

# **CS 50B Web Development 2**

Ron Liskey, Instructor

February 20, 2016

---

# Contents

<b>I. Lessons</b>	<b>5</b>
<b>Backgrounds</b>	<b>6</b>
Assignment: Border Radius . . . . .	7
Overview . . . . .	7
Step 1: Create the HTML file . . . . .	7
Step 2: Add a circle . . . . .	8
Step 3: Create skewed curves . . . . .	9
Step 4: Create an element with for different corners . . . . .	9
Step 5: Create a tab . . . . .	10
Step 6: Experiment and improve . . . . .	11
Assignment: Box Shadow . . . . .	12
Overview . . . . .	12
Step 1: Create HTML file . . . . .	13
Step 2: Add an element with a blurred shadow . . . . .	14
Step 3: Add an element with a transparent blurred shadow . . . . .	15
Step 4: Create a multiple transparent shadow . . . . .	15
Step 5: Create a multiple transparent shadow with a border radius . . . . .	16
Step 6: Create an embossed shadow . . . . .	17
Step 7: Create an engraved shadow . . . . .	18
Step 8: Experiment and improve . . . . .	19
Assignment: Gradient Fills . . . . .	20
Overview . . . . .	20
Step 1: Create HTML file . . . . .	20
Step 2: Add the top to bottom gradient . . . . .	21
Step 3: Add a linear-left gradient . . . . .	22
Step 4: Add a multi-color gradient . . . . .	22
Step 5: Add a radial gradient . . . . .	23
Step 5: Add an offset radial gradient . . . . .	24
Step 6: Experiment and improve . . . . .	24
Assignment: Multiple Background Layers . . . . .	25
Overview . . . . .	25
Step 1: Create the HTML page . . . . .	25

---

Step 2: Add a second background image . . . . .	26
Step 3: Add a gradient . . . . .	26
Step 4: Add a box-shadow . . . . .	27
Step 5: Add border radius . . . . .	27
Step 6: Modify the text . . . . .	27
Step 7: Rotate the title . . . . .	28
Step 8: Add a paragraph . . . . .	28
Step 9: Add a Google font . . . . .	28
Step 10: Ensure background support in all major browsers . . . . .	29
Step 11: Ensure rotation support in all major browsers . . . . .	30
Step 12: Experiment and improve . . . . .	30
Assignment: Image Gallery Thumbnails . . . . .	31
Overview . . . . .	31
Step 1: Create your images . . . . .	31
Step 2: Create the HTML page . . . . .	31
Step 3: Add code for a 3 by 3 grid of rows . . . . .	31
Step 4: Add three image links within each row . . . . .	32
Step 5: Add CSS styles . . . . .	33
Step 6: Create the 6 in a row gallery . . . . .	34
Step 7: Add six thumbnail images . . . . .	34
Step 8: Experiment and fine tune . . . . .	34

---

# List of Figures

0.0.1A transparent blurred shadow . . . . . 15

---

**Part I.**

**Lessons**

---

# Backgrounds

---

# Assignment: Border Radius

## Overview

In this assignment, your task is to make a sampler that demonstrates the **border-radius** property. Before CSS3, creating a block of color with rounded edges would require images placed in the background of nested **<div>** tags. This is now much easier using CSS3. In this sampler you will produce the following effects on **<div>** tags:

1. Identical rounded radius on all four corners
2. A perfect circle
3. A skewed curve
4. A different radius on each corner
5. The tabbed look

## Step 1: Create the HTML file

1. In your samples directory, create a new HTML file named: **border-radius-sampler.html**
2. Add the standard **<html>**, **<head>**, **<body>** and **<style>** sections to this file.
3. In the **<style>** section, add definitions for **<h1>**, **<p>**, and **#rounded**.
4. In the **<body>** section, add an **<h1>** tag to identify the page and a **<div>** tag with an id of **#rounded**. Your file should look similar to the following example.

```
1 <!DOCTYPE html >
2 <html lang="en" >
3 <head >
4 <title>Border Radius Sampler</title >
5 <style >
6 h1 { text-align: center; }
7 p { text-align:center; color: #fff; }
8 #rounded {
9     background: #0099cc;
10    height: 200px;
11    width: 400px;
12    margin: 10px auto;
13    padding: 10px;
14    border-radius: 50px;
15 }
16 </style >
17 </head >
```

```
18 <body>
19 <h1>Border Radius Sampler</h1>
20 <div id="rounded">
21 <p>A simple rounded rectangle</p>
22 </div>
23 </body>
24 </html>
```

## Step 2: Add a circle

1. In the body of your file, add a **<div>** tag with an **id** of **circle**.
2. In the **<style>** section, add a matching CSS definition.

The styling for this id, is similar to the previous example, however it has been assigned identical values of **200px** for the **height** and width. This produces a perfect square, which is the perfect container for a perfect circle.

**Note:** A border-radius of **100px** (half the total height or width) does not produce a perfect circle. We must also take the padding of **10px** into account. The correct equation for calculating the border-radius in this case is:

$$(width + paddingFirstSide + paddingSecondSide)/2 = (200px + 10px + 10px)/2 = 110px$$

## Example HTML code

```
1 <div id="circle">
2   <p>A perfect circle!</p>
3 </div>
```

## Example CSS code

```
1 #circle {
2   background: #0099cc;
3   height: 200px;
4   width: 200px;
5   margin: 10px auto;
6   padding: 10px;
7   border-radius: 110px;
8 }
```

---

### Step 3: Create skewed curves

1. In the `<body>` section of your file, create a new `<div>` with an id of `#skewed`.
2. In the `<style>` section of your file, create a CSS definition for `#skewed`. Include two values for `border-radius` separated by a slash. The first value is applied to the x-axis (vertical axis) and the second value is applied to the y-axis (horizontal axis).

#### Example HTML code

```
1 <div id="skewed">
2   <p>A skewed shape!</p>
3 </div>
```

#### Example CSS code

```
1 #skewed {
2   background:#0099cc;
3   height:200px;
4   width:400px;
5   margin:10px auto;
6   padding:10px;
7   border-radius:200px / 50px;
8 }
```

### Step 4: Create an element with for different corners

1. In the `<body>` section of your file, create a new `<div>` with an id of `#four-unique`.
2. In the `<style>` section of your file, create a CSS definition for `#four-unique`. Include four unique values for `border-radius`, each separated by a space. The following example applies a `border-radius` of `10px` to the top left corner, `50px` to top right corner, `100px` to bottom right corner and `200px` to bottom left corner.

---

### Example HTML code

```
1 <div id="four-unique">
2   <p>Four unique corners!</p>
3 </div>
```

### Example CSS code

```
1 #four-unique {
2   background:#0099cc;
3   height:200px;
4   width:400px; margin:10px auto;
5   padding:10px;
6   border-radius:10px 50px 100px 200px;
7 }
```

### Step 5: Create a tab

Tabs are created in CSS by assigning a **border-radius** to the top left and right corners, and leaving the bottom corners unchanged. In the following example, a border-radius of **50px** is applied only to the top left and top right corners.

1. In the **<body>** section of your file, create a new **<div>** with an **id** of **#tab**.
2. In the **<style>** section of your file, create a CSS definition for **#tab**. Include appropriate values for **border-radius**. The following example applies a border-radius of **15px** to the top left and right corners.

### Example HTML code

```
1 <div id="tab">
2   <p>the tab look</p>
3 </div>
```

---

## Example CSS code

```
1 #tab {  
2     background:#0099cc;  
3     height:100px;  
4     width:400px;  
5     margin:10px auto;  
6     border-radius:15px 15px 0px 0px;  
7     padding:10px;  
8 }
```

## Step 6: Experiment and improve

Be creative. Seek originality. Use these examples as starting points. Modify each element in some way. Add new elements of your own creation. Combine these techniques with other CSS tricks.

---

## Assignment: Box Shadow

### Overview

In this assignment you will use the **box-shadow** property to create various box shadow effects. The **box-shadow** property defines one or more shadows as a comma-separated list. It can be applied to most HTML elements. If a **border-radius** property is also defined on an element, the corners of the box shadow will conform to the **border-radius** settings. The **z-order** of multiple box shadows is in order of their definition, with the first defined shadow on top.

See the W3C Specification for complete details:

<https://www.w3.org/TR/css3-background/#the-box-shadow>

### Border Radius Options

**inset** The inset keyword displays a shadow within the element's frame, producing an engraved look. If inset not specified a default drop shadow is displayed.

**<offset-x> <offset-y>** Controls the shadow offset. **<offset-x>** sets the horizontal distance. Negative values set the shadow to the left of the element. **<offset-y>** sets the vertical distance. Negative values place the shadow above the element. If both values are set to 0, the shadow is placed behind the element, and can be used to create a blur effect if **<blur-radius>** and/or **<spread-radius>** are used.

**<blur-radius>** The larger this value, the bigger the blur, so the shadow becomes bigger and lighter. Negative values are not allowed. If not specified, it will be 0 (the shadow's edge is sharp).

**<spread-radius>** Positive values cause the shadow to expand and grow. Negative values cause the shadow to shrink. If not specified, the spread-radius will be 0, and the shadow will be the same size as the element.

**<color>** If not specified, the browser's default color is used.

### Example Syntax

```
1 /* offset-x | offset-y | color */
2 .teal-shadow{box-shadow:60px -16px teal;}
3 /* offset-x | offset-y | blur-radius | color */
4 .black-shadow{box-shadow:10px 5px 5px black;}
5 /* offset-x | offset-y | blur-radius | spread-radius | color */
6 .blury-shadow{2px 2px 2px 1px rgba(0, 0, 0, 0.2);}
7 /* inset | offset-x | offset-y | color */
8 .inset-shadow{box-shadow:inset 5em 1em blue;}
```

```
9 /* Multiple shadows (separated by commas) */
10 .multi-shadow{box-shadow:3px 3px green, -1em 0 0.4em gold;}
```

## Step 1: Create HTML file

Create an HTML file with the following content:

```
1 <html>
2   <head>
3     <title>Box Shadow Sampler</title>
4   </head>
5   <body>
6   </body>
7 </html>
```

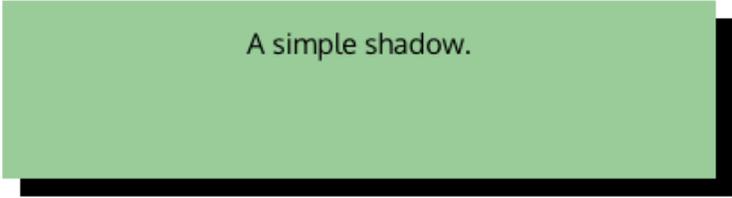
Within the **<head>** section add the following CSS:

```
1 <style>
2   p {text-align:center; color:black; text-weight:bold;}
3   h1 {text-align:center;}
4   #shadow {
5     background:#99cc99;
6     height:200px;
7     width:400px;
8     /* horizontal offset, vertical ofset, color */
9     box-shadow:10px 10px #000;
10    /* vertical margins are 10px, horizontal margins are auto */
11    margin:10px auto;
12    padding:10px;
13  }
14 </style>
```

Within the **<body>** section add the following HTML code.

```
1 <h1>Box Shadow Sampler</h1>
2 <div id="shadow">
3   <p>A simple shadow.</p>
4 </div>
```

---

A simple shadow. A light green rectangular box with a solid black drop shadow to its right and bottom.

## Step 2: Add an element with a blurred shadow

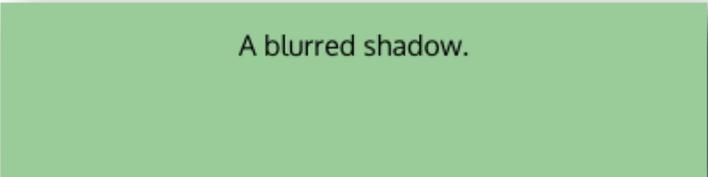
The style in this example is similar to the previous one, with the addition of a blur value.

### HTML

```
1 <div id="shadow-blur">
2   <p>A blurred shadow.</p>
3 </div>
```

### CSS

```
1 #shadow-blur {
2   background:#99cc99;
3   height:200px;
4   width:400px;
5   box-shadow:10px 10px 30px #000; /* blur value = 30px */
6   margin:50px auto;
7   padding:10px;
8 }
```

A blurred shadow. A light green rectangular box with a soft, blurred black drop shadow to its right and bottom.

---

### Step 3: Add an element with a transparent blurred shadow

The key concept in this example, is the color of the shadow. In the **rgba** color model the **a (alpha channel)** adds transparency. Transparency values can range from .1 to 1.0, with smaller values producing more transparency and large values producing more opacity.

#### HTML

```
1 <div id="shadow-transparent">
2   <p>A blurred shadow with a transparent color.</p>
3 </div>
```

#### CSS

```
1 #shadow-transparent {
2   background:#99cc99;
3   height:200px;
4   width:400px;
5   box-shadow:10px 10px 30px rgba(0, 0, 0, .5); /* transparency value = .5
6   */
7   margin:50px auto;
8   padding:10px;
9 }
```

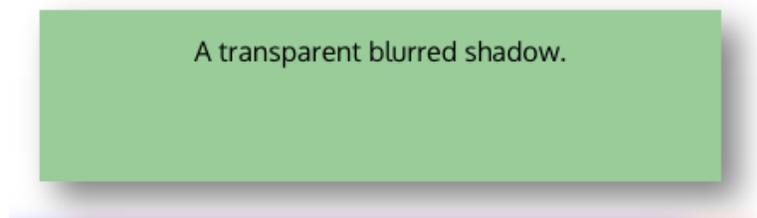


Figure 0.0.1.: A transparent blurred shadow

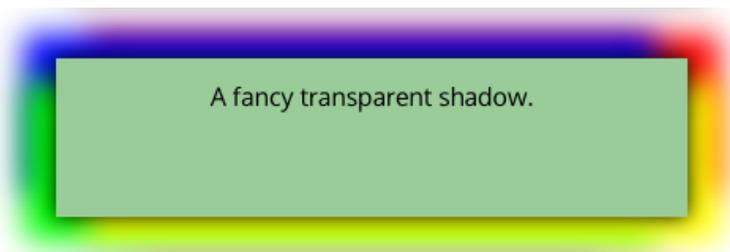
### Step 4: Create a multiple transparent shadow

#### HTML

```
1 <div id="multi-transparent-shadow">
2   <p>A fancy transparent shadow.</p>
3 </div>
```

## CSS

```
1 #multi-transparent-shadow {
2   height:100px;
3   width:400px;
4   margin:50px auto;
5   padding:10px;
6   background:#99cc99;
7   box-shadow: 0 0 20px black, /* centered */
8               20px 15px 30px yellow, /* lower right */
9               -20px 15px 30px lime, /* lower left */
10              -20px -15px 30px blue, /* upper left */
11              20px -15px 30px red; /* upper right */
12 }
```



## Step 5: Create a multiple transparent shadow with a border radius

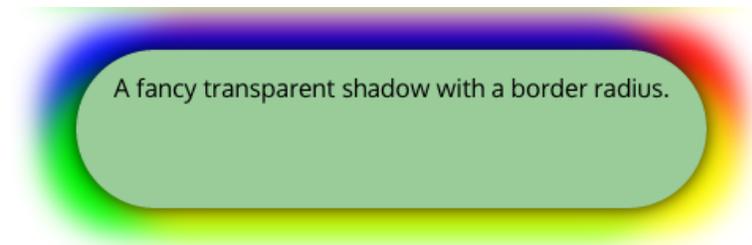
### HTML

```
1 <div id="multi-transparent-border-radius-shadow">
2   <p>A fancy transparent shadow with a border radius.</p>
3 </div>
```

---

## CSS

```
1 #multi-transparent-border-radius-shadow {
2   height:100px;
3   width:400px;
4   margin:50px auto;
5   padding:10px;
6   background:#99cc99;
7   box-shadow: 0 0 20px black, /* centered */
8     20px 15px 30px yellow, /* lower right */
9     -20px 15px 30px lime, /* lower left */
10    -20px -15px 30px blue, /* upper left */
11     20px -15px 30px red; /* upper right */
12   border-radius: 60px;
13 }
```



## Step 6: Create an embossed shadow

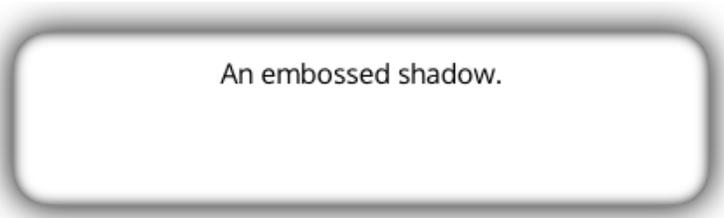
### HTML

```
1 <div id="embossed-shadow">
2   <p>Your comments here.</p>
3 </div>
```

### CSS

```
1 #embossed-shadow {
2   background:#fff;
3   height:100px;
4   width:400px;
5   margin:50px auto;
6   padding:10px;
```

```
7 box-shadow: 0 0 30px #000000, inset 0 0 10px #000000;
8 border-radius: 20px;
9 }
```



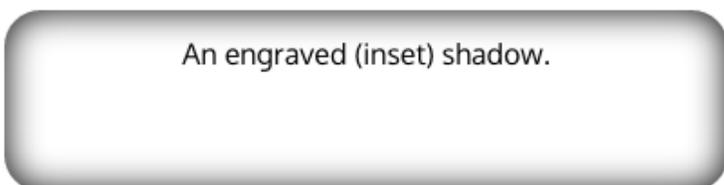
## Step 7: Create an engraved shadow

### HTML

```
1 <div id="engraved-shadow">
2   <p>Your comments here.</p>
3 </div>
```

### CSS

```
1 #engraved-shadow {
2   background:#fff;
3   height:100px;
4   width:400px;
5   margin:50px auto;
6   padding:10px;
7   box-shadow: inset 0 0 30px #000000;
8   border-radius: 20px;
9 }
```



---

## **Step 8: Experiment and improve**

Be creative. Seek originality. Use these examples as starting points. Modify each element in some way. Add new elements of your own creation. Add an image to your page and apply the shadow of your choice to the image element. Combine box shadow techniques with other CSS tricks you know.

---

## Assignment: Gradient Fills

### Overview

In this assignment you will create various gradient fill effects. The **gradient** property was added to the specification in CSS3. Before CSS3, Web developers used images to produce gradients. In this assignment you will create the various gradient fills using pure CSS3.

Because the standard for this property has not been finalized we use browser specific prefixes to ensure that all major browsers understand how to correctly display the effect. Once the standard is finalized we will be able to simply use the **gradient** property. Examples of unique properties for all major browsers follows. Note that browser-specific properties always begin with a dash (-) character. This list ends with the generic **gradient** version to catch all CSS3 enabled browsers. Such browsers ignore the browser-specific definitions.

### Linear gradient examples

```
1 -moz-linear-gradient /* For Firefox */
2 -webkit-gradient /* For newer versions of Safari and Chrome */
3 -webkit-linear-gradient /* For newer versions of Safari & Chrome */
4 -ms-linear-gradient /* For IE 10 */
5 -o-linear-gradient /* For Opera */
6 gradient /* For all future browsers */
```

### Step 1: Create HTML file

1. In your samples directory, create a file named: **gradient.html**
2. Add the following HTML to your file.

### HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Gradient Sampler</title>
5   </head>
6   <body>
7     <h1>Gradient Sampler</h1>
```

```
8 </body>
9 </html>
```

## Step 2: Add the top to bottom gradient

1. Add the following HTML and CSS to your file.
2. After adding a **linear-gradient** value, add arguments within the parenthesis for each property. The value **top** indicates that the first color will be displayed at the top of the element. In this example, the value **#f9f** sets the first color, and the value **#000** sets the second color. Four of the five values follow this pattern, the values for older versions of Safari and Chrome are more complex, however this standard will be abandoned in the final version of CSS3.

### HTML

```
1 <div id="linear-top"></div>
```

### CSS

```
1 #linear-top{
2   width:500px;
3   height:100px;
4   margin: 10px auto;
5   padding:10px;
6   background: -moz-linear-gradient(top, #f9f, #000); /* Firefox 3.6+ */
7   background: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#000), to(#f9f
8     )); /* Safari 4-5, Chrome 1-9 */
9   background: -webkit-linear-gradient(top, #f9f, #000); /* Safari 5.1,
10     Chrome 10+ */
11   background: -ms-linear-gradient(top, #f9f, #000); /* IE 10 */
12   background: -o-linear-gradient(top, #f9f, #000); } /* Opera 11.10+ */
13   linear-gradient(top, #f9f, #000); } /* All future browsers */
```

---

### Step 3: Add a linear-left gradient

Add the following code to your file. Note the use of the **left** keyword.

#### HTML

```
1 <div id="linear-left"></div>
```

#### CSS

```
1 #linear-left{  
2   width:500px;  
3   height:100px;  
4   margin: 10px auto;  
5   padding:10px;  
6   background: -moz-linear-gradient(left, #f9f, #000);  
7   background: -webkit-gradient(linear ,left top, right top, from(#f9f), to  
8     (#000));  
9   background: -webkit-linear-gradient(left, #f9f, #000);  
10  background: -ms-linear-gradient(left, #f9f, #000);  
11  background: -o-linear-gradient(left, #f9f, #000);  
12  linear-gradient(left, #f9f, #000);  
}
```

### Step 4: Add a multi-color gradient

Add the following code to your file.

#### HTML

```
1 <div id="linear-pos"></div>
```

---

## CSS

```
1 #linear-pos {
2   width:500px;
3   height:100px;
4   margin: 10px auto;
5   padding:10px;
6   background: -moz-linear-gradient(left, #f9f, #000, #f9f, #000, #f9f);
7   background: -webkit-gradient(linear, left top, right top, from(#f9f),
8     color-stop(0.25, #000), color-stop(0.5, #f9f), color-stop(0.75,
9     #000), to(#f9f));
10  background: -webkit-linear-gradient(left, #f9f, #000, #f9f, #000, #f9f);
11  background: -ms-linear-gradient (left, #f9f, #000, #f9f, #000, #f9f);
12  background: -o-linear-gradient (left, #f9f, #000, #f9f, #000, #f9f);
13  linear-gradient (left, #f9f, #000, #f9f, #000, #f9f);
14 }
```

## Step 5: Add a radial gradient

### HTML

```
1 <div id="radial"></div>
```

### CSS

```
1 #radial {
2   width:300px;
3   height:300px;
4   margin: 10px auto;
5   padding:10px;
6   background: -moz-radial-gradient(circle, #f9f, #000);
7   background: -webkit-gradient(radial, center center, 0, center center,
8     460, from(#9f9), to(#000));
9   background: -webkit-radial-gradient(circle, #f9f, #000);
10  background: -ms-radial-gradient(circle, #f9f, #000);
11  background: -o-radial-gradient(circle, #f9f, #000);
12  radial-gradient(circle, #f9f, #000);
13 }
```

---

## Step 5: Add an offset radial gradient

The default origin for a radial gradient is the center of the element. We can adjust this by setting a unique position for the origin and end point. In this example we set the origin at 80% from the left edge, and 205px from the top edge of the element. We set the end point at the **closest-corner**.

### HTML

```
1 <div id="radial-pos"></div>
```

### CSS

```
1 #radial-pos {
2   width:300px;
3   height:300px;
4   margin: 10px auto;
5   padding:10px;
6   background: -moz-radial-gradient(80% 20%, closest-corner, #f9f, #000);
7   background: -webkit-gradient(radial, 80% 20%, 0, 80% 40%, 100, from(#f9f
8   ), to(#000));
9   background: -webkit-radial-gradient(80% 20%, closest-corner, #f9f, #000)
10  ;
11  background: -ms-radial-gradient(80% 20%, closest-corner, #f9f, #000);
12  background: -o-radial-gradient(80% 20%, closest-corner, #f9f, #000);
13  radial-gradient(80% 20%, closest-corner, #f9f, #000);
14 }
```

## Step 6: Experiment and improve

Be creative. Seek originality. Use these examples as starting points. Modify each element in some way. Add new elements of your own creation. Combine these techniques with other CSS tricks.

---

# Assignment: Multiple Background Layers

## Overview

In this assignment you will combine your understanding of most of the topics covered in the background lesson. The project makes use of **border-radius**, box-shadow, multiple background layers, **gradient**, rotated text, and a font pulled from the Google API.

## Step 1: Create the HTML page

1. Create a file called **background-project.html**, and save it in your **samples** directory.
2. Add the following code to this file.
3. The #wrapper tag uses a background image. Add an image of your choice to your project files, and update the URL path of the #wrapper tag background property as appropriate.

## HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Background Project</title>
5   </head>
6   <body>
7     <div id="wrapper">
8       <h1>Techie Gardener</h1>
9       <div id="description">
10        <p>This page demonstrates layered backgrounds.</p>
11      </div>
12    </div>
13  </body>
14 </html>
```

## CSS

```
1 <style>
2 h1 {font-size:2.5em; line-height:1em;}
```

```
3 #description {font-size:1.2em;}
4
5 #wrapper {
6   width:700px;
7   height:500px;
8   margin: 10px auto;
9   padding:10px;
10  background:url(path/to/your/image.png) 250px 100px no-repeat;
11 }
12 </style>
```

## Step 2: Add a second background image

1. Multiple background image declarations are separated by commas. Place a comma after the first background declaration and add another declaration. In this example, the image will be placed in the top left corner of the **#wrapper <div>**, and it will not repeat.

### CSS

```
1 background:url(path/to/your/image.png) 250px 100px no-repeat, url(path/to/
   your/image2.png) top left no-repeat;
```

At this point, if you view the page in a browser, you should see one image on top of the other. The background images, colors or gradients are stacked in the order listed, the first image will be on top, the last element will be on the bottom.

## Step 3: Add a gradient

1. Add code for a gradient, similar to the following example. Remember that many browsers require unique gradient properties. Add the correct variants for all major browsers.

### CSS

```
1 background:url(path/to/your/image.png) 250px 100px no-repeat, url(path/to/
   your/image2.png) top left no-repeat,
```

---

```
2 linear-gradient(top, #fff, #CCC); /* Remember to add browser-specific
   versions */
```

#### Step 4: Add a box-shadow

1. Add a box-shadow property similar to the following example to the **#wrapper** style.

#### CSS

```
1 box-shadow:0px 0px 10px rgba(0, 0, 0, 0.7);
```

#### Step 5: Add border radius

1. Add code to create rounded corners, such as in the following example.

#### CSS

```
1 border-radius:10px;
```

#### Step 6: Modify the text

1. Change the placement of the text and modify the color. In the following example, we modify the **<h1>** tag to display as white text, with a top margin of 125px, right and bottom margins of zero pixels, and a left margin of 50px. We also set a width of 200px.

#### CSS

```
1 h1{
2   font-size: 2.5em;
3   line-height: 1em;
4   color: #fff;
5   margin: 125px 0 0 50px;
```

---

```
6 width: 200px;
7 }
```

### Step 7: Rotate the title

Rotation is not fully supported in all browsers, therefore use browser-specific prefixes for all major browsers. In the example below. The property is **transform**, the value is **rotate**, and the argument is **-20deg**, which rotates the text 20 degrees counterclockwise.

#### CSS

```
1 -moz-transform: rotate(-15deg); /* Firefox */
2 transform: rotate(-20deg); /* All */
```

### Step 8: Add a paragraph

Adjust the style of **#description**. In the below example, the width, line-height, margins, and text color have been altered.

#### CSS

```
1 #description{
2   font-size:1.2em;
3   width:200px;
4   line-height:1.5em;
5   margin:70px 0 0 20px;
6   color:#777;
7 }
```

### Step 9: Add a Google font

1. Select a font on the Google Fonts website: <http://www.google.com/webfonts#ChoosePlace:sel>
2. Use HTML **<link>** code similar to the following to pull in the font. Be sure to replace FONTNAME-HERE with your actual font.

- 
3. Update your CSS as appropriate. See below for one possible example. Here too, be sure to replace FONTNAME-HERE with your actual font.

## HTML

```
1 <link href='http://fonts.googleapis.com/css?family=FONTNAME-HERE' rel='
  stylesheet' type='text/css'>
```

## CSS

```
1 h1{
2   font-size:2.5em;
3   line-height:1em;
4   color:#fff;
5   margin:125px 0 0 50px;
6   width:200px;
7   font-family:'FONTNAME-HERE', cursive;
8 }
```

## Step 10: Ensure background support in all major browsers

1. Experiment with the following examples and test your code in as many browser types as possible.

## CSS

```
1 -webkit-box-shadow:0px 0px 10px rgba(0, 0, 0, 0.7);
2 /* Firefox */
3 background:url(path/to/your/image.png) 250px 100px no-repeat,
4   url(path/to/your/image2.png) top left no-repeat,
5 -moz-linear-gradient(top, #fff, #ccc);
6 /* Safari and Chrome */
7 background:url(path/to/your/image.png) 250px 100px no-repeat,
8   url(path/to/your/image2.png) top left no-repeat,
9 -webkit-gradient(linear, left top, left bottom, from(#fff), to(#ccc));
10 /* IE 9 */
11 background:url(path/to/your/image.png) 250px 100px no-repeat,
12   url(path/to/your/image2.png) top left no-repeat,
```

---

```
13 filter: progid:DXImageTransform.Microsoft.Gradient(  
14   StartColorStr='#fff', EndColorStr='#ccc', GradientType=0);  
15 /* Opera */  
16 background:url(path/to/your/image.png) 250px 100px no-repeat,  
17   url(path/to/your/image2.png) top left no-repeat,  
18   -o-gradient(linear, left top, left bottom, from(#fff), to(#ccc));}
```

## Step 11: Ensure rotation support in all major browsers

1. The following properties may be added to the <h1> style to ensure that it rotates in all browsers.

### CSS

```
1 -moz-transform: rotate(-20deg); /* Firefox */  
2 -webkit-transform: rotate(-20deg); /* Safari and Chrome */  
3 -ms-transform: rotate(-20deg); /* IE 9 */  
4 -o-transform: rotate(-20deg); /* Opera */
```

## Step 12: Experiment and improve

Be creative. Seek originality. Use these examples as starting points. Add new elements of your own creation. Combine these techniques with other CSS tricks you know.

---

## Assignment: Image Gallery Thumbnails

### Overview

In this assignment you will create two sets of thumbnail images that link to larger images. One gallery will be a  $3 \times 3$  grid of thumbnails. The other will be a row of six images. Both galleries will use CSS for user interaction and Bootstrap styles for viewport responsiveness.

### Step 1: Create your images

1. Create nine images each with a width of 600px. Use a logical file naming convention, such as **image1.png**, **image2.png**, **image3.png**, etc.
2. Create thumbnail versions of each image that are each 100px wide. Use a logical file naming convention, such as adding **-tn** to each file name. For example, **image1-tn.png**, **image2-tn.png**, **image3-tn.png**, etc.
3. Move all the images into a directory near your **thumb-gallery.html** file.

### Step 2: Create the HTML page

1. Create a file in your **samples** directory named: **thumb-gallery.html**, and enter the following code.

#### HTML

```
1 <html>
2   <title>Image Gallery Thumbnails</title>
3   <style>
4   </style>
5   </head>
6   <body>
7   </body>
8 </html>
```

### Step 3: Add code for a 3 by 3 grid of rows

1. Add the following code to the **<body>** section of your file.

---

## HTML

```
1 <div class="container">
2 <div class="wrapper">
3   <h2>3 &times; 3 grid</h2>
4   <div class="row"> <!-- row 1 -->
5     <!-- A set of three images will soon go here. -->
6   </div><!-- /row 1 -->
7   <div class="row"> <!-- row 2 -->
8     <!-- A set of three images will soon go here. -->
9   </div><!-- /row 2 -->
10  <div class="row"> <!-- row 3 -->
11    <!-- A set of three images will soon go here. -->
12  </div><!-- /row 3 -->
13 </div><!-- /wrapper -->
14 </div><!-- /container -->
```

### Step 4: Add three image links within each row

1. Add the following code within each of the three rows created above. At the same time, you can delete the comments within the rows.
2. Modify the image paths and names as needed to correspond to your directory structure.
3. Note that image file names will increment from 1 to 9. Do not repeat the file names for images 1 through 3 within all three rows.
4. Note the Bootstrap style, **col-sm-4**. This ensures that the gallery works well within all viewports.

## HTML

```
1 <div class="col-sm-4">
2   <a href="/path/to/image1.png" target="_blank" class="thumbnail">
3     
4   </a>
5 </div>
6
7 <div class="col-sm-4">
8   <a href="/path/to/image2.png" target="_blank" class="thumbnail">
9     
10  </a>
```

```
11 </div>
12
13 <div class="col-sm-4"><a href="/path/to/image3.png" target="_blank" class=
    "thumbnail">
14     
15 </a>
16 </div>
```

## Step 5: Add CSS styles

1. Add the following CSS styles within the <style> section.
2. Once you have completed this step, you should be able to view the 9 by 9 thumbnail gallery. View it now to check your work.

```
1 .wrapper {
2   background: #000;
3   margin-top: 20px;
4   padding: 25px;
5   color: #fff;
6   border-radius: 10px;
7 }
8
9 .wrapper h2 {
10  text-align: center;
11 }
12
13 .thumbnail {
14  text-align: center;
15  margin-bottom: 25px;
16  background: #000;
17  border: 1px solid #333;
18 }
19
20 .thumbnail:hover {
21  border: 1px solid teal;
22  background: #333;
23 }
24
25 .thumbnail img {
26  background: #fff;
27 }
```

---

## Step 6: Create the 6 in a row gallery

1. Below the previously entered closing `</wrapper>` tag, and above the closing `</container>` tag, add a new `<wrapper>` section containing a single `<div class="row">`.
2. Also enter an `<h2>` title.

```
1 <div class="wrapper">
2 <h2>6 in a row</h2>
3 <div class="row">
4
5 </div><!-- /row -->
6 </div><!-- /wrapper -->
```

## Step 7: Add six thumbnail images

1. Within this new `<div class="row">` section enter the following code six times.
2. In each copy, update the file paths and image file names as needed for your setup.
3. Once you have completed this step, you should be able to view the 6 in a row thumbnail gallery. View it now to check your work.

```
1 <div class="col-sm-2">
2   <a href="/path/to/image1.png" target="_blank" class="thumbnail">
3     
4   </a>
5 </div>
```

## Step 8: Experiment and fine tune

Be creative. Seek originality. Use these examples as starting points. Add new elements of your own creation. Combine these techniques with other CSS tricks you know.

---

# Index

## A

alpha channel, [15](#)

## B

Bootstrap, [32](#)

border-radius, [7](#)

## I

image gallery

    thumbnail, [31](#)

## L

linear-gradient, [21](#)

## R

radial gradient, [24](#)

## T

transform, [28](#)

    rotate, [28](#)

transparency, [15](#)