



University of Babylon
College of Information Technology
Department of Information Networks

Web Page Design

Introduction to CSS3

Part I

Lecturer

Haider M. Habeeb

Second Year, Second Course

2012 - 2013

Introduction to CSS3

CSS3 is the latest standard for CSS. CSS is used to control the style and layout of Web pages. In this lecture we are going to learn some of new features in CSS3.

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 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:

Property	Browser Support				
border-radius					
box-shadow					
border-image					

Till the date of this lecture:

- 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, 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>
div
{
border:2px solid #a1a1a1;
padding:10px 40px;
background:#dddddd;
width:300px;
```

```
border-radius:25px;
-moz-border-radius:25px; /* Old Firefox */
}
</style>
</head>
<body>

<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:



Example

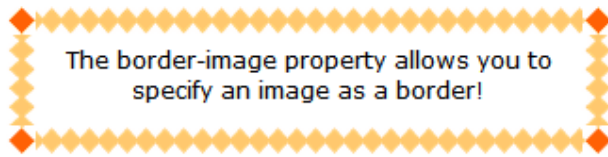
```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
width:300px;
height:100px;
background-color:yellow;
-moz-box-shadow: 10px 10px 5px #888888; /* Old Firefox */
box-shadow: 10px 10px 5px #888888;
}
</style>
</head>
<body>

<div></div>

</body>
</html>
```

CSS3 Border Image

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



The original image used to create the border above is:



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

#round
{
-moz-border-image:url(border.png) 30 30 round; /* Old 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; /* Old 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>


</body>
</html>
```

The result of the above:

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

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

Here, the image is tiled (repeated) to fill the area.

Here, the image is stretched to fill the area.

Here is the image used:



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
<pre><!DOCTYPE html> <html> <head> <style> * {margin: 0px; padding: 0px;} body { background:url(img_flwr.gif); background-size:100px 80px; -moz-background-size:80px 60px; /* Old Firefox */ background-repeat:no-repeat; } </style> </head> <body> <p> 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. </p> <p>Original image: </p> </body> </html></pre>

The result of the above:

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.



Original image:

Stretch the background image to completely fill the content area:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
background:url(img_flwr.gif);
background-size:100% 100%;
-moz-background-size:100% 100%; /* Old Firefox */
background-repeat:no-repeat;
}
</style>
</head>
<body>

<div>
CSS3 The background-size Property<br />
```

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.


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

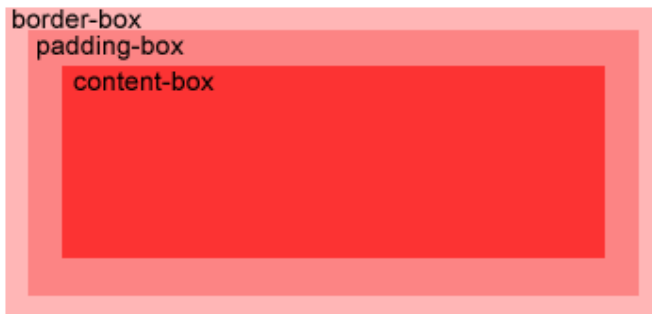
The result of the above:

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.


CSS3 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.



Position the background image within the content-box:

The image is: 

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;
}
#div3
{
background-origin:padding-box;
}

</style>
</head>
<body>
```

```
<p>background-origin:border-box:</p>
<div id="div1">
CSS3 The background-origin Property<br />
```

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.
</div>

```
<p>background-origin:content-box:</p>
<div id="div2">
CSS3 The background-origin Property<br />
```

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.</div>

```
<p>background-origin:padding-box:</p>
<div id="div3">
CSS3 The background-origin Property<br />
```

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.</div>

</body>


</html>

The result of the above:


background-origin: border-box:

 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.

background-origin: content-box:

 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.

background-origin: padding-box:

 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.

CSS3 Multiple Background Images

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



Set two background images for the body element:

Example

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

</body>
</html>
```

The result of the above:



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

Property	Browser Support				
text-shadow					
word-wrap					

Till the date of this lecture:

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.

Text shadow effect!

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>

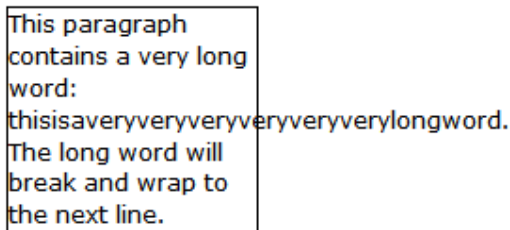
<h1>Text-shadow effect!</h1>

<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:

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

The CSS code is as follows:

Example
<pre><!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: thisisaveryveryveryeryveryverylongword. The long word will break and wrap to the next line.</p> </body> </html></pre>

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

Property	Browser Support				
@font-face					

Till the date of this lecture:

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

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



```
</head>
<body>

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

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

</body>
</html>
```

The result of the above:

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>
@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>

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

</body>
</html>
```

The result of the above:

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 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

Property	Browser Support				
transform					

Till now:

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()

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

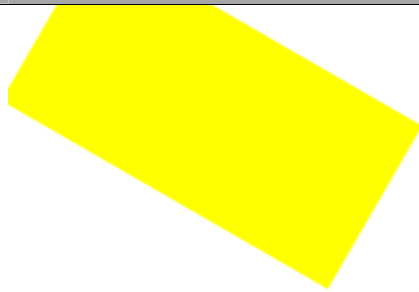
Example
<pre><!DOCTYPE html> <html> <head> <style> div { width:200px; height:100px; background-color:yellow; /* Rotate div */ transform:rotate(30deg); -ms-transform:rotate(30deg); /* IE 9 */ -moz-transform:rotate(30deg); /* Firefox */ -webkit-transform:rotate(30deg); /* Safari and Chrome */</pre>

```
-o-transform:rotate(30deg); /* Opera */
}
</style>
</head>
<body>

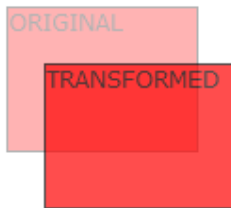
<div>Hello</div>

</body>
</html>
```

The result of the above:



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>
div
{
width:100px;
height:75px;
```

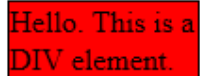
```
background-color:red;
border:1px solid black;
}
div#div2
{
transform:translate(50px,100px);
-ms-transform:translate(50px,100px); /* IE 9 */
-moz-transform:translate(50px,100px); /* Firefox */
-webkit-transform:translate(50px,100px); /* Safari and Chrome */
-o-transform:translate(50px,100px); /* Opera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

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

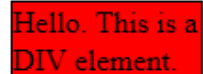
<div id="div2">Hello. This is a DIV element.</div>

</body>
</html>
```

The result of the above:



Hello. This is a
DIV element.



Hello. This is a
DIV element.

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>
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>
```

The result of above:



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
<pre><!DOCTYPE html> <html> <head> <style> div { width:100px; height:75px; background-color:red; border:1px solid black; } div#div2 { margin:100px;</pre>

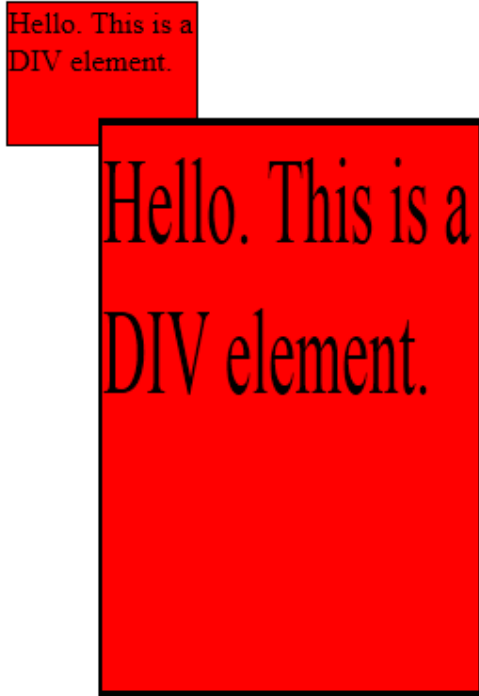
```
transform:scale(2,4);
-ms-transform:scale(2,4); /* IE 9 */
-moz-transform:scale(2,4); /* Firefox */
-webkit-transform:scale(2,4); /* Safari and Chrome */
-o-transform:scale(2,4); /* Opera */
}
</style>
</head>
<body>

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

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

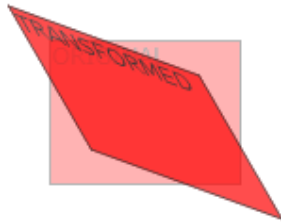
</body>
</html>
```

The result of the above:



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>
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>
```

The result of the above:



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





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

Property	Browser Support				
<code>transform</code>					

Till the date of this lecture:

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>
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>

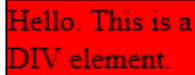
<div>Hello. This is a DIV element.</div>

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

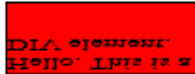
</body>
</html>
```

The result of the above:

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

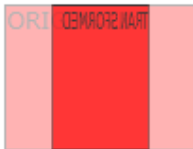


Hello. This is a
DIV element.



DIV element
Hello. This is a

The rotateY() Method



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

Example

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

The result of the above:

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

