CSS3

Target CSS3

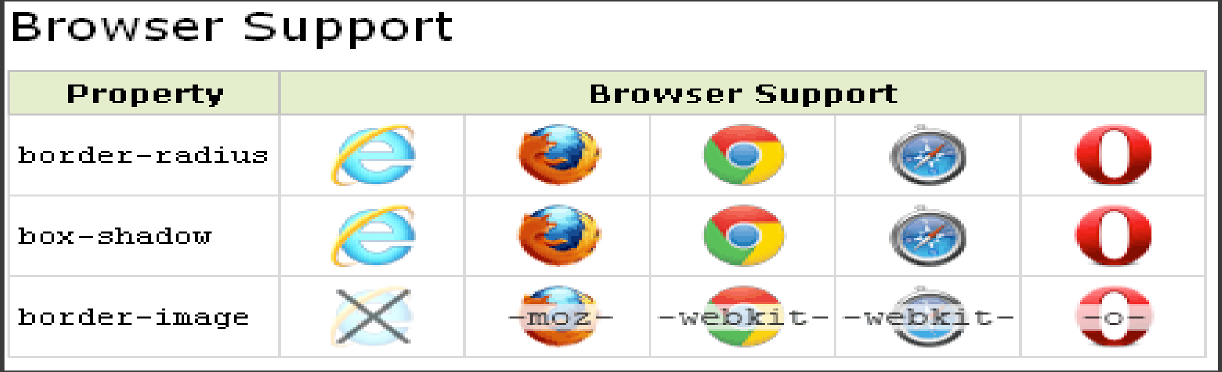
- Meningkatkan “kemampuan” dari elemen dan konten dari HTML5.
- Menggantikan beberapa teknik kuno, seperti menggunakan gambar untuk menampilkan sudut lengkung ataupun efek bayangan.
- Menggantikan beberapa aksi ataupun tingkah laku yang sebelumnya hanya bisa dilakukan dengan menggunakan javascript.

1. Introduction to CSS3

- Successor of CSS2
- It comes with new modules
- Below is a non exhaustive list of features that tend to be labelled as “css3" in the medias:
  - 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

2. CSS3 Borders

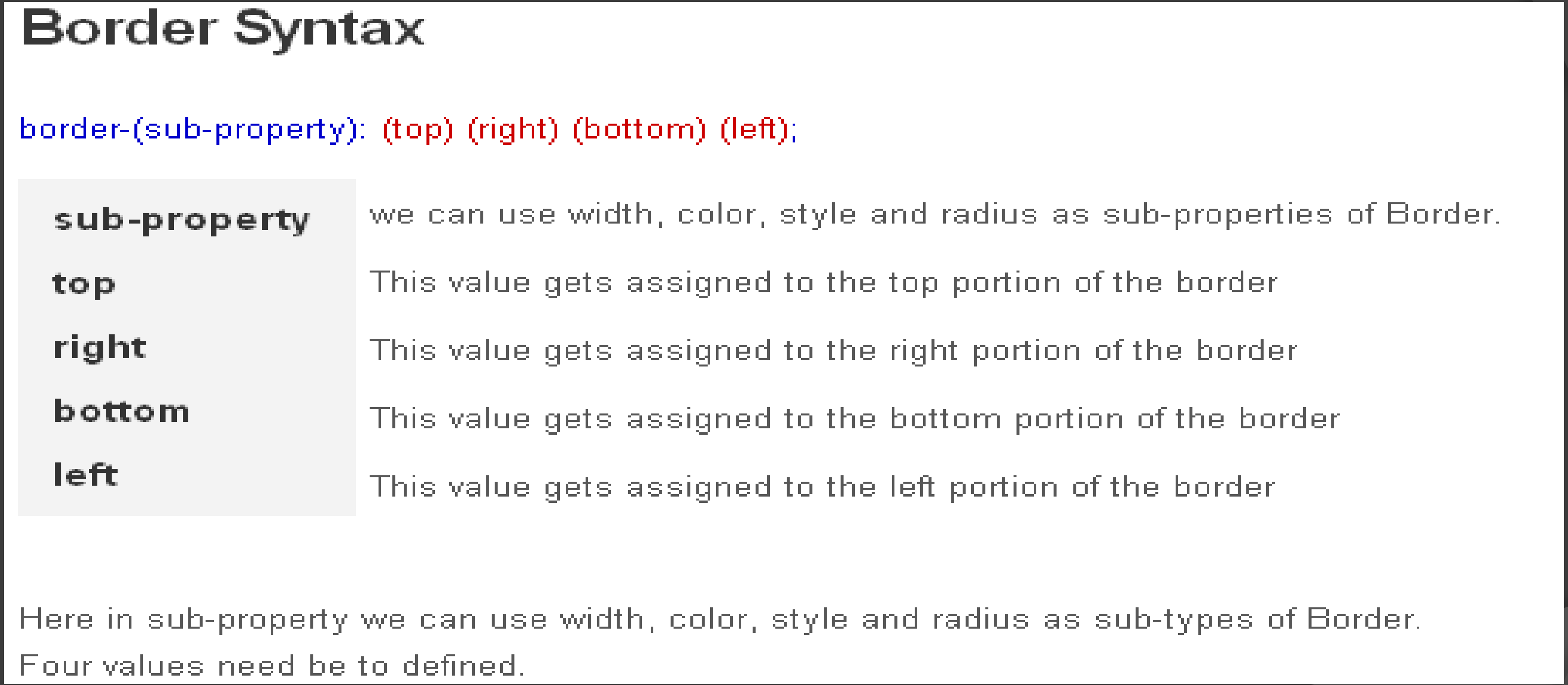
With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border - without using a design program, like Photoshop



Browser Support

Property	Browser Support				
border-radius	✓	✓	✓	✓	✓
box-shadow	✓	✓	✓	✓	✓
border-image	✗	-moz-	-webkit-	-webkit-	-o-

2. CSS3 Borders



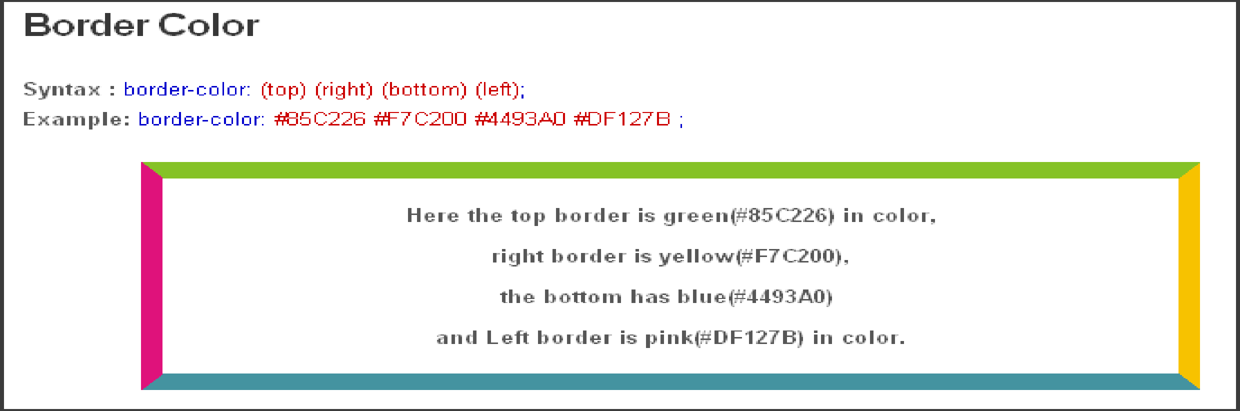
Border Syntax

border-(sub-property): (top) (right) (bottom) (left);

sub-property	we can use width, color, style and radius as sub-properties of Border.
top	This value gets assigned to the top portion of the border
right	This value gets assigned to the right portion of the border
bottom	This value gets assigned to the bottom portion of the border
left	This value gets assigned to the left portion of the border

Here in sub-property we can use width, color, style and radius as sub-types of Border.
Four values need be to defined.

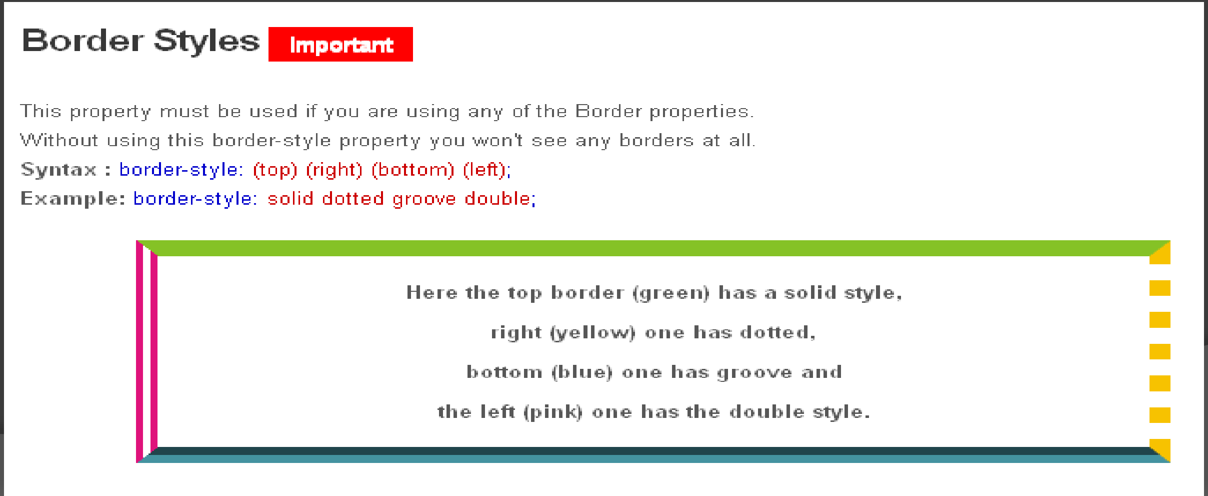
2. CSS3 Borders



Border Color

Syntax : border-color: (top) (right) (bottom) (left);
Example: border-color: #85C226 #F7C200 #4493A0 #DF127B ;

Here the top border is green(#85C226) in color,
right border is yellow(#F7C200),
the bottom has blue(#4493A0)
and Left border is pink(#DF127B) in color.

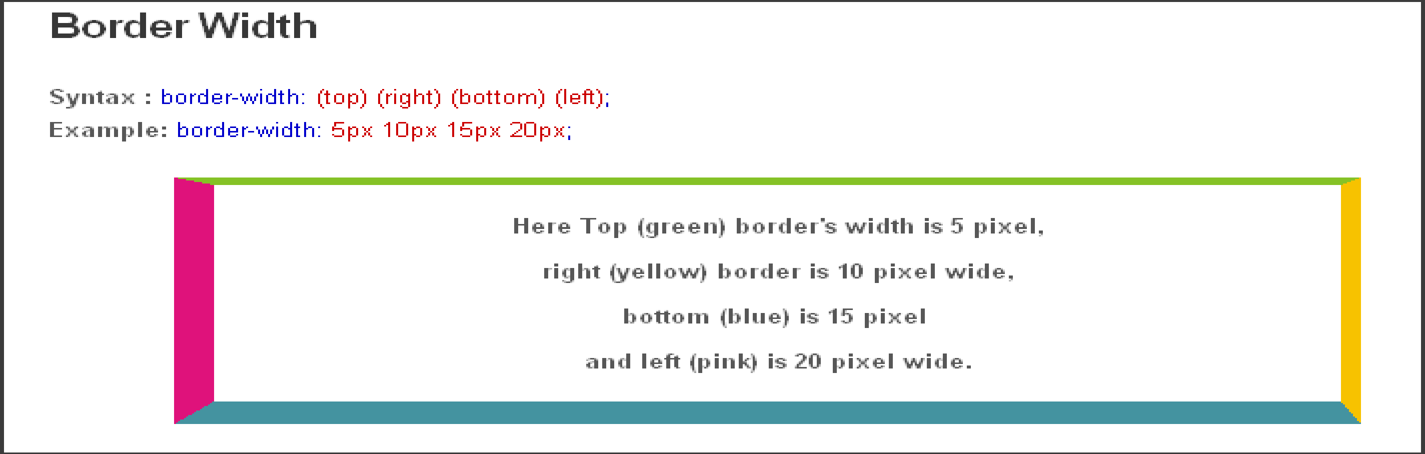


Border Styles Important

This property must be used if you are using any of the Border properties.
Without using this border-style property you won't see any borders at all.
Syntax : border-style: (top) (right) (bottom) (left);
Example: border-style: solid dotted groove double;

Here the top border (green) has a solid style,
right (yellow) one has dotted,
bottom (blue) one has groove and
the left (pink) one has the double style.

2. CSS3 Borders



Border Width

Syntax : border-width: (top) (right) (bottom) (left);

Example: border-width: 5px 10px 15px 20px;

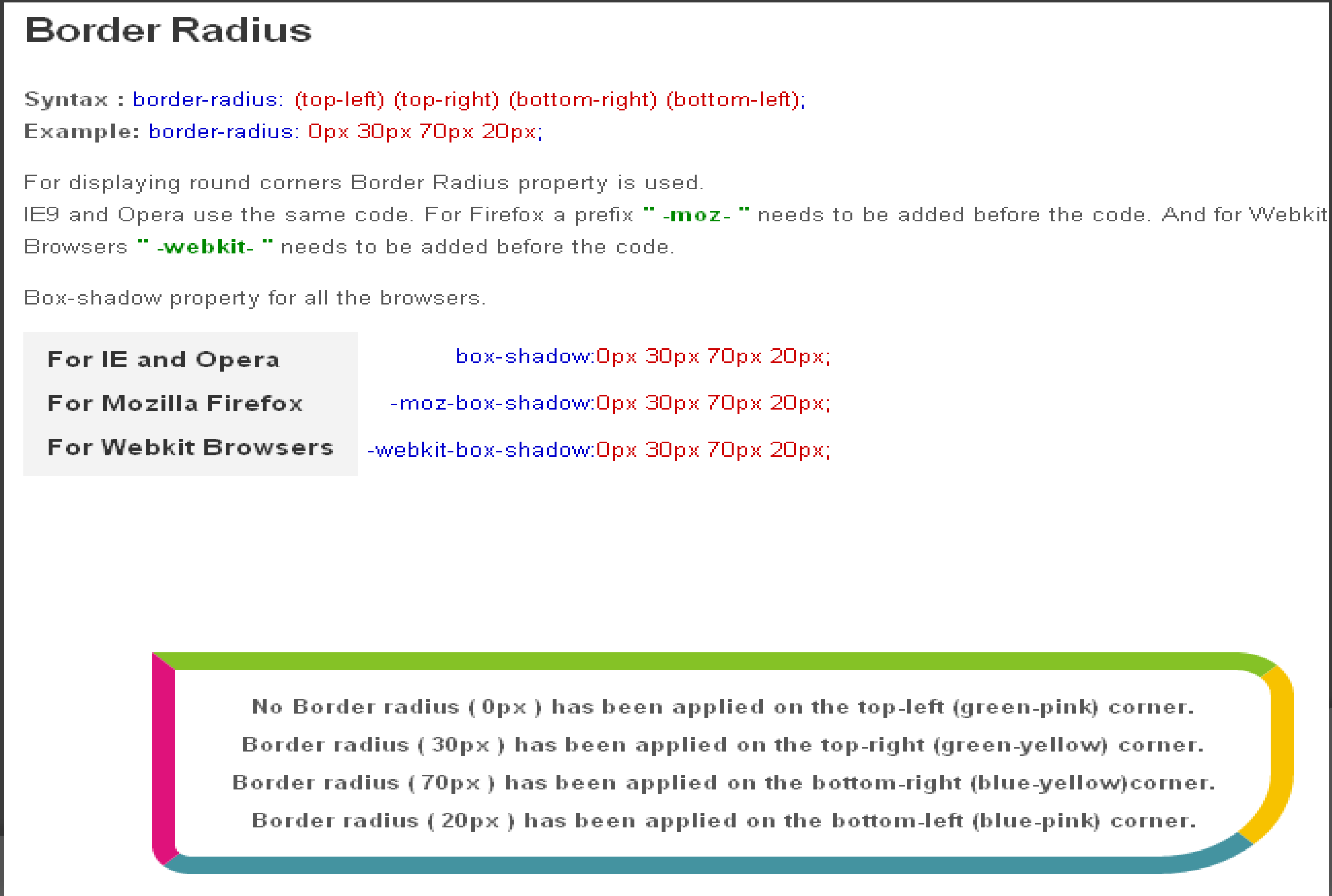
Here Top (green) border's width is 5 pixel,

right (yellow) border is 10 pixel wide,

bottom (blue) is 15 pixel

and left (pink) is 20 pixel wide.

2. CSS3 Borders



Border Radius

Syntax : border-radius: (top-left) (top-right) (bottom-right) (bottom-left);
Example: border-radius: 0px 30px 70px 20px;

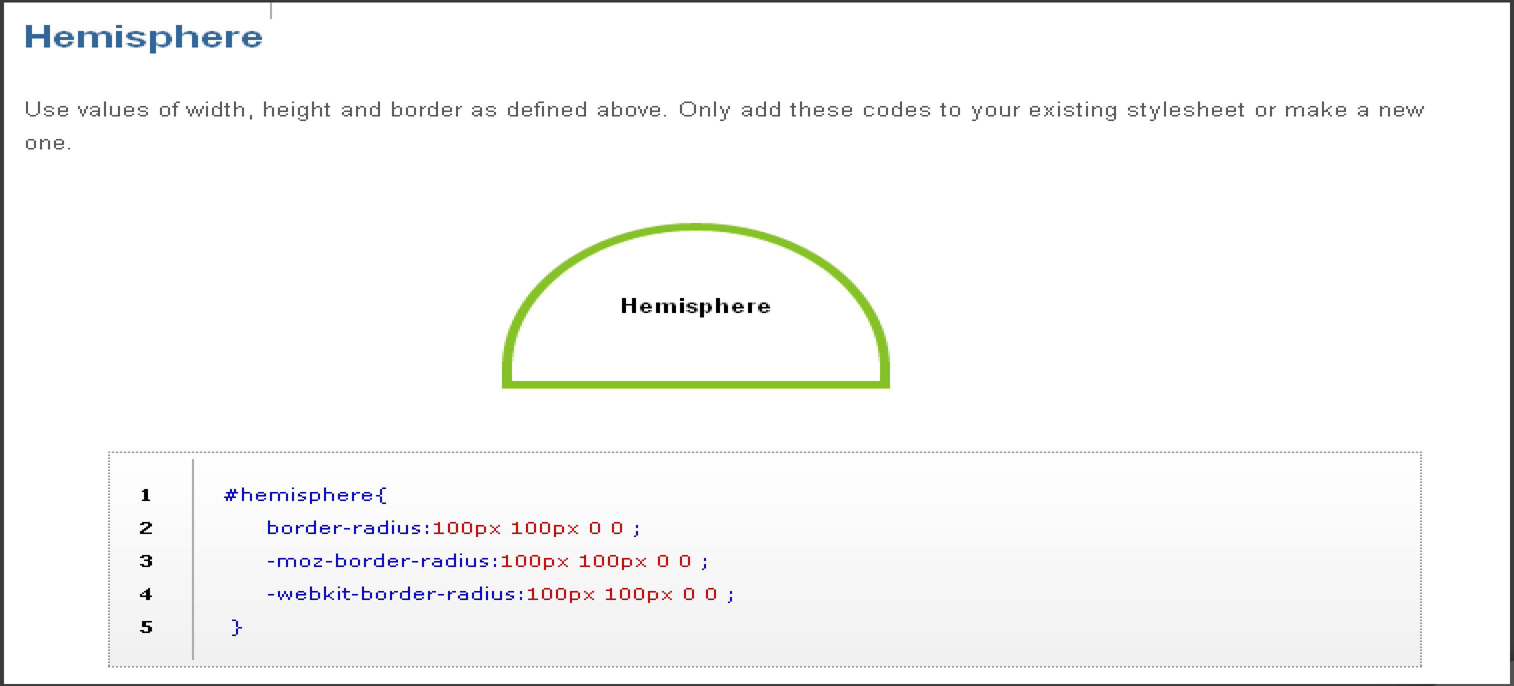
For displaying round corners Border Radius property is used.
IE9 and Opera use the same code. For Firefox a prefix **" -moz- "** needs to be added before the code. And for Webkit
Browsers **" -webkit- "** needs to be added before the code.

Box-shadow property for all the browsers.

For IE and Opera	box-shadow:0px 30px 70px 20px;
For Mozilla Firefox	-moz-box-shadow:0px 30px 70px 20px;
For Webkit Browsers	-webkit-box-shadow:0px 30px 70px 20px;

No Border radius (0px) has been applied on the top-left (green-pink) corner.
Border radius (30px) has been applied on the top-right (green-yellow) corner.
Border radius (70px) has been applied on the bottom-right (blue-yellow)corner.
Border radius (20px) has been applied on the bottom-left (blue-pink) corner.

2. CSS3 Borders

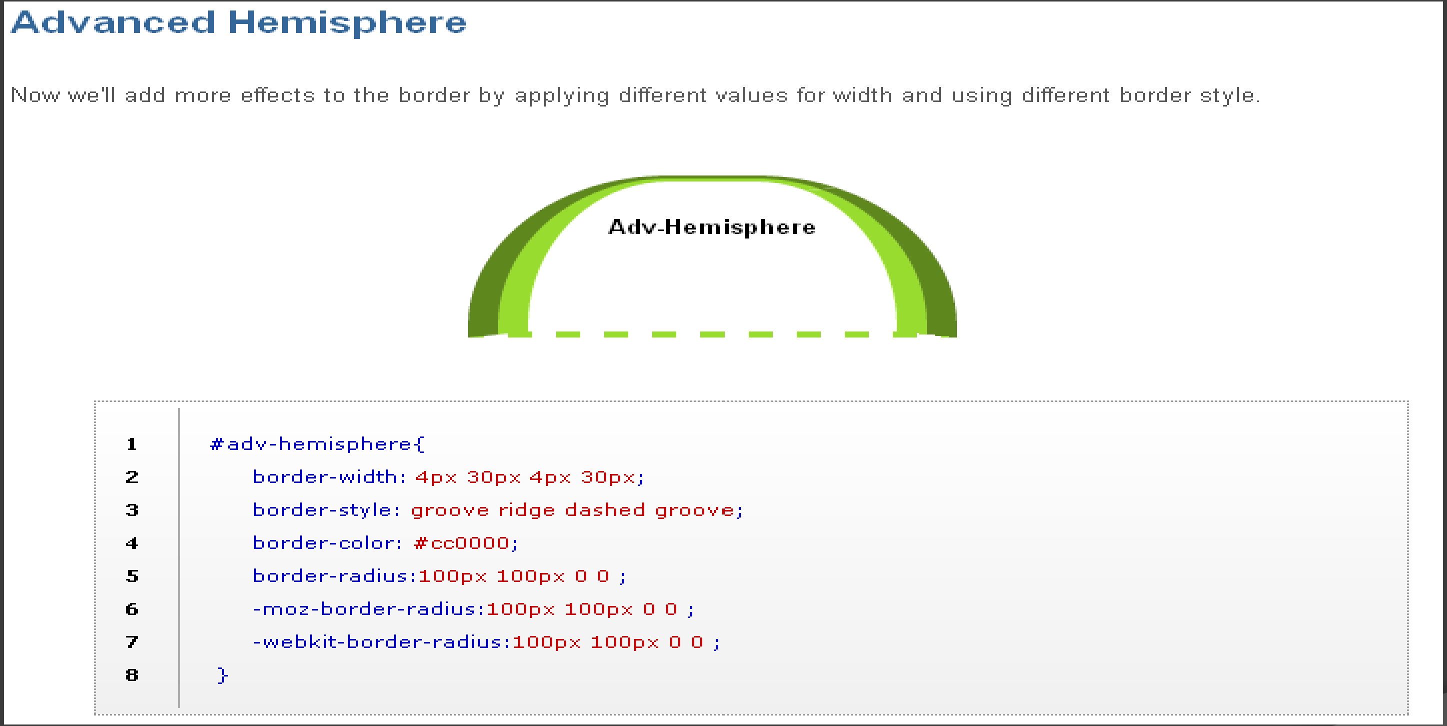


Hemisphere

Use values of width, height and border as defined above. Only add these codes to your existing stylesheet or make a new one.

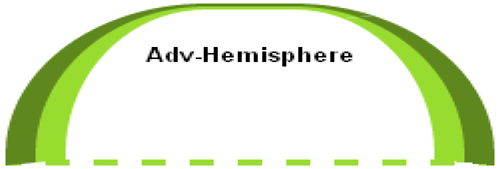
```
#hemisphere{
    border-radius:100px 100px 0 0 ;
    -moz-border-radius:100px 100px 0 0 ;
    -webkit-border-radius:100px 100px 0 0 ;
 }
```

2. CSS3 Borders



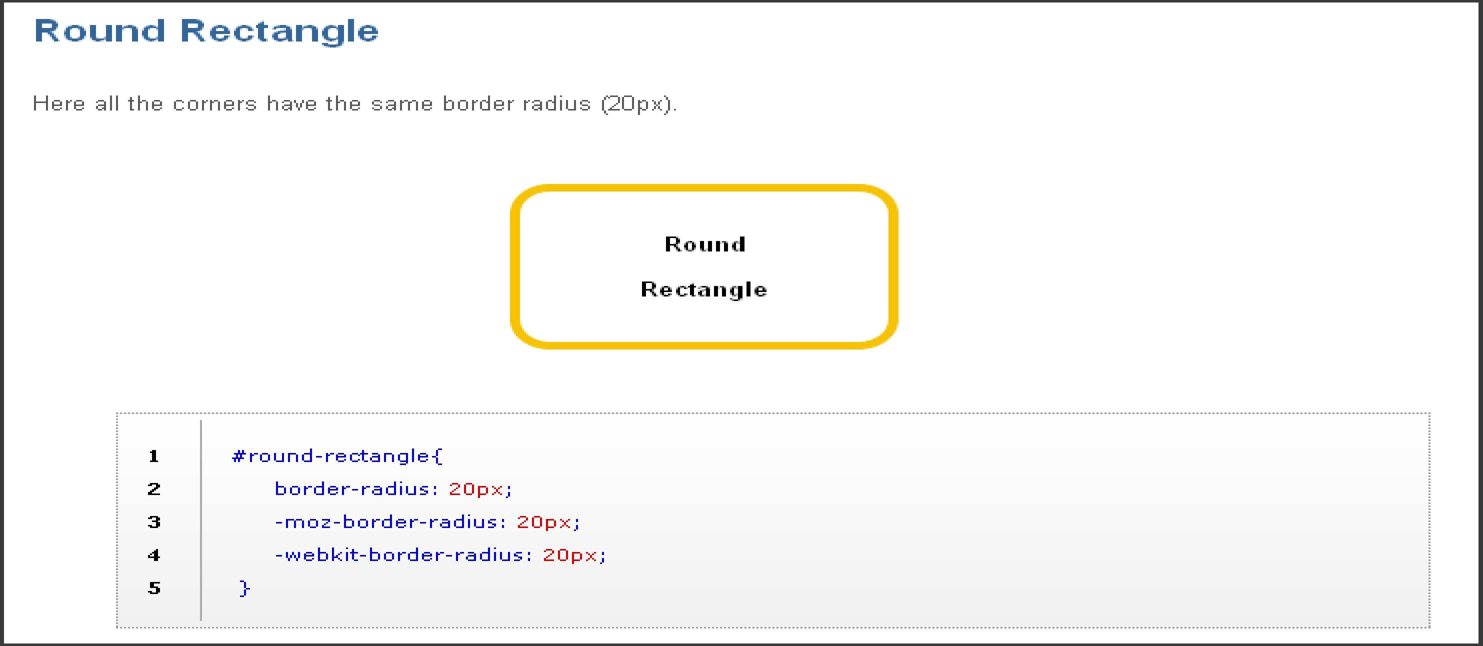
Advanced Hemisphere

Now we'll add more effects to the border by applying different values for width and using different border style.



```
#adv-hemisphere{
    border-width: 4px 30px 4px 30px;
    border-style: groove ridge dashed groove;
    border-color: #cc0000;
    border-radius:100px 100px 0 0 ;
    -moz-border-radius:100px 100px 0 0 ;
    -webkit-border-radius:100px 100px 0 0 ;
 }
```

2. CSS3 Borders

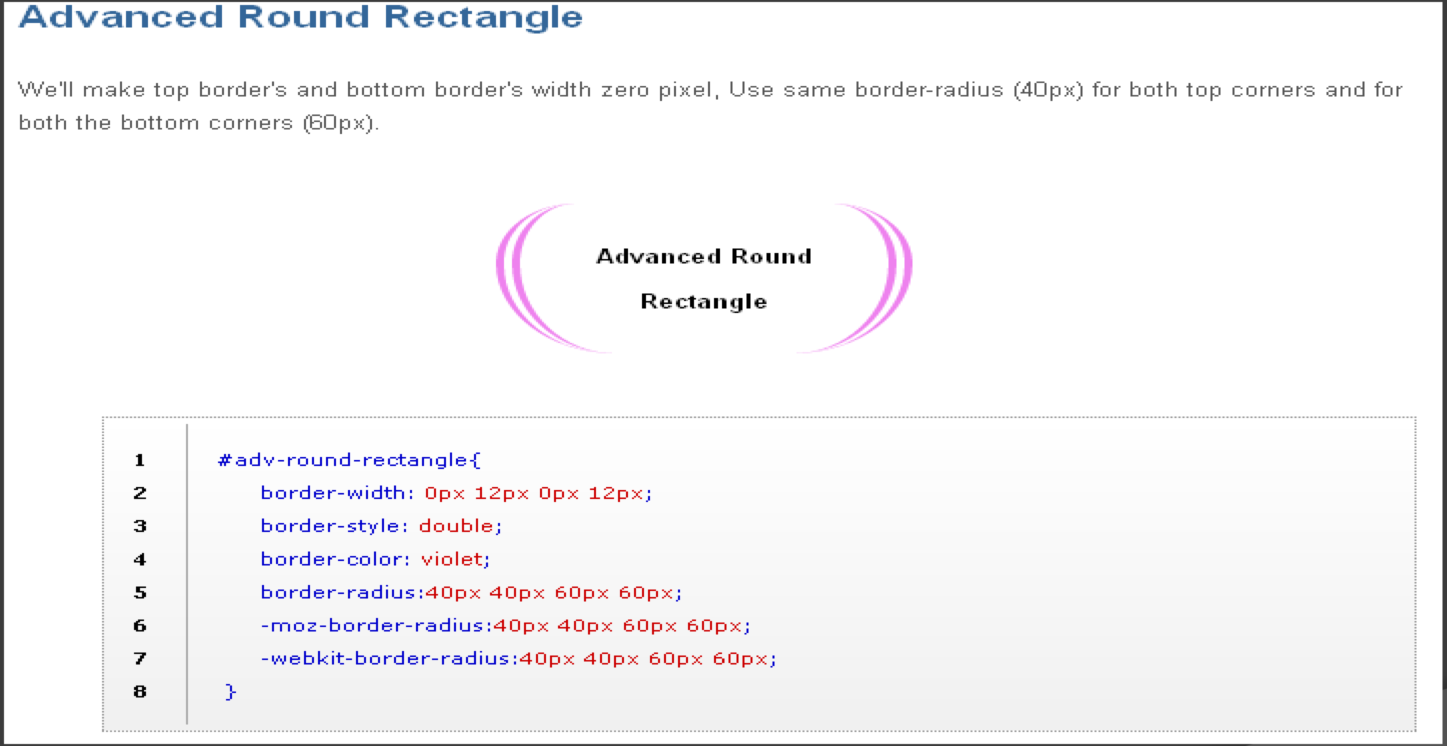


Round Rectangle

Here all the corners have the same border radius (20px).

```css
#round-rectangle{
    border-radius: 20px;
    -moz-border-radius: 20px;
    -webkit-border-radius: 20px;
 }
```

2. CSS3 Borders



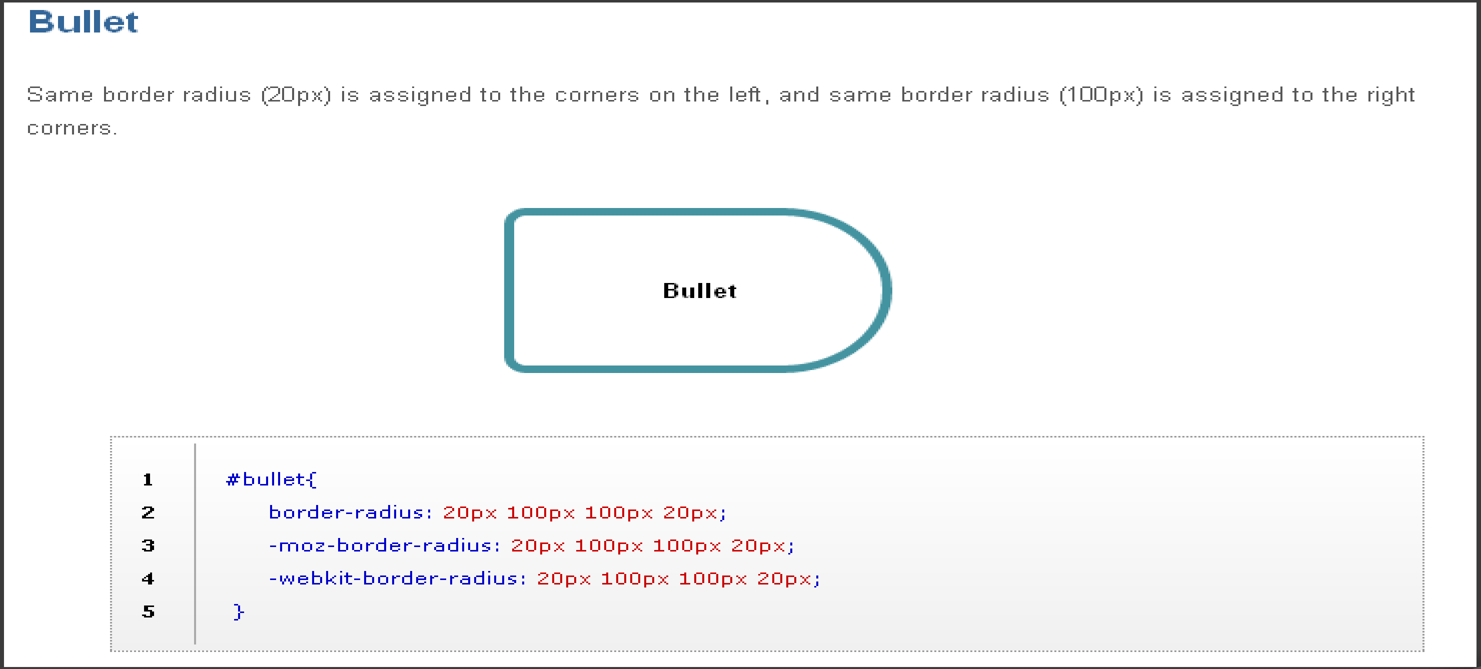
Advanced Round Rectangle

We'll make top border's and bottom border's width zero pixel, Use same border-radius (40px) for both top corners and for both the bottom corners (60px).

Advanced Round Rectangle

```
#adv-round-rectangle{
    border-width: 0px 12px 0px 12px;
    border-style: double;
    border-color: violet;
    border-radius:40px 40px 60px 60px;
    -moz-border-radius:40px 40px 60px 60px;
    -webkit-border-radius:40px 40px 60px 60px;
 }
```

2. CSS3 Borders

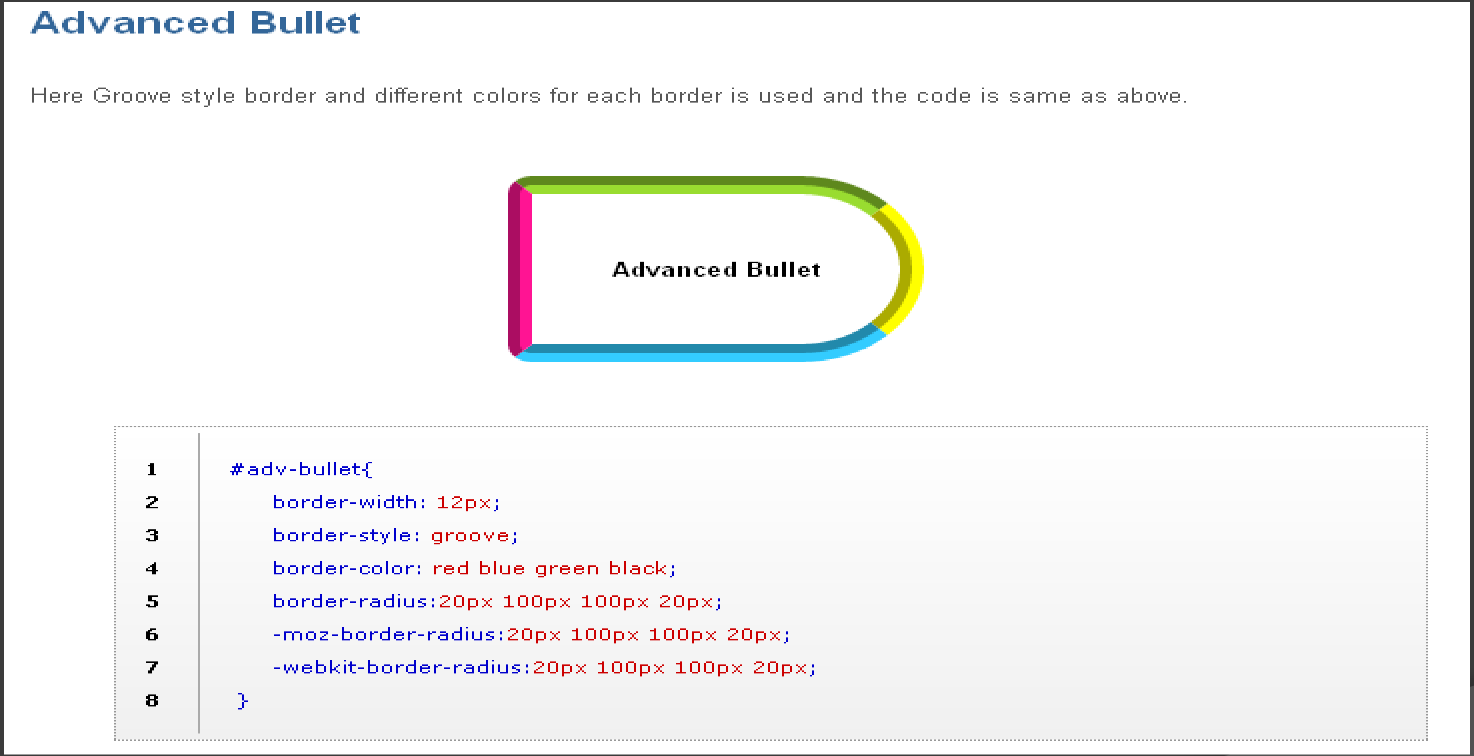


Bullet

Same border radius (20px) is assigned to the corners on the left, and same border radius (100px) is assigned to the right corners.

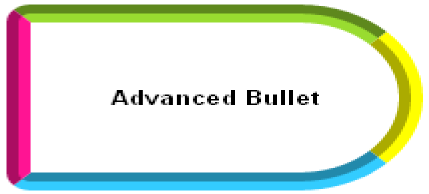
```
#bullet{
    border-radius: 20px 100px 100px 20px;
    -moz-border-radius: 20px 100px 100px 20px;
    -webkit-border-radius: 20px 100px 100px 20px;
 }
```

2. CSS3 Borders



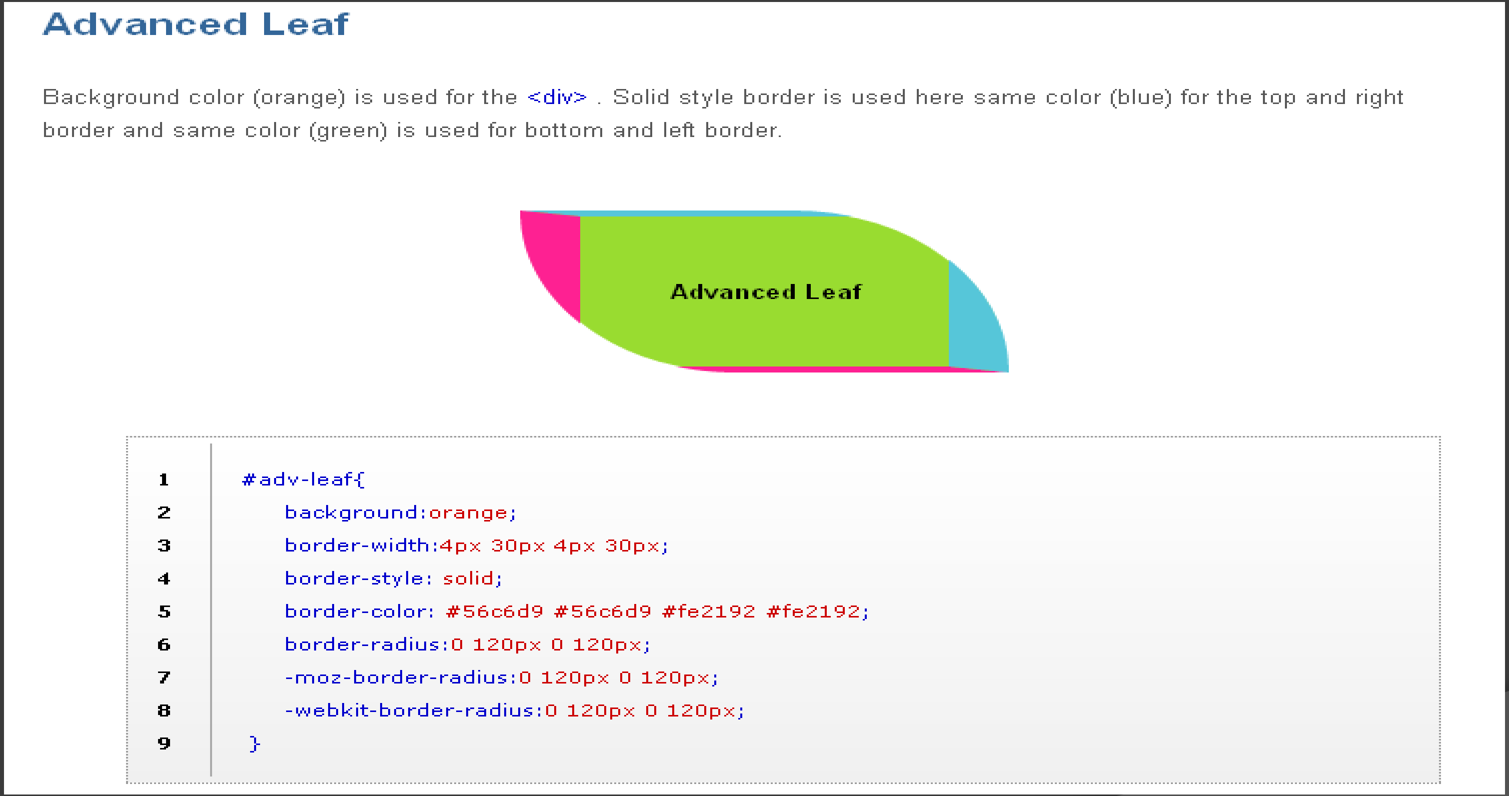
Advanced Bullet

Here Groove style border and different colors for each border is used and the code is same as above.



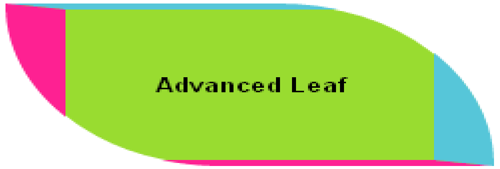
```
#adv-bullet{
    border-width: 12px;
    border-style: groove;
    border-color: red blue green black;
    border-radius:20px 100px 100px 20px;
    -moz-border-radius:20px 100px 100px 20px;
    -webkit-border-radius:20px 100px 100px 20px;
 }
```

2. CSS3 Borders



Advanced Leaf

Background color (orange) is used for the <div> . Solid style border is used here same color (blue) for the top and right border and same color (green) is used for bottom and left border.



```
#adv-leaf{
    background:orange;
    border-width:4px 30px 4px 30px;
    border-style: solid;
    border-color: #56c6d9 #56c6d9 #fe2192 #fe2192;
    border-radius:0 120px 0 120px;
    -moz-border-radius:0 120px 0 120px;
    -webkit-border-radius:0 120px 0 120px;
 }
```

3. CSS3 Background

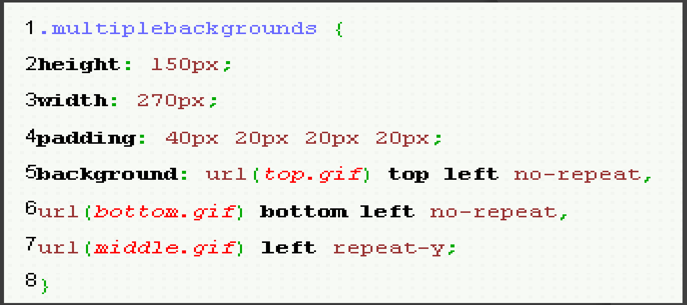
we will be taking a look at the new background properties. These include background size, using more than one background for an element, and background origin (which effects the position of a background).

The new features allow greater control of the background element and will provide designers with a whole array of new possibilities.

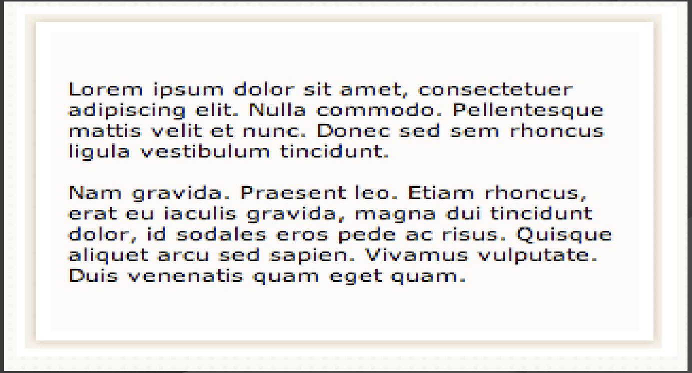
Multiple Backgrounds

The new ability to use multiple backgrounds is a great time saver, allowing you to achieve effects which previously required more than one div. Whether it will be possible to combine this with background-size will be interesting to see.

The example below uses one background for the top border, one repeated vertically for the left and right borders and a third at the bottom.



```
.multiplebackgrounds {
height: 150px;
width: 270px;
padding: 40px 20px 20px 20px;
background: url(top.gif) top left no-repeat,
url(bottom.gif) bottom left no-repeat,
url(middle.gif) left repeat-y;
}
```

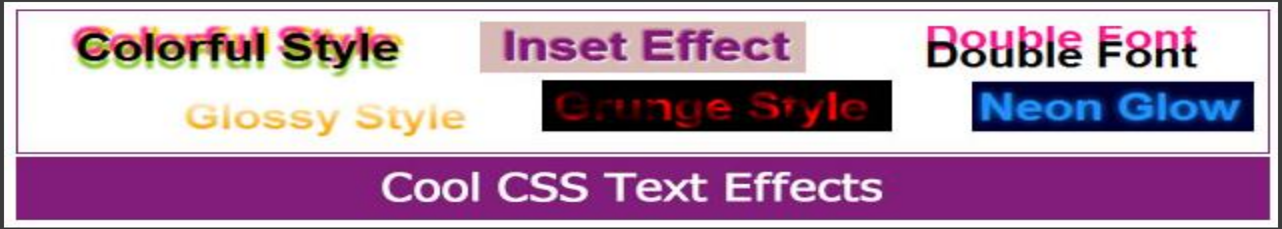


Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla commodo. Pellentesque mattis velit et nunc. Donec sed sem rhoncus ligula vestibulum tincidunt.

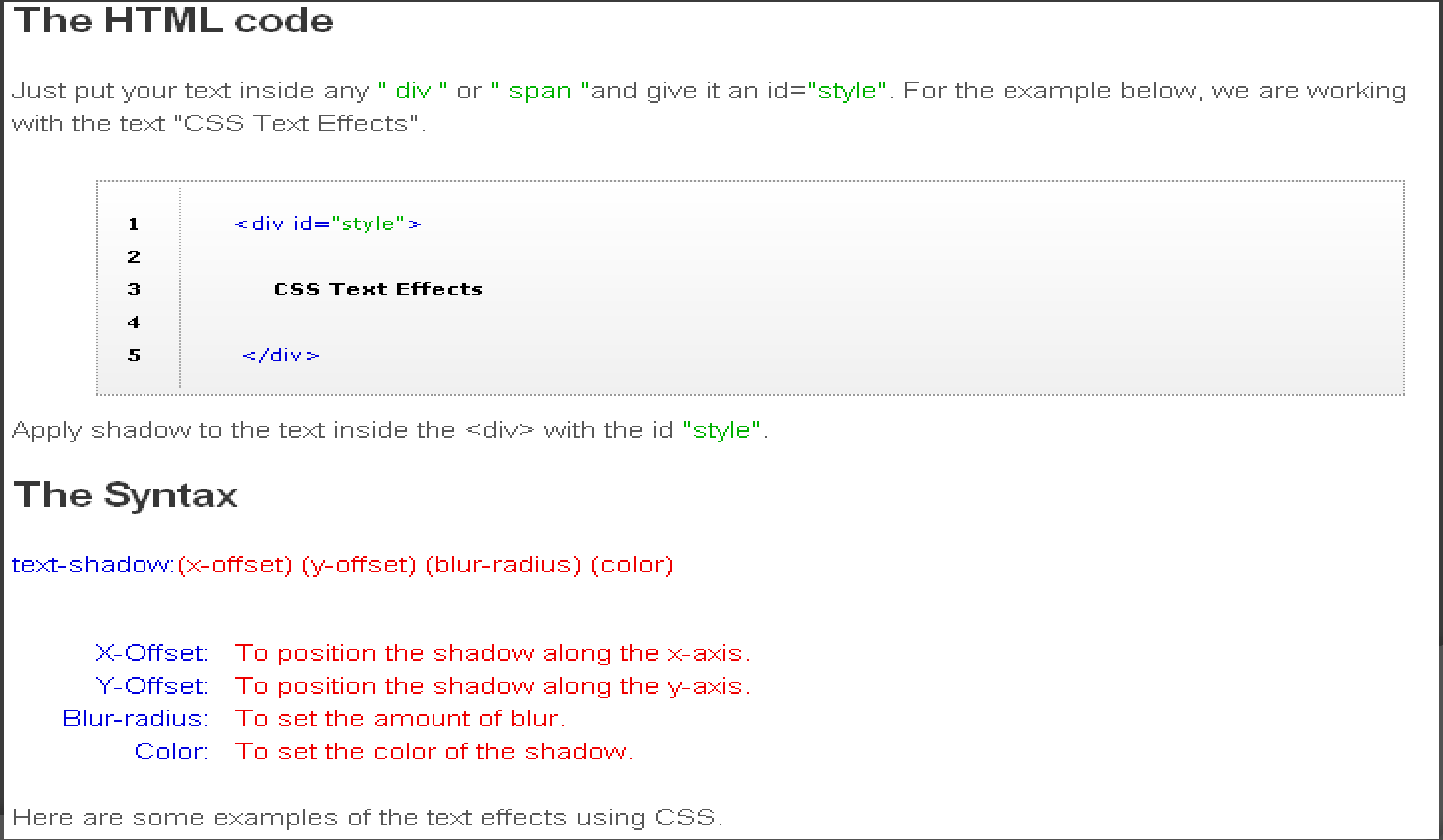
Nam gravida. Praesent leo. Etiam rhoncus, erat eu iaculis gravida, magna dui tincidunt dolor, id sodales eros pede ac risus. Quisque aliquet arcu sed sapien. Vivamus vulputate. Duis venenatis quam eget quam.

4. CSS3 Text effects

In this tutorial we will learn how to use CSS Styles to give stylish effects to text.
Learn how to create shadow effects, inset effects, neon style effects, glossy styles, grunge style effects and more!



4. CSS3 Text effects



The HTML code

Just put your text inside any " div " or " span "and give it an id="style". For the example below, we are working with the text "CSS Text Effects".

```
<div id="style">

    CSS Text Effects

 </div>
```

Apply shadow to the text inside the <div> with the id "style".

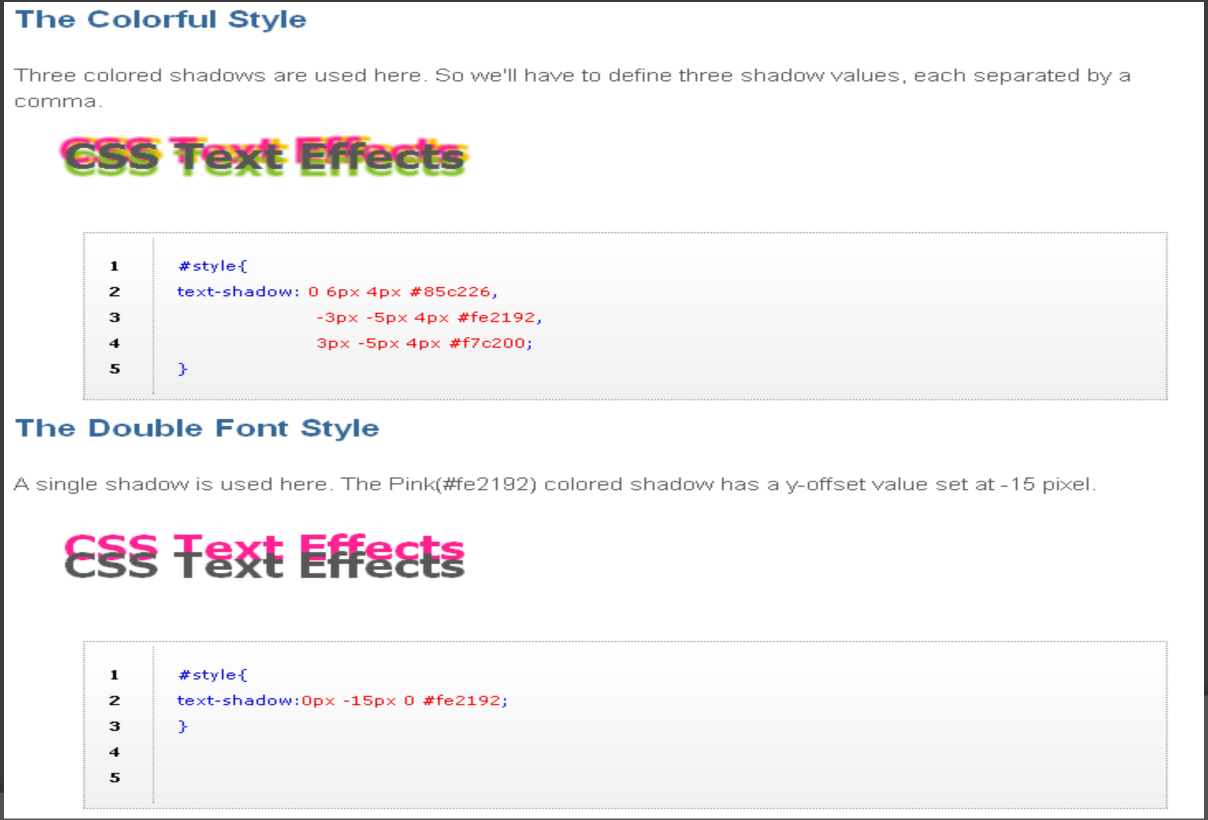
The Syntax

text-shadow:(x-offset) (y-offset) (blur-radius) (color)

X-Offset:	To position the shadow along the x-axis.
Y-Offset:	To position the shadow along the y-axis.
Blur-radius:	To set the amount of blur.
Color:	To set the color of the shadow.

Here are some examples of the text effects using CSS.

4. CSS3 Text effects



The Colorful Style

Three colored shadows are used here. So we'll have to define three shadow values, each separated by a comma.

CSS Text Effects

```
#style{
text-shadow: 0 6px 4px #85c226,
                -3px -5px 4px #fe2192,
                3px -5px 4px #f7c200;
}
```

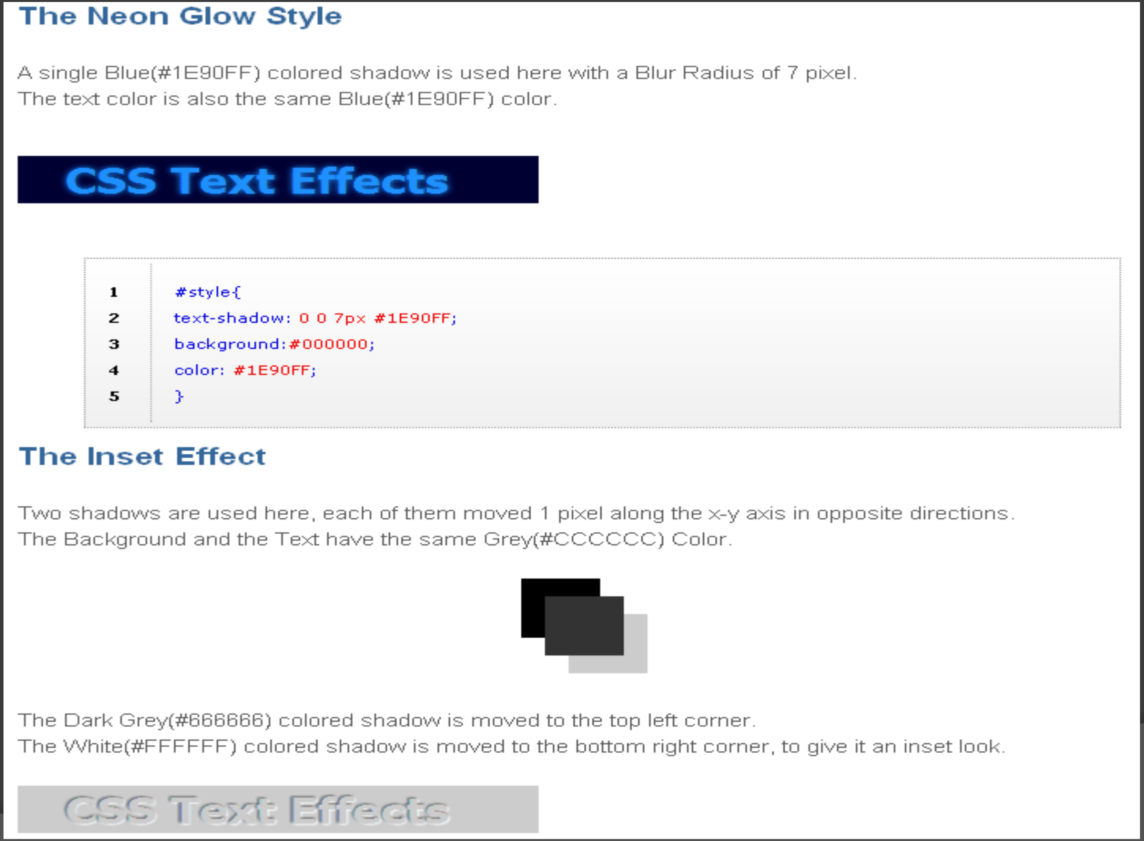
The Double Font Style

A single shadow is used here. The Pink(#fe2192) colored shadow has a y-offset value set at -15 pixel.

CSS Text Effects

```
#style{
text-shadow:0px -15px 0 #fe2192;
}
```

4. CSS3 Text effects



The Neon Glow Style

A single Blue(#1E90FF) colored shadow is used here with a Blur Radius of 7 pixel.
The text color is also the same Blue(#1E90FF) color.

CSS Text Effects

```
#style{
text-shadow: 0 0 7px #1E90FF;
background:#000000;
color: #1E90FF;
}
```

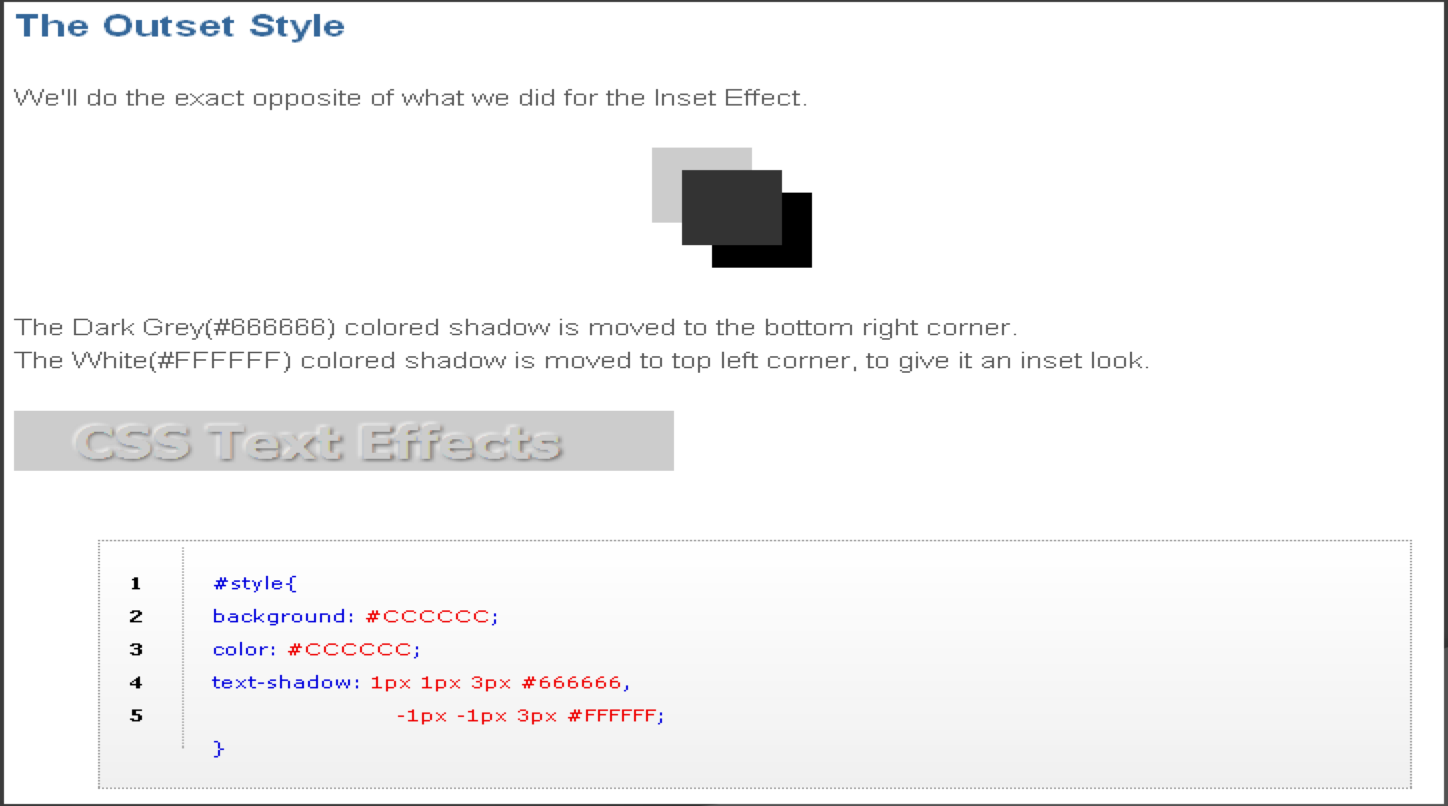
The Inset Effect

Two shadows are used here, each of them moved 1 pixel along the x-y axis in opposite directions.
The Background and the Text have the same Grey(#CCCCCC) Color.

The Dark Grey(#666666) colored shadow is moved to the top left corner.
The White(#FFFFFF) colored shadow is moved to the bottom right corner, to give it an inset look.

CSS Text Effects

3. CSS3 Text effects



The Outset Style

We'll do the exact opposite of what we did for the Inset Effect.

The Dark Grey(#666666) colored shadow is moved to the bottom right corner.
The White(#FFFFFF) colored shadow is moved to top left corner, to give it an inset look.

CSS Text Effects

```
#style{
background: #CCCCCC;
color: #CCCCCC;
text-shadow: 1px 1px 3px #666666,
             -1px -1px 3px #FFFFFF;
}
```

4. CSS3 fonts

Steps to Use the fonts
We'll follow these steps to apply the fonts to your webpage.
→ **Download Fonts**
→ **Convert them Online**
→ **Use the Stylesheet Code**
→ **Use Anywhere**

→ **Download Fonts**
There are many website where you can download fonts for free.
Here are few website with a huge collection of free as well as paid fonts.

› **dafont.com**, **urbanfonts.com**, **1001freefonts.com** ‹

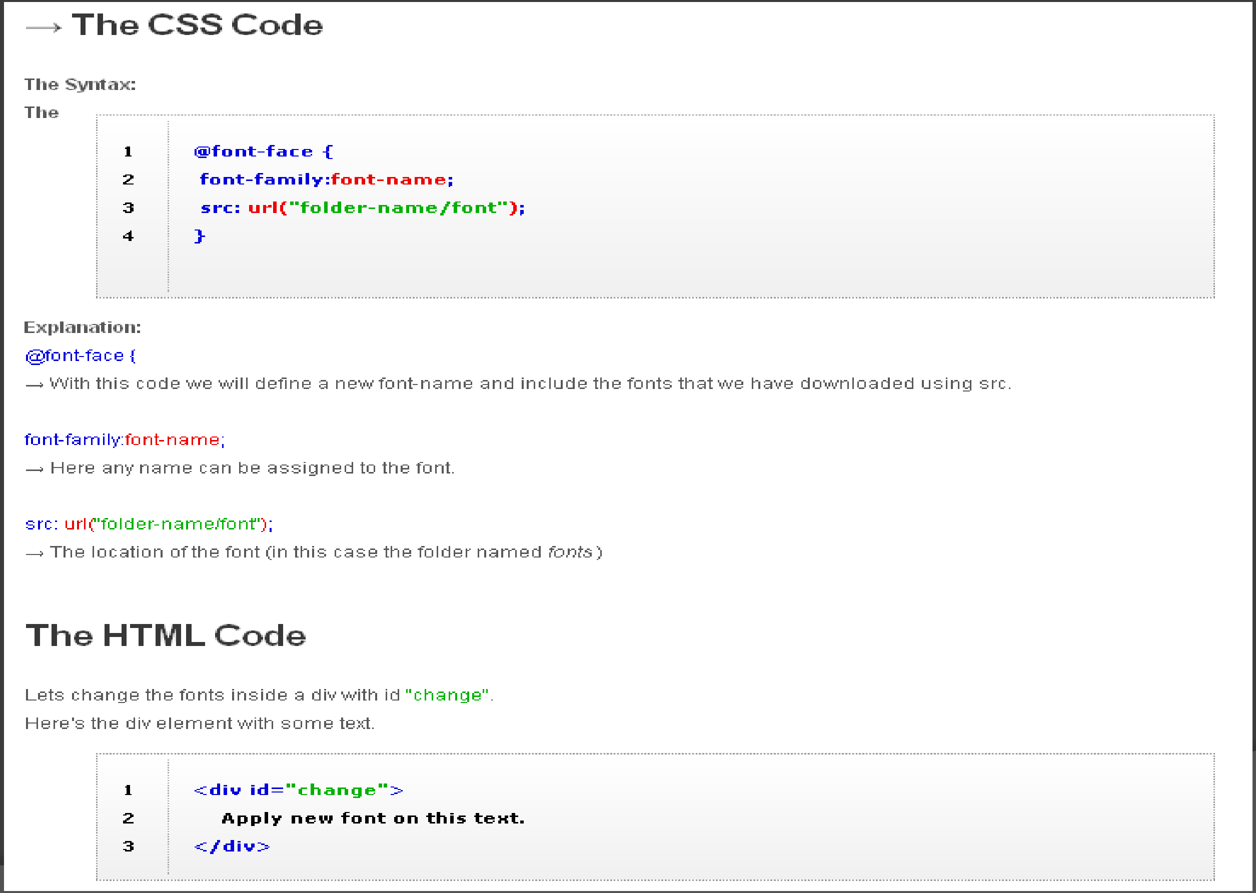
Choose the font you like and download it.

→ **Convert The Fonts**
The downloaded fonts will work on most of the browsers but won't work on IE.
To make the fonts appear on IE we'll have to convert it to a format(eot) that IE understands.
How to Convert
There are many software's available on the internet, most of them paid.
Here's a website where we can convert the TTF font to EOT font for IE for **Free**!

› **Online TTF To EOT converter** ‹

Upload the font you want to convert on this website and then download the converted font.
Notice the font extension before converting was TTF after converting changes to EOT.
Keep both these files in the same folder *Fonts*.

4. CSS3 fonts



→ The CSS Code

The Syntax:

The

```
@font-face {
 font-family:font-name;
 src: url("folder-name/font");
}
```

Explanation:

@font-face {

→ With this code we will define a new font-name and include the fonts that we have downloaded using src.

font-family:font-name;

→ Here any name can be assigned to the font.

src: url("folder-name/font");

→ The location of the font (in this case the folder named *fonts*)

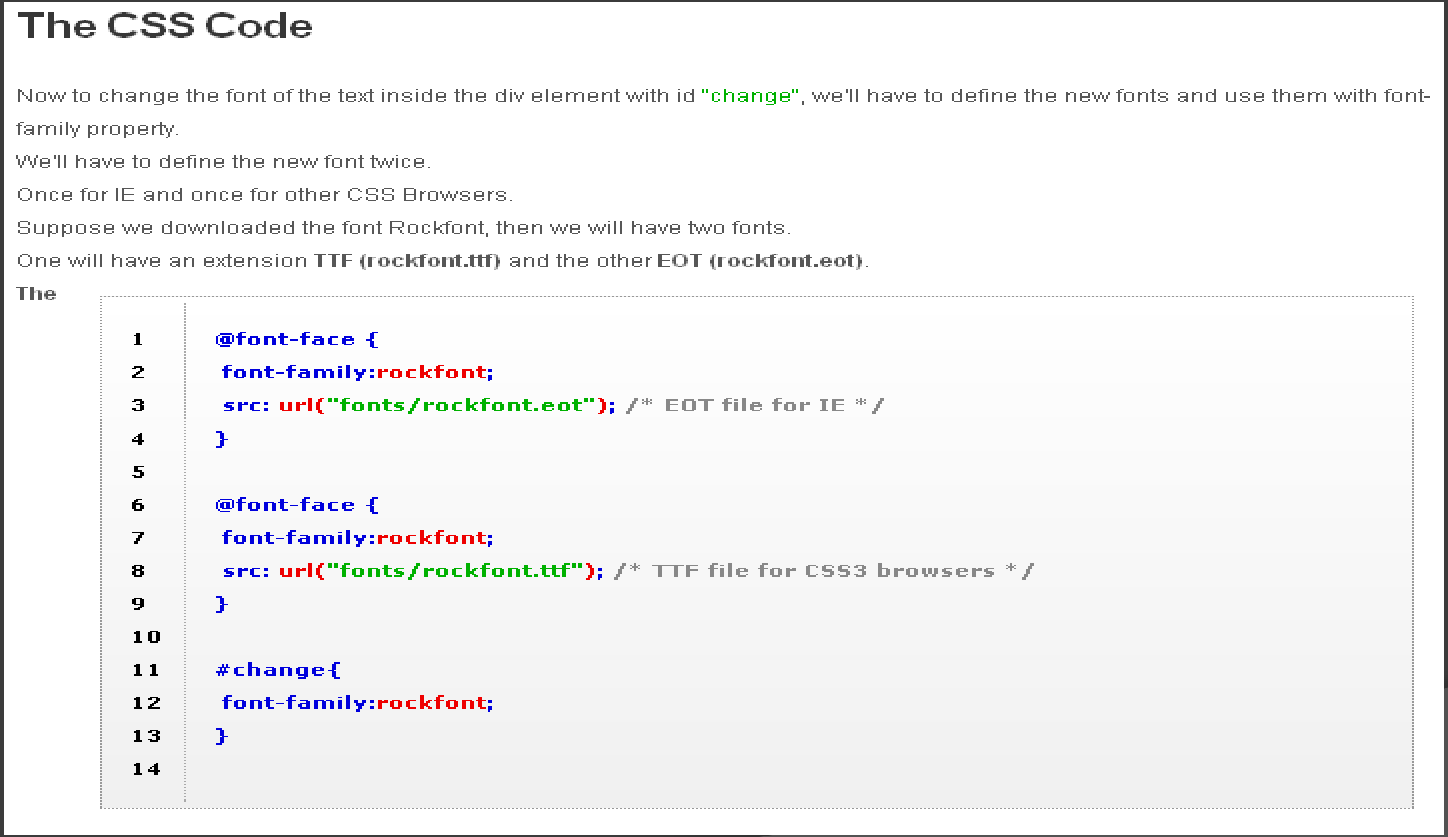
The HTML Code

Lets change the fonts inside a div with id "change".

Here's the div element with some text.

```
<div id="change">
   Apply new font on this text.
</div>
```

5. CSS3 fonts



The CSS Code

Now to change the font of the text inside the div element with id "change", we'll have to define the new fonts and use them with font-family property.

We'll have to define the new font twice.

Once for IE and once for other CSS Browsers.

Suppose we downloaded the font Rockfont, then we will have two fonts.

One will have an extension **TTF (rockfont.ttf)** and the other **EOT (rockfont.eot)**.

The

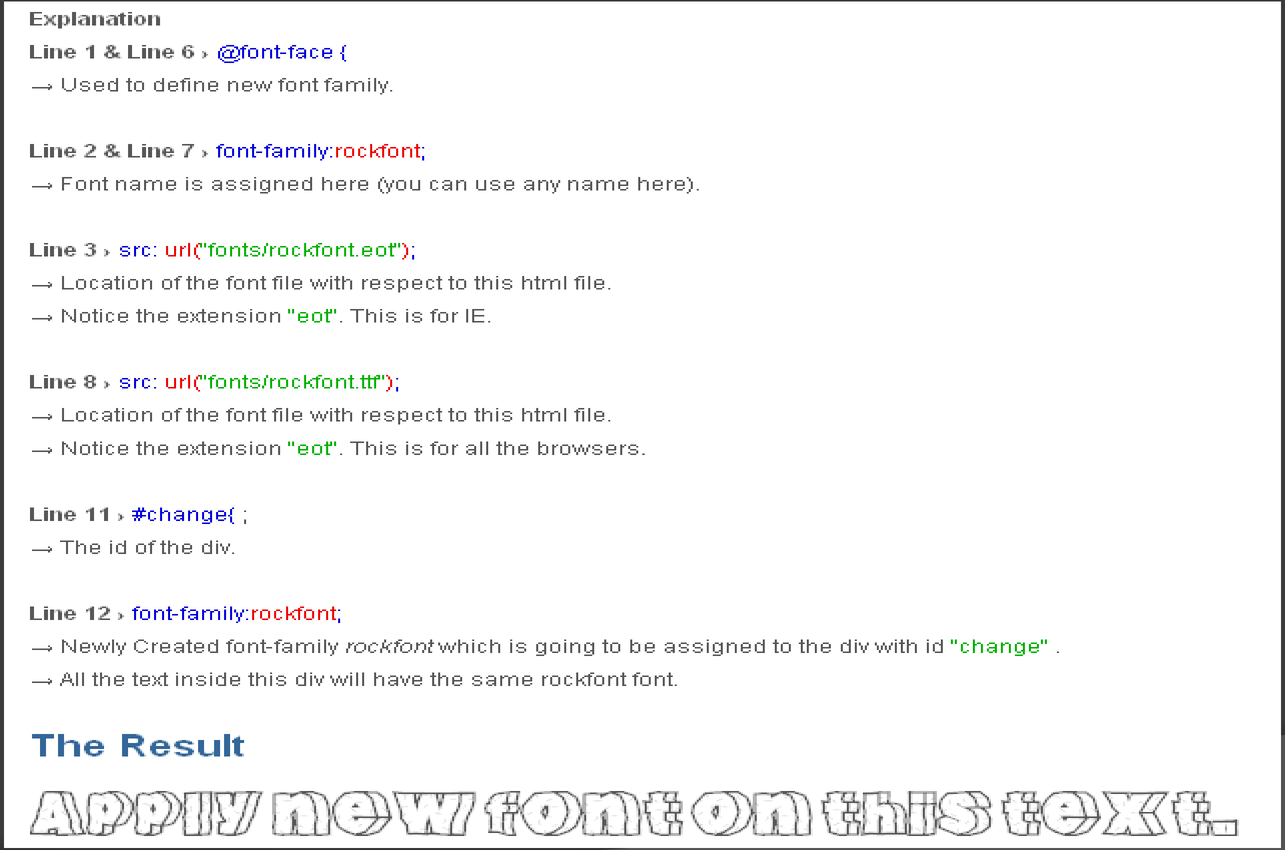
```
@font-face {
 font-family:rockfont;
 src: url("fonts/rockfont.eot"); /* EOT file for IE */
}

@font-face {
 font-family:rockfont;
 src: url("fonts/rockfont.ttf"); /* TTF file for CSS3 browsers */
}

#change{
 font-family:rockfont;
}

```

5. CSS3 fonts



Explanation

Line 1 & Line 6 › @font-face {

→ Used to define new font family.

Line 2 & Line 7 › font-family:rockfont;

→ Font name is assigned here (you can use any name here).

Line 3 › src: url("fonts/rockfont.eot");

→ Location of the font file with respect to this html file.

→ Notice the extension "eot". This is for IE.

Line 8 › src: url("fonts/rockfont.ttf");

→ Location of the font file with respect to this html file.

→ Notice the extension "eot". This is for all the browsers.

Line 11 › #change{ ;

→ The id of the div.

Line 12 › font-family:rockfont;

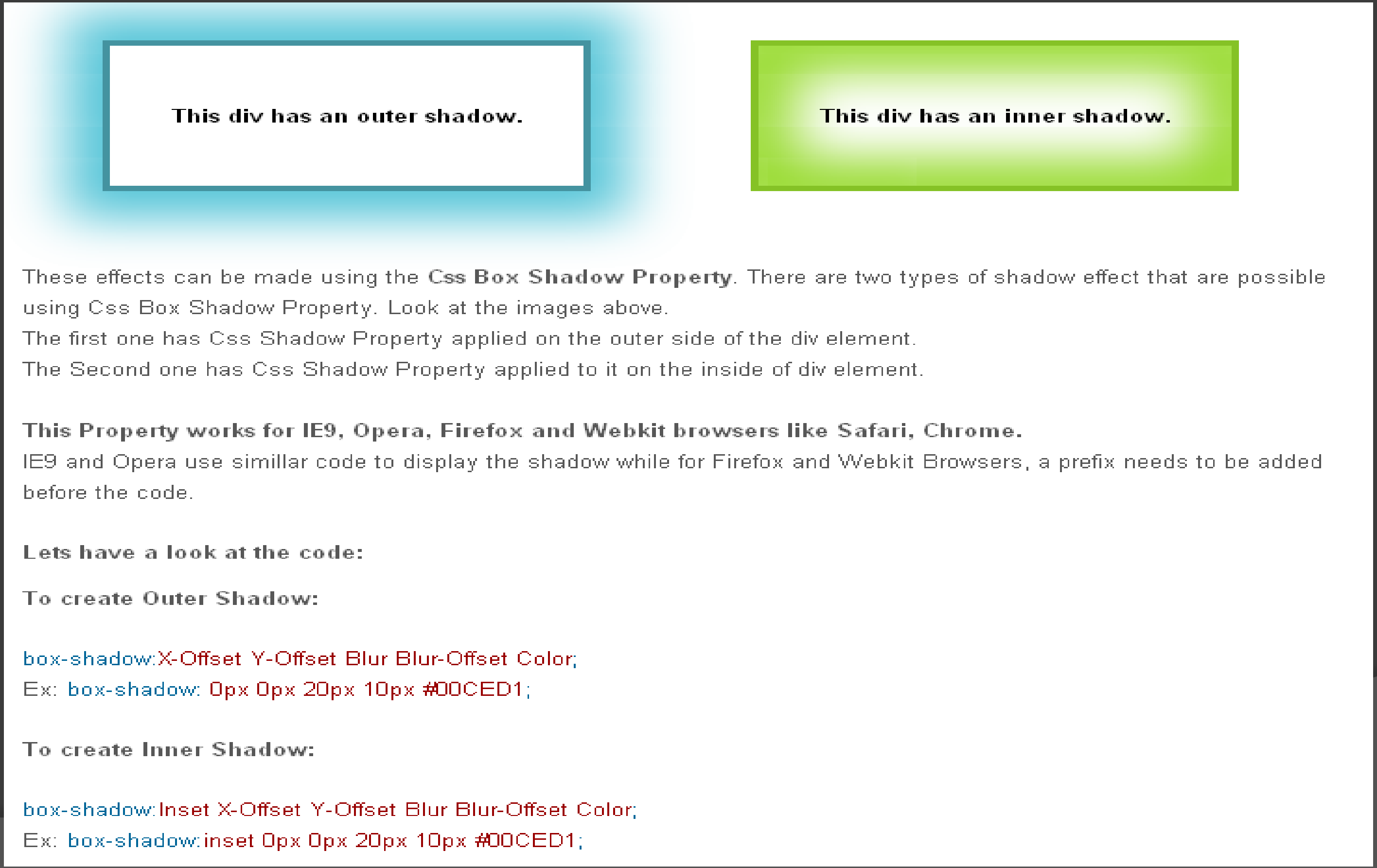
→ Newly Created font-family *rockfont* which is going to be assigned to the div with id "change" .

→ All the text inside this div will have the same rockfont font.

The Result

Apply new font on this text.

6. CSS3 shadows



This div has an outer shadow.

This div has an inner shadow.

These effects can be made using the **Css Box Shadow Property**. There are two types of shadow effect that are possible using Css Box Shadow Property. Look at the images above.
The first one has Css Shadow Property applied on the outer side of the div element.
The Second one has Css Shadow Property applied to it on the inside of div element.

This Property works for IE9, Opera, Firefox and Webkit browsers like Safari, Chrome.
IE9 and Opera use simillar code to display the shadow while for Firefox and Webkit Browsers, a prefix needs to be added before the code.

Lets have a look at the code:

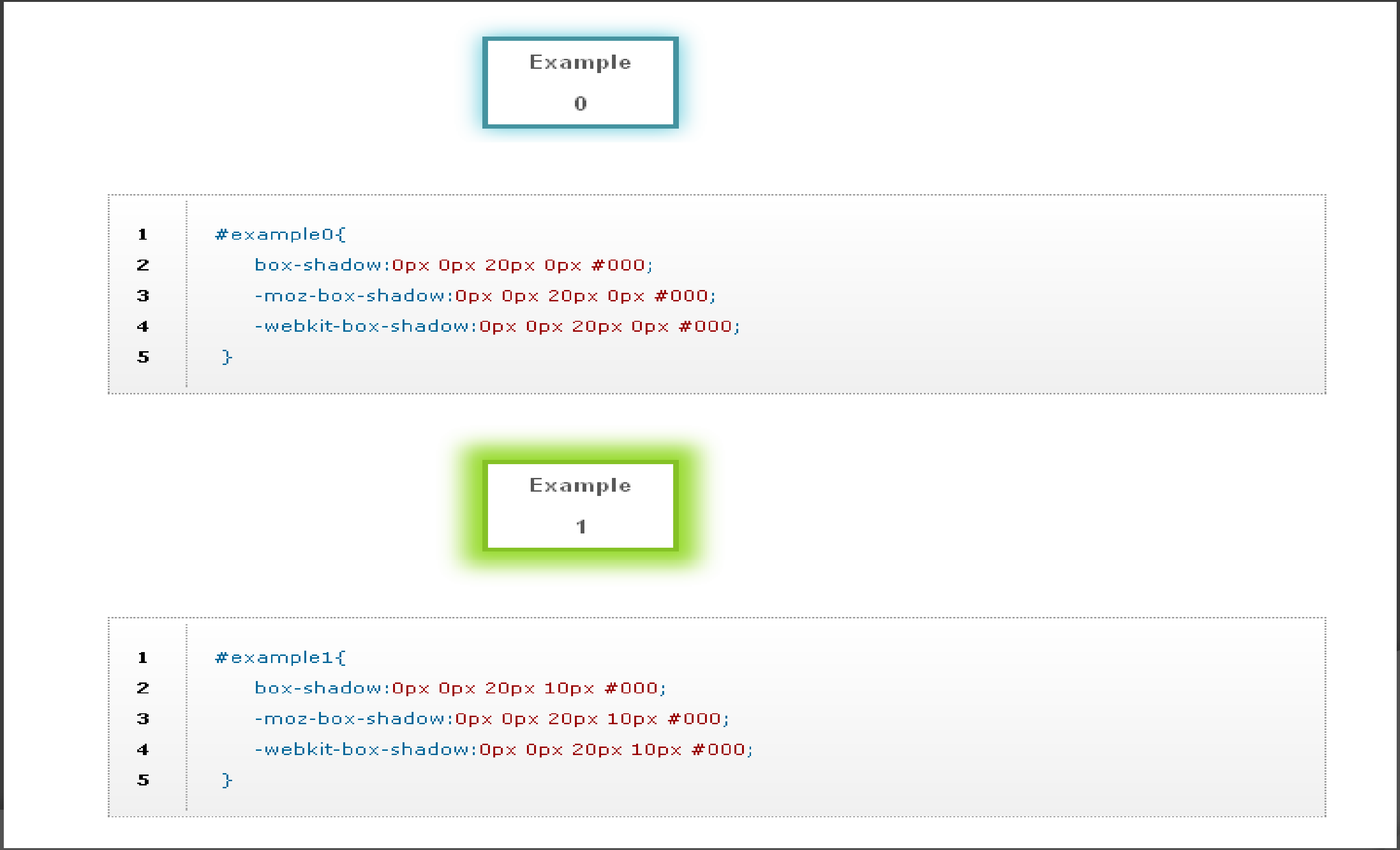
To create Outer Shadow:

box-shadow:X-Offset Y-Offset Blur Blur-Offset Color;
Ex: box-shadow: 0px 0px 20px 10px #00CED1;

To create Inner Shadow:

box-shadow:Inset X-Offset Y-Offset Blur Blur-Offset Color;
Ex: box-shadow:inset 0px 0px 20px 10px #00CED1;

6. CSS3 shadows



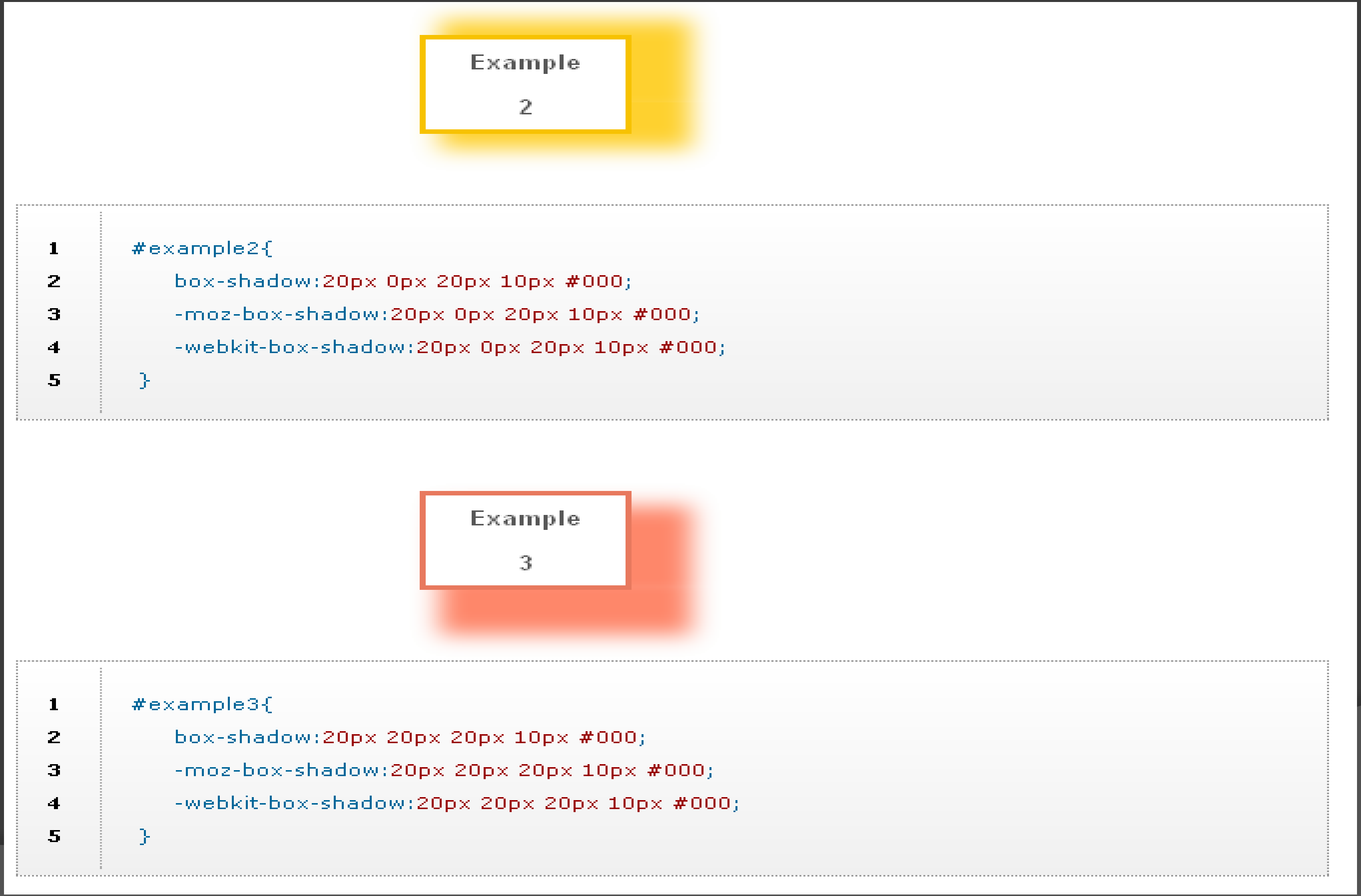
Example 0

```
#example0{
    box-shadow:0px 0px 20px 0px #000;
    -moz-box-shadow:0px 0px 20px 0px #000;
    -webkit-box-shadow:0px 0px 20px 0px #000;
 }
```

Example 1

```
#example1{
    box-shadow:0px 0px 20px 10px #000;
    -moz-box-shadow:0px 0px 20px 10px #000;
    -webkit-box-shadow:0px 0px 20px 10px #000;
 }
```

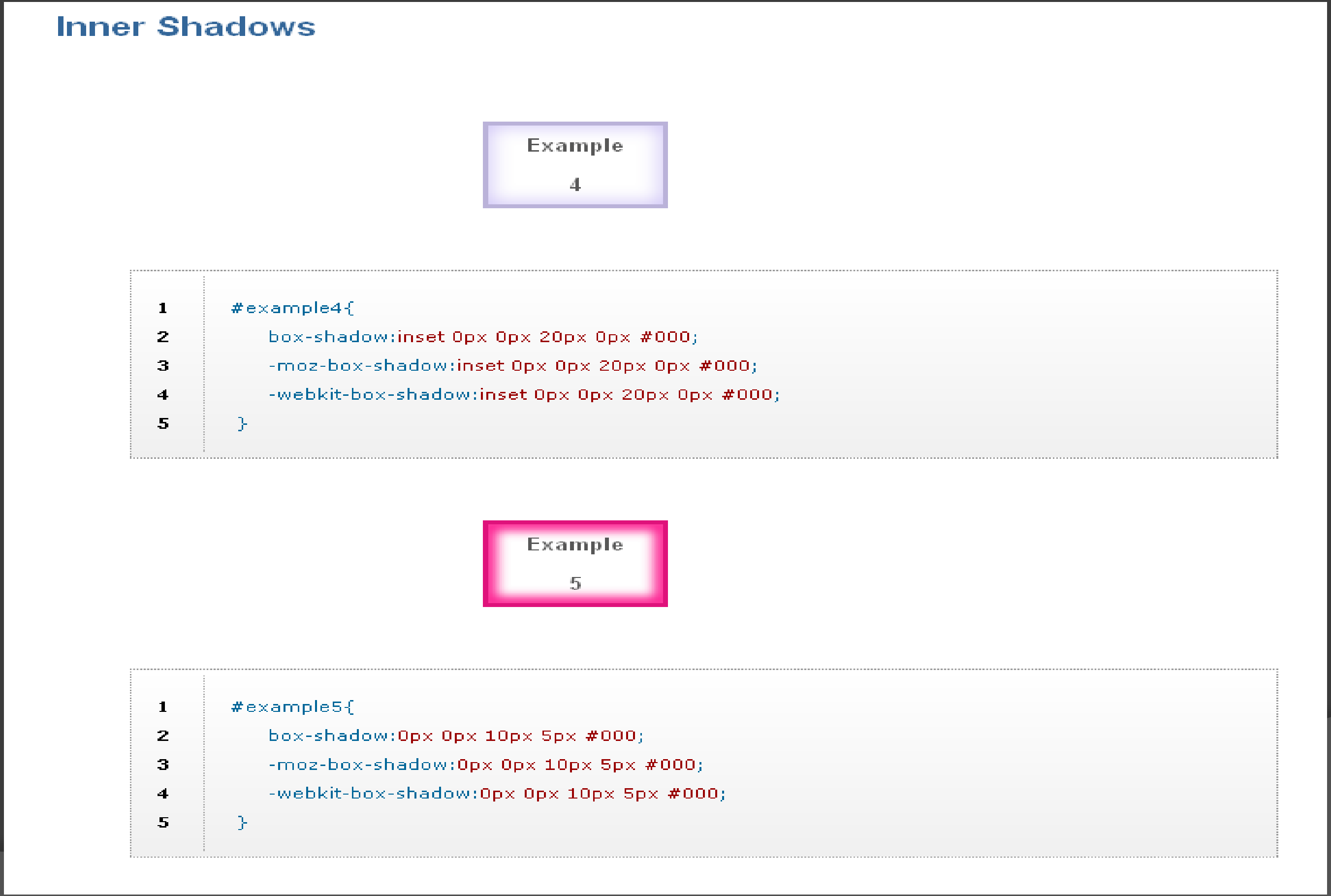
6. CSS3 shadows



```
#example2{
    box-shadow:20px 0px 20px 10px #000;
    -moz-box-shadow:20px 0px 20px 10px #000;
    -webkit-box-shadow:20px 0px 20px 10px #000;
 }
```

```
#example3{
    box-shadow:20px 20px 20px 10px #000;
    -moz-box-shadow:20px 20px 20px 10px #000;
    -webkit-box-shadow:20px 20px 20px 10px #000;
 }
```

6. CSS3 shadows



Inner Shadows

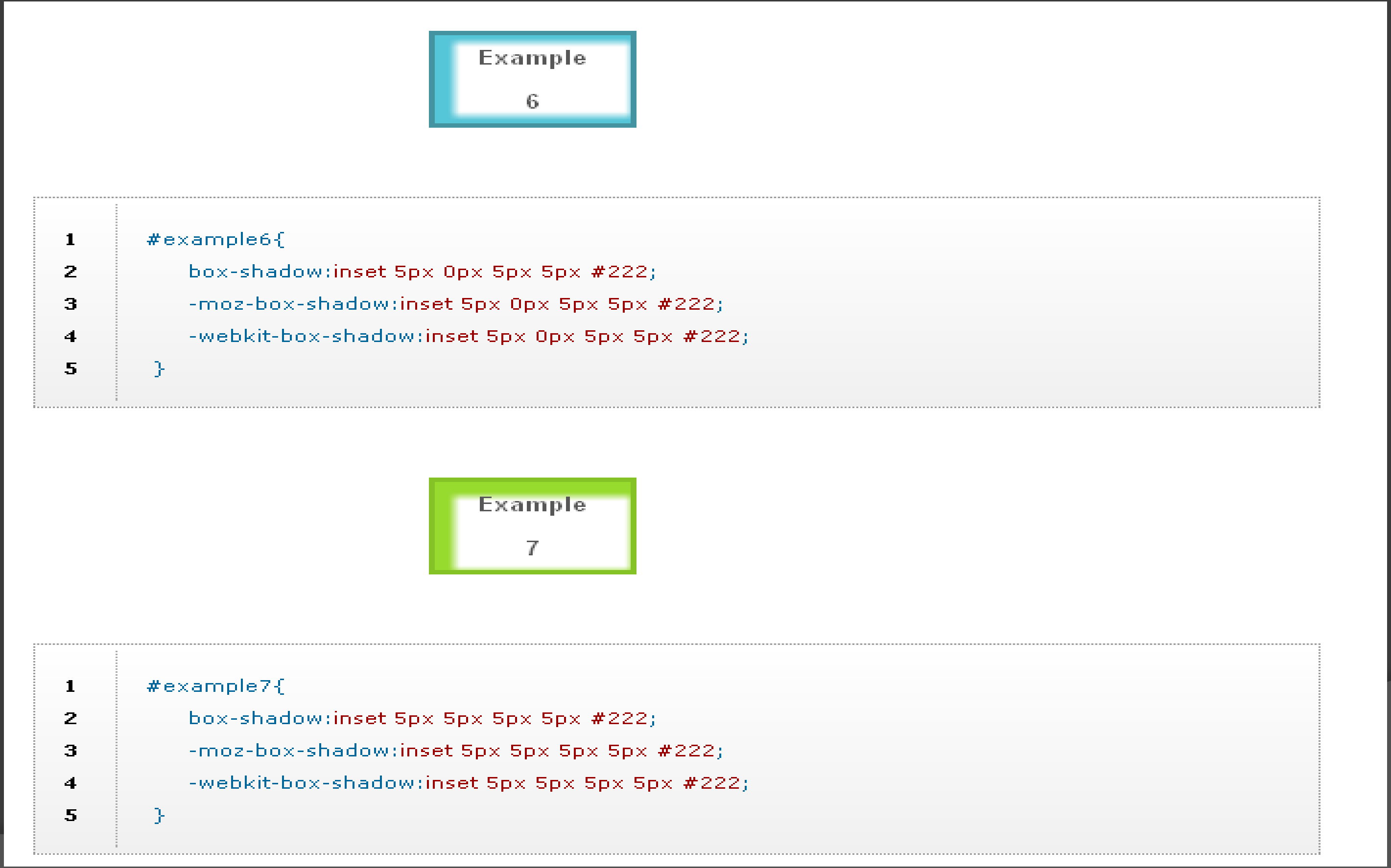
Example 4

```
#example4{
    box-shadow:inset 0px 0px 20px 0px #000;
    -moz-box-shadow:inset 0px 0px 20px 0px #000;
    -webkit-box-shadow:inset 0px 0px 20px 0px #000;
}
```

Example 5

```
#example5{
    box-shadow:0px 0px 10px 5px #000;
    -moz-box-shadow:0px 0px 10px 5px #000;
    -webkit-box-shadow:0px 0px 10px 5px #000;
}
```

6. CSS3 shadows



Example 6

```
#example6{
    box-shadow:inset 5px 0px 5px 5px #222;
    -moz-box-shadow:inset 5px 0px 5px 5px #222;
    -webkit-box-shadow:inset 5px 0px 5px 5px #222;
 }
```

Example 7

```
#example7{
    box-shadow:inset 5px 5px 5px 5px #222;
    -moz-box-shadow:inset 5px 5px 5px 5px #222;
    -webkit-box-shadow:inset 5px 5px 5px 5px #222;
 }
```

6. CSS3 Transforms & Transitions

Transforms

CSS3 transform property lets you translate, rotate, scale, or skew any element on the page.
While some of these effects were possible using previously existing CSS features
(like relative and absolute positioning), CSS3 gives you unprecedented control over many more aspects of an element’ s appearance.

Translation

Translation functions allow you to move elements left, right, up, or down.

These functions are similar to the behaviour of position: relative; where you declare top and left.
When you employ a translation function, you’ re moving elements without impacting the flow of the document.

Unlike position: relative, which allows you to position an element either against its current position or against a parent or other ancestor, a translated element can only be moved relative to its current position.

The translate(x,y) function moves an element by x from the left, and y from the top:

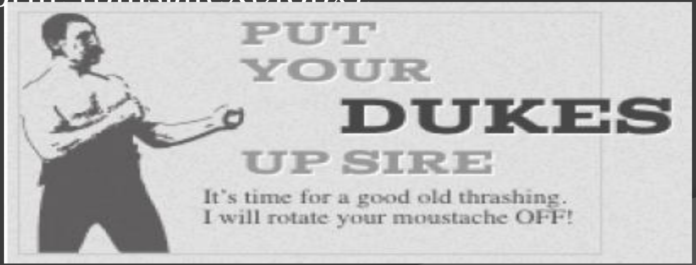
6. CSS3 Transforms & Transitions

Examples

PUT
YOUR
DUKES
UP SIRE
It's time for a good old thrashing.
I will rotate your moustache OFF!

let' s say we want to move the word "dukes" over to the right when the user hovers over it,

```
#ourExample h1:hover span {
color: #484848;
-webkit-transform: translateX(40px);
-moz-transform: translateX(40px);
-ms-transform: translateX(40px);
-o-transform:translateX(40px);
transform: translateX(40px);
}
```


PUT
YOUR
DUKES
UP SIRE
It's time for a good old thrashing.
I will rotate your moustache OFF!

6. CSS3 Transforms & Transitions

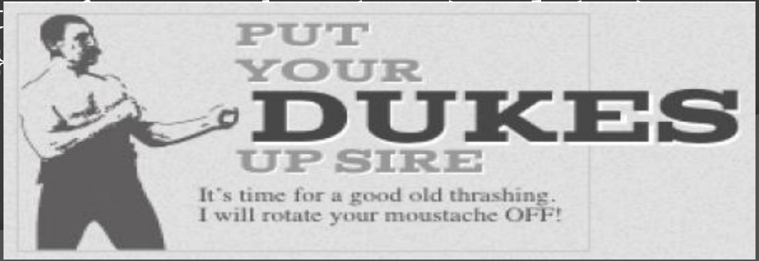
Scaling

The scale(x,y) function scales an element by the defined factors horizontally and vertically, espectively. If only one value is provided, it will be used for both the x and y scaling.

PUT
YOUR
DUKES
UP SIRE
It's time for a good old thrashing.
I will rotate your moustache OFF!

Lets scale the same example

```
#ourExample h1:hover span {
color: #484848;
-webkit-transform: translateX(40px) scale(1.5);
-moz-transform: translateX(40px) scale(1.5);
-ms-transform: translateX(40px) scale(1.5);
-o-transform:translateX(40px) scale(1.5);
t
}
```


PUT
YOUR
DUKES
UP SIRE
It's time for a good old thrashing.
I will rotate your moustache OFF!

6. CSS3 Transforms & Transitions

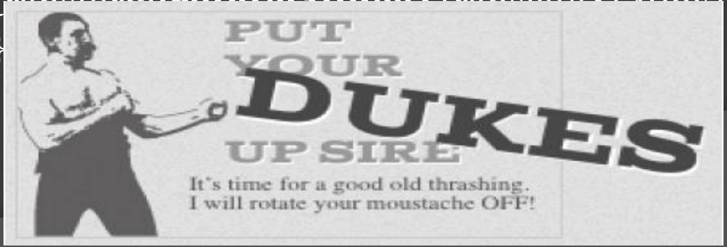
Rotation

The rotate() function rotates an element around the point of origin (as with scale,by default this is the element’ s center), by a specified angle value. Generally, angles are declared in degrees, with positive degrees moving clockwise and negativemoving counter-clockwise.

PUT
YOUR
DUKES
UP SIRE
It's time for a good old thrashing.
I will rotate your moustache OFF!

Lets scale the same example

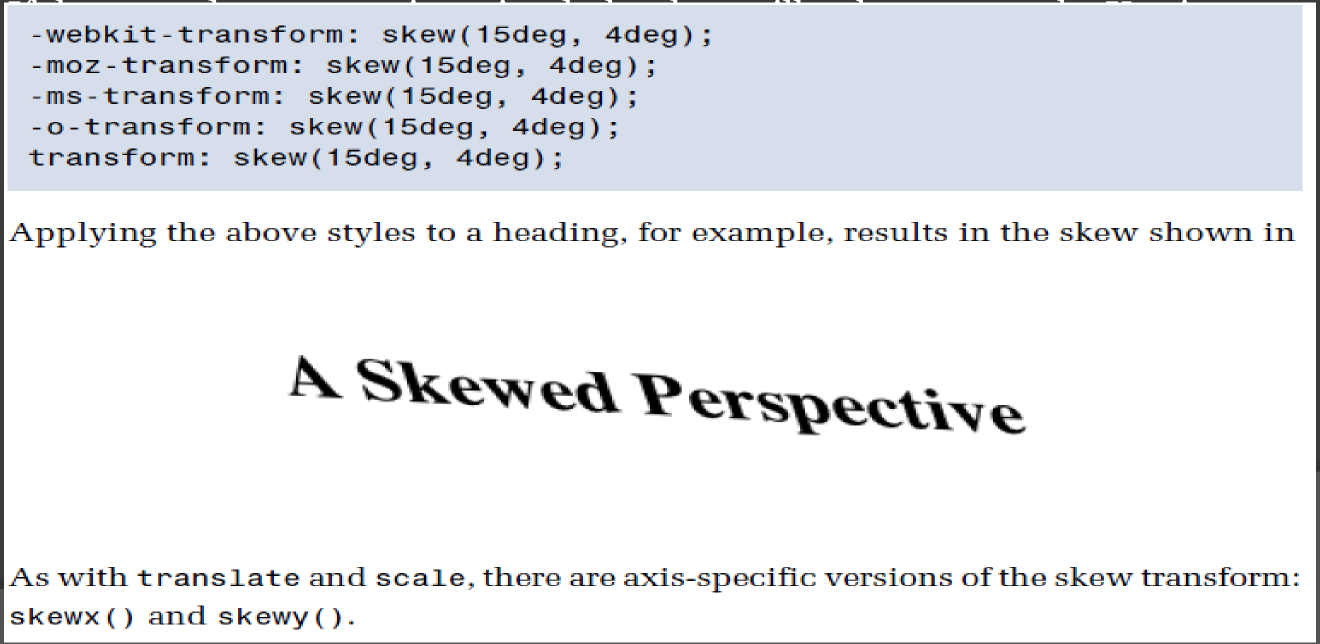
```
#ourExample h1:hover span {
color: #484848;
-webkit-transform:rotate(10deg) translateX(40px) scale(1.5);
-moz-transform:rotate(10deg) translateX(40px) scale(1.5);
-ms-transform:rotate(10deg) translateX(40px) scale(1.5);
-o-transform:rotate(10deg) translateX(40px) scale(1.5);
transform:rotate(10deg) translateX(40px) scale(1.5);
}
```


PUT
YOUR
DUKES
UP SIRE
It's time for a good old thrashing.
I will rotate your moustache OFF!

6. CSS3 Transforms & Transitions

Skew

The skew(x,y) function specifies a skew along the X and Y axes. As you’ d expect, the x specifies the skew on the X axis, and the y specifies the skew on the Y axis.



```
-webkit-transform: skew(15deg, 4deg);
-moz-transform: skew(15deg, 4deg);
-ms-transform: skew(15deg, 4deg);
-o-transform: skew(15deg, 4deg);
transform: skew(15deg, 4deg);
```

Applying the above styles to a heading, for example, results in the skew shown in

A Skewed Perspective

As with `translate` and `scale`, there are axis-specific versions of the skew transform: `skewx()` and `skewy()`.

6. CSS3 Transforms & Transitions

Transitions

Transitions allow the values of CSS properties to change over time, essentially providing simple animations. For example, if a link changes color on hover, you can have it gradually fade from one color to the other, instead of a sudden change

Here are the steps to create a simple transition using only CSS:

1. Declare the original state of the element in the default style declaration.
2. Declare the final state of your transitioned element; for example, in a hover state.
3. Include the transition functions in your default style declaration, using a few different properties: transition-property, transition-duration,transition-timing-function, and transition-delay.

list of properties that can be transitioned

■ background-color and background-position ■ border-color, border-spacing, and border-width ■ bottom, top, left, and right ■ clip ■ color ■ crop ■ font-size and font-weight ■ height and width ■ letter-spacing ■ line-height ■ margin ■ max-height, max-width, min-height, and min-width ■ opacity ■ outline-color, outline-offset, and outline-width ■ padding ■ text-indent ■ text-shadow ■ vertical-align ■ visibility ■ word-spacing ■ z-index

6. CSS3 Transforms & Transitions

Transition-duration

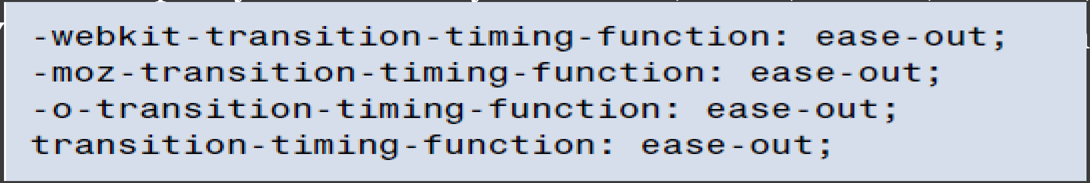
The transition-duration property sets how long the transition will take. You can specify this either in seconds (s) or milliseconds (ms). We’ d like our animation tobe fairly quick, so we’ ll specify 0.2

```
-webkit-transition-duration: 0.2s;
-moz-transition-duration: 0.2s;
-o-transition-duration: 0.2s;
transition-duration: 0.2s;
```

6. CSS3 Transforms & Transitions

Transition-timing-function

The transition-timing-function lets you control the pace of the transition in even more granular detail.
Do you want your animation to start off slow and get faster, start off fast and end slowe
You can specify one of the key terms ease, linear, ease-in, ease-out, or easein-out.
d try them all.

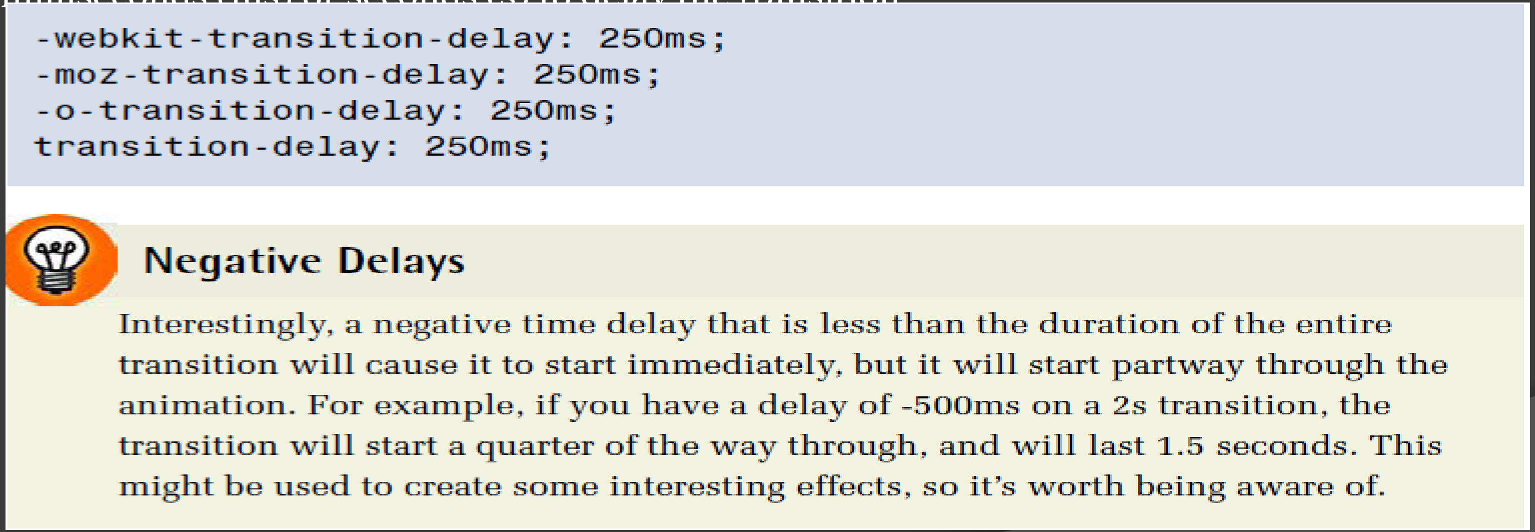


```
-webkit-transition-timing-function: ease-out;
-moz-transition-timing-function: ease-out;
-o-transition-timing-function: ease-out;
transition-timing-function: ease-out;
```

6. CSS3 Transforms & Transitions

Transition-delay

Finally, by using the transition-delay property, it' s also possible to introduce a delay before the animation begins. Normally, a transition begins immediately, so the default is 0. Include the number of milliseconds (ms) or seconds (s) to delay the transition:



```
-webkit-transition-delay: 250ms;
-moz-transition-delay: 250ms;
-o-transition-delay: 250ms;
transition-delay: 250ms;
```

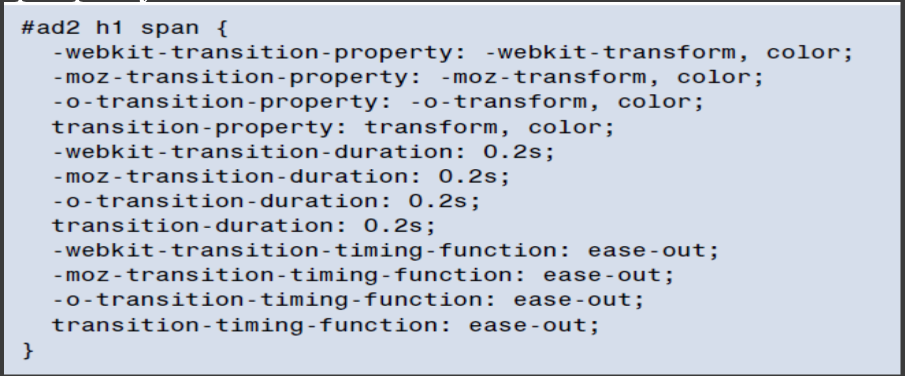
Negative Delays

Interestingly, a negative time delay that is less than the duration of the entire transition will cause it to start immediately, but it will start partway through the animation. For example, if you have a delay of -500ms on a 2s transition, the transition will start a quarter of the way through, and will last 1.5 seconds. This might be used to create some interesting effects, so it's worth being aware of.

6. CSS3 Transforms & Transitions

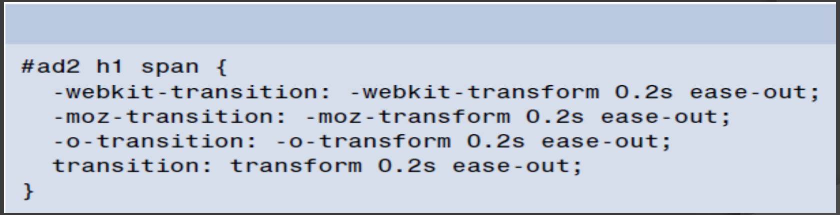
Transition-shorthand property

With four transition properties and three vendor prefixes, you could wind up with 16 lines of CSS for a single transition. Fortunately, as with other properties, there’ s a shorthand available. The transition property is shorthand for the four transition functions described above. Let’ s take another look at our



```
#ad2 h1 span {
  -webkit-transition-property: -webkit-transform, color;
  -moz-transition-property: -moz-transform, color;
  -o-transition-property: -o-transform, color;
  transition-property: transform, color;
  -webkit-transition-duration: 0.2s;
  -moz-transition-duration: 0.2s;
  -o-transition-duration: 0.2s;
  transition-duration: 0.2s;
  -webkit-transition-timing-function: ease-out;
  -moz-transition-timing-function: ease-out;
  -o-transition-timing-function: ease-out;
  transition-timing-function: ease-out;
}
```

shorthand



```
#ad2 h1 span {
  -webkit-transition: -webkit-transform 0.2s ease-out;
  -moz-transition: -moz-transform 0.2s ease-out;
  -o-transition: -o-transform 0.2s ease-out;
  transition: transform 0.2s ease-out;
}
```

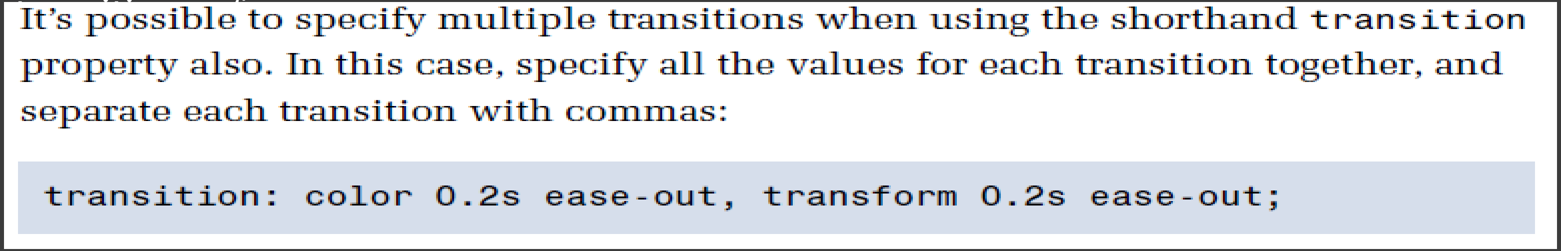
Note that order of the values is important and must be as follows (though you don’ t need to specify all four values):

1. transition-property
2. transition-duration
3. transition-function

6. CSS3 Transforms & Transitions

Multiple transition

With four transition properties and three vendor prefixes, you could wind up with 16 lines of CSS for a single transition. Fortunately, as with other properties, there’ s a shorthand available. The transition property is shorthand for the four transition functions described above. Let’ s take another look at our



It’s possible to specify multiple transitions when using the shorthand `transition` property also. In this case, specify all the values for each transition together, and separate each transition with commas:

```
transition: color 0.2s ease-out, transform 0.2s ease-out;
```

6. CSS3 Animation

Transitions animate elements over time; however, they're limited in what they can do. You can define starting and ending states, but there's no fine-grained control over any intermediate states. CSS animations, unlike transitions, allow you to control each step of an animation via **keyframes**. If you've ever worked with Flash, you're likely very familiar with the concept of keyframes; if not, don't worry, it's fairly straightforward.

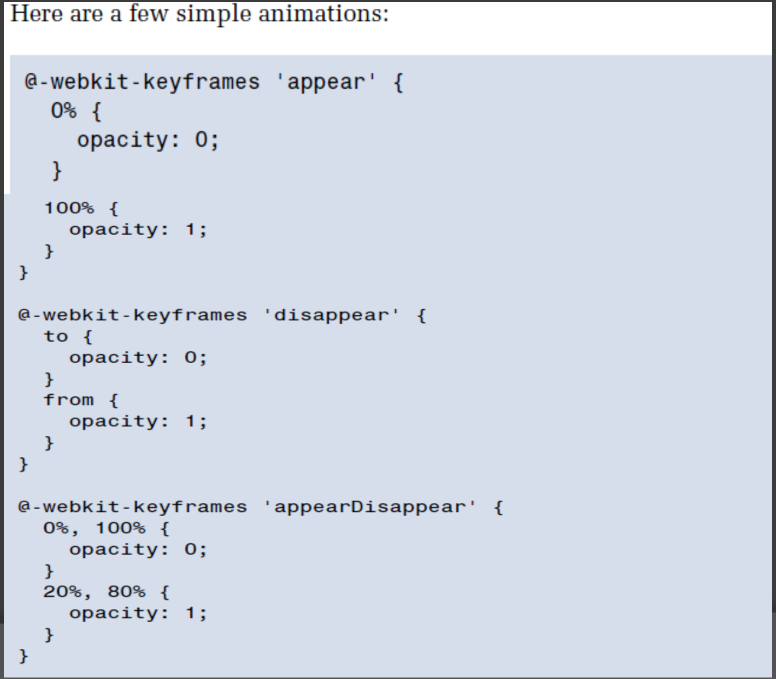
A keyframe is a snapshot that defines a starting or end point of any smooth transition. With CSS transitions, we're essentially limited to defining the first and last keyframes. CSS animations allow us to add any number of keyframes in between, to guide our animation in more complex ways.

Animation properties:

animation-name
animation-duration
animation-timing-function
animation-iteration-count
animation-direction
animation-delay
animation-fill-mode

6. CSS3 Animation

To animate an element in CSS, you first create a named animation, then attach it to an element in that element’ s property declaration block. Animations in themselves don’ t do anything; in order to animate
an element, you will need to associate the animation with that element.


Here are a few simple animations:

```
@-webkit-keyframes 'appear' {
  0% {
    opacity: 0;
  }
  100% {
    opacity: 1;
  }
}

@-webkit-keyframes 'disappear' {
  to {
    opacity: 0;
  }
  from {
    opacity: 1;
  }
}

@-webkit-keyframes 'appearDisappear' {
  0%, 100% {
    opacity: 0;
  }
  20%, 80% {
    opacity: 1;
  }
}
```

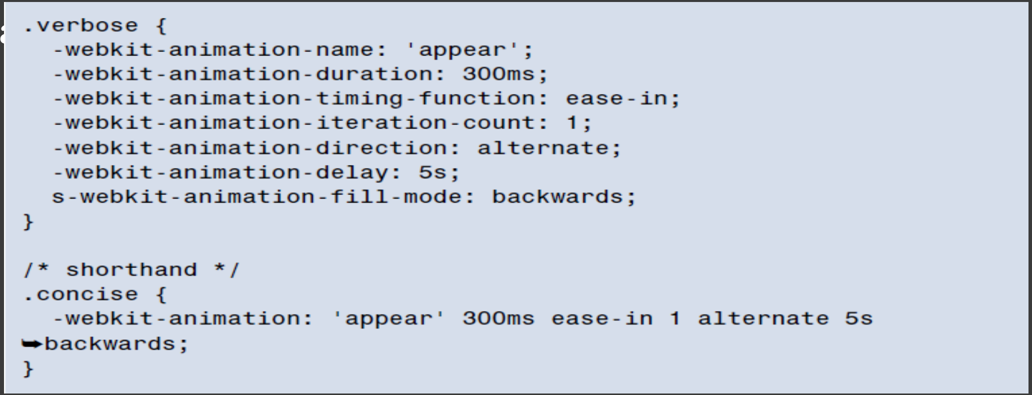
The last animation is worth paying extra attention to: we’ ve applied the same styles
to 0% and 100%, and to 20% and 80%. In this case, it means the element will start
out invisible (opacity: 0;), fade in to visible by 20% of the way through the duration,
remain visible until 80%, then fade out.

We’ ve created three animations, but they aren’ t attached to any elements. Once we have defined an animation, the next step is to apply it to one or more elements using
the various animation properties.

6. CSS3 Animation

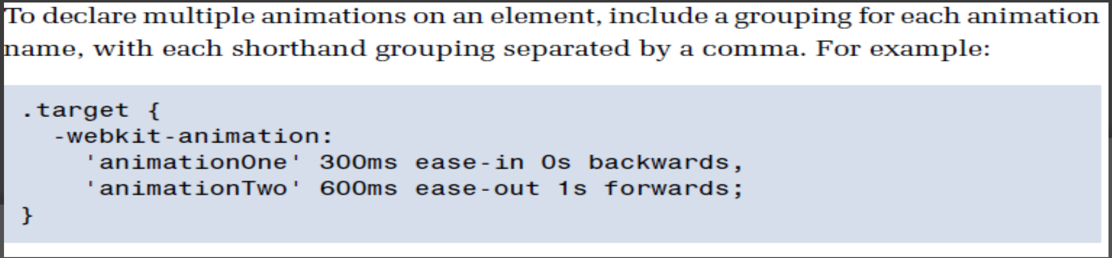
Shorthand

The animation property takes as its value a space-separated list of values for the longhand animation name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count,



```
.verbose {
  -webkit-animation-name: 'appear';
  -webkit-animation-duration: 300ms;
  -webkit-animation-timing-function: ease-in;
  -webkit-animation-iteration-count: 1;
  -webkit-animation-direction: alternate;
  -webkit-animation-delay: 5s;
  s-webkit-animation-fill-mode: backwards;
}

/* shorthand */
.concise {
  -webkit-animation: 'appear' 300ms ease-in 1 alternate 5s
➥backwards;
}
```



To declare multiple animations on an element, include a grouping for each animation name, with each shorthand grouping separated by a comma. For example:

```
.target {
  -webkit-animation:
    'animationOne' 300ms ease-in 0s backwards,
    'animationTwo' 600ms ease-out 1s forwards;
}
```

6. CSS3 multiple columns

Newspaper-style columns have been close to impossible to accomplish with CSS and HTML without forcing column breaks at fixed positions

With CSS3 columns, the browser determines when to end one column and begin the next without requiring any extra markup. You retain the flexibility to change the number of columns as well as their width, without having to go back in and alter the page's markup.

The column-count-property

The column-count property specifies the number of columns desired, and the maximum number of columns allowed. The default value of auto means that the element has one column.

Ex:
```
#primary article .content {
        -webkit-column-count: 3;
        -moz-column-count: 3;
        column-count: 3;
    }
```

6. CSS3 multiple columns

The column-gap-property

The column-gap property specifies the width of the space between columns:

Ex:
```
#primary article .content {
        -webkit-column-gap: 10px;
        -moz-column-gap: 10px;
        column-gap: 10px;
    }
```

The column-width-property

The column-width property is like having a min-width for your columns. The browser will include as many columns of at least the given width as it can to fill up the element

For example, if we have a parent that is 400 pixels wide, a 10-pixel column gap, and the column-width is declared as 150px, the browser can fit two columns:(400px width – 10px column gap) ÷ 150px width 2.6 The browser rounds down to two columns, making columns that are as large as possible in the allotted space; in this case that's 195px for each column—the total width minus the gap, divided by the number of columns. Even if the column-count were set to 3, there would still only be two columns, as there's not enough space to include three columns of the specified width. In other words, you can think Of the column-count property as specifying the *maximum* column count.

Ex:
```
#primary article .content {
        -webkit-column-width: 150px;
        -moz-column-width: 150px;
        column-width: 150px;
    }
```

6. CSS3 multiple columns

The column-rule-property

Column **rules** are essentially borders between each column. The column-rule property specifies the color, style, and width of the column rules. The rule will appear in the middle of the column gap

```
Ex: #primary article .content {
 -webkit-column-rule: 1px solid #ccc;
 -moz-column-rule: 1px solid #ccc;
 column-gap: 1px solid #ccc;
}
```

6. CSS3 selectors

Selectors are at the heart of CSS. Without selectors to target elements on the page, the only way to modify the CSS properties of an element would be to use the element's style attribute and declare the styles inline. This, of course, is ugly, awkward, and unmaintainable. So we use selectors.

Relational selectors

Descendant (E F)

You should definitely be familiar with this one. The descendant selector targets any element F that is a descendant (child, grandchild, great grandchild, and so on) of an element E.

Child (E > F)

This selector matches any element F that is a *direct child* of element E—any further nested elements will be ignored. Continuing the above example, ol > li would only target li elements directly inside the ol, and would omit those nested inside a ul.

Adjacent Sibling (E + F)

This will match any element F that shares the same parent as E, and comes *directly after* E in the markup. For example, li + li will target all li elements except the first li in a given container.

General Sibling (E ~ F)

This one's a little trickier. It will match any element F that shares the same parent as any E and comes after it in the markup. So, h1~h2 will match any h2 that follows an h1, as long as they both share the same direct parent—that is, as long as the h2 is not nested in any other element.

6. CSS3 selectors

Attribute selectors

E[attr$=val] (IE8+, WebKit, Opera, Mozilla)
Matches any element E whose attribute attr ends in val. In other words, the val matches the end of the attribute value.

E[attr*=val] (IE8+, WebKit, Opera, Mozilla)
Matches any element E whose attribute attr matches val anywhere within the attribute. In other words, the string val is matched anywhere in the attribute value. It is similar to E[attr~=val] above, except the val can be part of a word.

Using the same example as above, .fakelink[title~=info] {} would match any element with the class fakelink that has a title attribute containing the string info, such as "Click here for more information."

6. CSS3 selectors

Pseudo-classes

:enabled
A user interface element that’s enabled.
:disabled
Conversely, a user interface element that’s disabled.
:checked
Radio buttons or checkboxes that are selected or ticked.
:valid
Applies to elements that are valid, based on the type or pattern attributes
:invalid
Applies to empty required elements, and elements failing to match the requirements defined by the type or pattern attributes.
:in-range
Applies to elements with range limitations, where the value is within those limitations. This applies, for example, to number and range input types with min and max attributes
:out-of-range
The opposite of :in-range: elements whose value is *outside* the limitations of their range.
:required
Applies to form controls that have the required attribute set.
:optional
Applies to all form controls that *do not* have the required attribute.
:read-only
Applies to elements whose contents are unable to be altered by the user. This is usually most elements other than form fields.
:read-write
Applies to elements whose contents are user-alterable, such as text input fields

6. CSS3 selectors

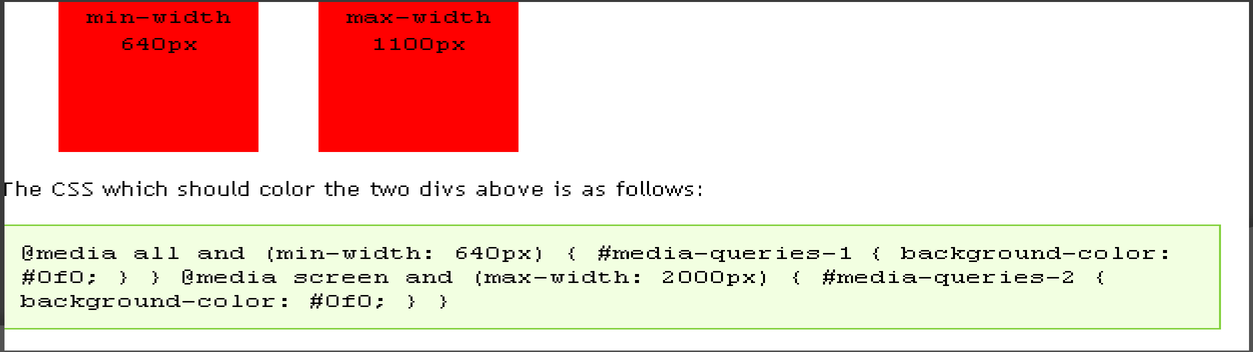
Structural Pseudo-classes

:root
The root element, which is always the html element.
E F:nth-child(n)
The element F that is the nth child of its parent E.
E F:nth-last-child(n)
The element F that is the nth child of its parent E, counting backwards from the last one. li:nth-last-child(1) would match the last item in any list—this is the same as li:last-child (see below).
E:nth-of-type(n)
The element that is the nth element of its type in a given parent element.
E:nth-last-of-type(n)
Like nth-of-type(n), except counting backwards from the last element in a parent.
E:first-child
The element E that is the first child E of its parent. This is the same as :nthchild(1).
E:last-child
The element E that is the last child E of its parent, same as :nth-last-child(1).
E:first-of-type
Same as :nth-of-type(1).
E:last-of-type
Same as :nth-last-of-type(1).
E:only-child
An element that's the only child of its parent.
E:only-of-type
An element that's the only one of its type inside its parent element.
E:empty
An element that has no children; this includes text nodes, so <p>hello</p> will not be matched.
E:lang(en)
An element in the language denoted by the two-letter abbreviation (en).
E:not(exception)
This is a particularly useful one: it will select elements that *don't* match the selector in the parentheses.

7. CSS3 media queries

CSS2 added support for the media="screen" way of defining which stylesheet to use for which representation of the data. CSS3 adds a new feature to this functionality, by adding media queries. Basically, this means you can change stylesheets based on for instance the width and height of the viewport. In a broader sense, this means as the spec puts it: *"by using Media Queries, presentations can be tailored to a specific range of output devices without changing the content itself."*

Below are two tests, for min-width and max-width, currently only functional (and thus green) in Safari 3, Opera, and Firefox 3.1 (Alpha). This *is* however the future of web development, and could make building pages that are usable in both the mobile as the normal web a lot easier.



The CSS which should color the two divs above is as follows:

```
@media all and (min-width: 640px) { #media-queries-1 { background-color:
#0f0; } } @media screen and (max-width: 2000px) { #media-queries-2 {
background-color: #0f0; } }
```

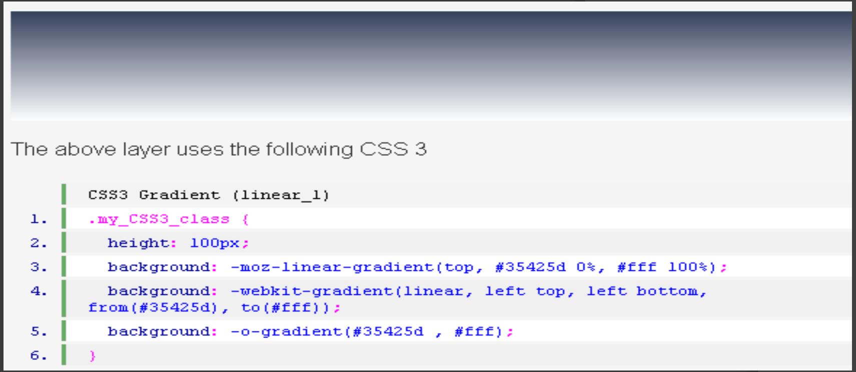
7. CSS3 Gradients

Use CSS 3 to create some nice subtle gradients; or very compelling gradients! CSS 3 Gradients are not
the simplest things by any means; there are several values to account for when projecting your
gradient and it only worsens when working with radial gradients.

background:-moz-linear-gradient(pos, #AAA B, #XXX Y);
or :-moz-radial-gradient(pos, #AAA B, #XXX Y);

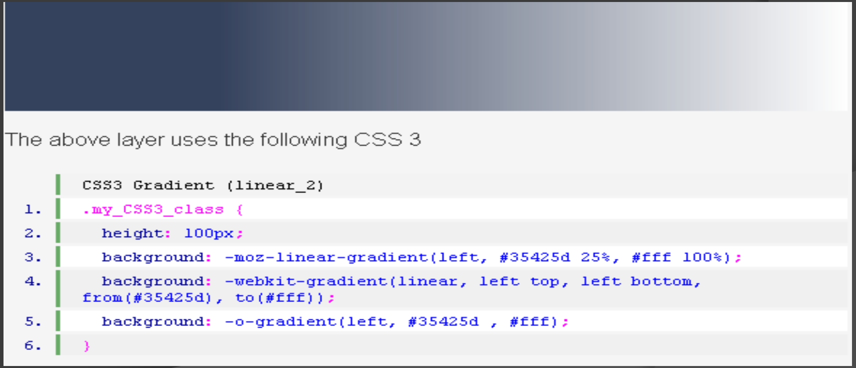
pos = the position of the first colour, giving direction to the gradient
#AAA = primary colour
B = where the fade begins (%)
#XXX = secondary colour
Y = where the fade begins (%)

The -webkit-gradient() is built slightly differently but works in the same manner, creating a start
position and then the colors from() and to();



The above layer uses the following CSS 3

```
CSS3 Gradient (linear_1)
.my_CSS3_class {
  height: 100px;
  background: -moz-linear-gradient(top, #35425d 0%, #fff 100%);
  background: -webkit-gradient(linear, left top, left bottom,
from(#35425d), to(#fff));
  background: -o-gradient(#35425d , #fff);
}
```



The above layer uses the following CSS 3

```
CSS3 Gradient (linear_2)
.my_CSS3_class {
  height: 100px;
  background: -moz-linear-gradient(left, #35425d 25%, #fff 100%);
  background: -webkit-gradient(linear, left top, left bottom,
from(#35425d), to(#fff));
  background: -o-gradient(left, #35425d , #fff);
}
```

Belajar Lebih Lanjut?

- http://www.w3schools.com/css/default.asp
- Google
- Youtube