma:

Example

Add transition effects on width, height, and transformation:

```
div
{
-webkit-transition: width 2s, height 2s,-webkit-transform 2s;
/* For Safari 3.1 to 6.0 */
transition: width 2s, height 2s, transform 2s;
}
```

9.5. More Transition Examples

The example below uses all the four transition properties:

Example

```
div
{
/* For Safari 3.1 to 6.0 */
-webkit-transition-property:width;
-webkit-transition-duration:1s;
-webkit-transition-timing-function:linear;
-webkit-transition-delay:2s;
/* Standard syntax */
transition-property: width;
transition-duration: 1s;
transition-timing-function: linear;
```

```
    transition-delay: 2s;
    }
```

The same transition effects as the example above. However, here we are using the shorthand transition property:

Example

```
    div
    {
    -webkit-transition:width 1s linear 2s; /* For Safari 3.1 to 6.0 */
    transition: width 1s linear 2s;
    }
```

9.6. CSS3 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-delay	Specifies when the transition effect will start	3
transition-duration	Specifies how many seconds or milliseconds a transition effect takes to complete	3
transition-property	Specifies the name of the CSS property the transition effect is for	3
transition-timing-function	Specifies the speed curve of the transition effect	3

10. CSS3 Animations

10.1. CSS3 Animations

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

10.2. Browser Support

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

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

Property					
@keyframes	10.0	4.0 -webkit-	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-
animation	10.0	4.0 -webkit-	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

10.3. 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.

10.4. CSS3 Animation

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

Bind the animation to a selector (element) by specifying at least these two properties:

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

Example

Bind the "myfirst" animation to the div element. Animation duration: 5 seconds:

```
div
{
-webkit-animation: myfirst 5s; /* Chrome, Safari,
Opera */
animation: myfirst 5s;
}

/* Chrome, Safari, Opera */
@-webkit-keyframes myfirst
{
from {background: red;}
to {background: yellow;}
}

/* Standard syntax */
@keyframes myfirst
{
from {background: red;}
to {background: yellow;}
}
```

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

10.5. What Are CSS3 Animations?

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

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

You can specify when the change will happen in percent, or you can use the keywords "from" and "to" (which represents 0% and 100%).

0% represents the start of the animation, 100% is when the animation is complete.

Example

Change the background color when the animation is 25%, and 50%, and again when the animation is 100% complete:

```
/* Chrome, Safari, Opera */
@-webkit-keyframes myfirst
{
0%   {background: red;}
25%  {background: yellow;}
50%  {background: blue;}
100% {background: green;}
}

/* Standard syntax */
@keyframes myfirst
{
0%   {background: red;}
```

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

Example

Change the background color and the position when the animation is 25%, 50%, 75%, and again when the animation is 100% complete:

```
    /* Chrome, Safari, Opera */
    @-webkit-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;}
    }

    /* Standard syntax */
    @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;}
    }
```

10.6. More Animation Examples

The example below uses seven of the animation properties:

Example

```
    div
    {
    /* Chrome, Safari, Opera */
    -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;

    /* Standard syntax */
    animation-name: myfirst;
    animation-duration: 5s;
    animation-timing-function: linear;
    animation-delay: 2s;
    animation-iteration-count: infinite;
```

```
    animation-direction: alternate;
    animation-play-state: running;
    }
```

The same animation effect as the example above (except the animation-play-state property). However, here we are using the shorthand animation property:

Example

```
    div
    {
    -webkit-animation: myfirst 5s linear 2s infinite alternate;
    /* Chrome, Safari, Opera */
    animation: myfirst 5s linear 2s infinite alternate;
    /* Standard syntax */
    }
```

10.7. 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 setting all the animation properties, except the animation-play-state and the animation-fill-mode property	3
animation-delay	Specifies when the animation will start	3
animation-direction	Specifies whether or not the animation should play in reverse on alternate cycles	3
animation-duration	Specifies how many seconds or milliseconds an animation takes to complete one cycle	3
animation-fill-mode	Specifies what styles will apply for the element when the animation is not playing (when it is finished, or when it has a "delay")	3
animation-iteration-count	Specifies the number of times an animation should be played	3
animation-name	Specifies the name of the @keyframes animation	3
animation-play-state	Specifies whether the animation is running or paused	3
animation-timing-function	Specifies the speed curve of the animation	3

11. CSS3 Multiple Columns

11.1. 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

11.2. Browser Support

The numbers in the table specifies 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					
column-count	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1
column-gap	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1
column-rule	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

11.3. 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:

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

11.4. CSS3 Specify the Gap Between Columns

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

Example

Specify a 40 pixels gap between the columns:

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

11.5. CSS3 Column Rules

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

Example

Specify the width, style and color of the rule between columns:

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

11.6. CSS3 Multiple Columns Properties

The following table lists all the 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

12. CSS3 User Interface

12.1. 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

12.2. Browser Support

The numbers in the table specifies 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					
resize	Not supported	4.0	5.0 4.0 -moz-	4.0	15.0
box-sizing	Partial from 8.0	10.0 4.0 -webkit-	29.0 2.0 -moz-	5.1 3.1 -webkit-	9.5
outline-offset	Not supported	4.0	5.0 4.0 -moz-	4.0	9.5

12.3. CSS3 Resizing

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

Example

Specify that a div element should be resizable by the user:

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

12.4. 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:

```
div {
-moz-box-sizing:border-box; /* Firefox */
box-sizing:border-box;
width:50%;
float:left;
}
```

12.5. 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:

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

12.6. CSS3 User-interface Properties

The following table lists all the user-interface properties:

Property	Description	CSS
appearance	Allows you to make an element look like a standard user interface element	3
box-sizing	Allows you to define certain elements to fit an area in a certain way	3
icon	Provides the author the ability to style an element with an iconic equivalent	3
nav-down	Specifies where to navigate when using the arrow-down navigation key	3
nav-index	Specifies the tabbing order for an element	3
nav-left	Specifies where to navigate when using the arrow-left navigation key	3
nav-right	Specifies where to navigate when using the arrow-right navigation key	3
nav-up	Specifies where to navigate when using the arrow-up navigation key	3
outline-offset	Offsets an outline, and draws it beyond the border edge	3

resize	Specifies whether or not an element is resizable by the user	3

Property