

Dreamweaver® **Introduction**

Dreamweaver Tools.	1
Panel Tip	
Application Bar	2
Layout Menu	
Building & Applying External CSS in Dreamweaver	9
Building a Site	10
Creating a Simple Page	11
Creating a CSS for Anchor Tags.	12
Cascading Style Sheets (CSS)	13
Three Methods for Applying CSS	14
Inline	
Embedded	
External	
CSS Vocabulary	16
Types of CSS Selectors	17
Element	
ID	
Class	
3-Digit Hexadecimal Values.	19
Resource Web Sites.	19
HTML5	20
The Need for New HTML Elements	
HTML5 Resources.	21
Browser Testing of HTML	21
Adobe BrowserLab for Desktop	
Adobe Shadow for Devices	
CSS3	22
Rounded CSS Corners	
Rounded Corners for Multiple Browsers	23
CSS Border-radius Prefix	
Rounded Corners for IE9	



©2012—Kelly McCathran

The information in this handout is furnished for instructional use and is subject to change without notice. No part of this handout may be reproduced or transmitted in any form or by any means, electronic or mechanical (including photocopying) without the expressed written consent of Kelly McCathran.

The software described in this handout is furnished under license and may only be used or copied in accordance with the terms of such license. Creative Suite 5 is a trademark of Adobe Systems Incorporated (“Adobe”), registered in the United States and elsewhere. InDesign, Photoshop, Illustrator, Acrobat & Bridge are trademarks of Adobe Systems Incorporated.

The value of each printed copy of this material is \$20 retail. Reprint rights may be purchased at a discount, depending on the quantity.

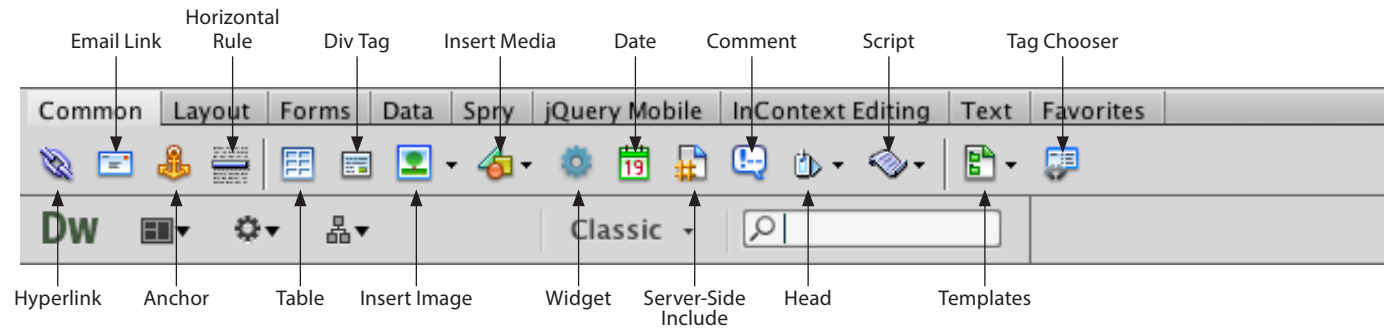
Helpful Links

<http://tv.adobe.com> - Thousands of free videos covering new features, intro topics and workflow in the Creative Suite

<http://creativesuitelovers.com> - Kelly’s ®Adobe User Group

Kelly McCathran
San Francisco, CA
(415) 312-2459
kelly@onlineinstruct.com

Dreamweaver Tools

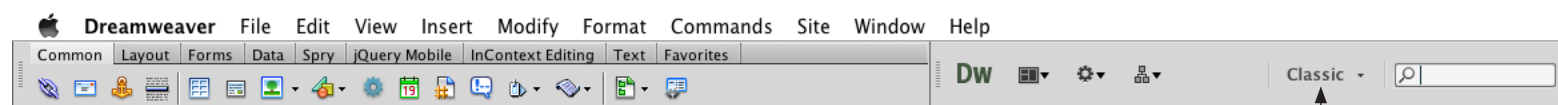


Panel Tip

Hide all Panels (**F4**)

Application Bar

All versions of the Creative Suite now include an Application Bar, this may be in a different location, depending on your platform **Mac** or **Windows**.



Classic

The first thing I do, before I begin working in Dreamweaver, is set the workspace to Classic. This gives me my Insert panel with the Common tab and other buttons that I use frequently.

Layout Menu

Most Creative Suite programs include the **Application bar** across the top of the screen. One of the nicest features of this bar is the **Layout menu** (*2nd button, next to the DW icon*) where you can easily switch from Code, Split Code, Design or Code & Design views. Where you see the word **Classic** is the new Workspace switcher.



Dreamweaver Over the Years

Over the years Dreamweaver has made many improvements that make building web pages and web sites easier and more efficient. Gone are the days when web pages consisted of simple HTML code. Today, web design includes combinations of Cascading Style Sheets, JavaScript and other related content. **Ajax** (*Asynchronous JavaScript and XML*) technology has also gained a lot of momentum in the last several years to add interactivity through rich internet applications. Emerging technologies that you should also become familiar with include CSS3 (*the latest revisions to CSS*) and HTML5 (*incorporating JavaScript, CSS3 and adding new, frequently used tags.*)

*Note: **XML** stands for eXtensible Markup Language and has been used in the print world for decades.*

Adobe Technologies

Flex & AIR



Adobe Flex or **Flash** framework is used to create *Rich Internet Applications (RIA)*. **Flash Player** (*for the web browser*) and **Adobe AIR** (*for the desktop*) are the runtimes for those applications. Flash was the standard for building these applications, but as more programmers became involved, they looked for another option. Flex offers extendable classes and an **Integrated Development Environment (IDE)** called Flex Builder. Flex Builder allows programmers to quickly build rich internet applications using ActionScript and MXML (*which is a flavor of XML created by Macromedia*). There are rumors that this stands for “Magic eXtensible Markup Language” but no one will officially confirm this. Ok, is that enough acronyms for one day?

Note: There is an excellent series of tutorials available free from Adobe using the Adobe Media Player. Simply add this link to your RSS Feed in the My Favorites section (at the top of the screen):

<http://sessions.adobe.com/FlexInAWeek/feed.xml>

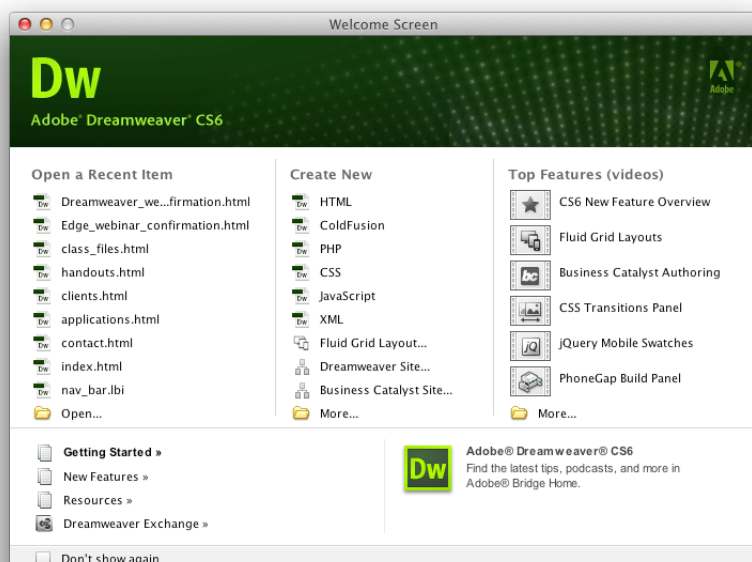
Creating a Site

There are about 4 places where you can start a new site. One of the easiest is the Welcome Screen, which I usually disable by checking the box in the lower left corner. If you have disabled your welcome screen and want it back you can go to **Dreamweaver > Preferences (Mac)** or **Edit > Preferences (Windows)** and turn it back on in the **General tab**.

Defining a Site lets Dreamweaver know where your files are located and where your files will be uploaded when they are ready (*although you don't have to specify this up front*).

Also in the Welcome screen on the right is a list of **Top Features (videos)**. I highly recommend you work through these as a refresher to this course. They will link you to **tv.adobe.com**, so you will need an internet connection.

*Note: When it comes to building websites, it is important to have a **site map** already in your head, even if all the content isn't developed yet. Be sure to **brainstorm** with all creators of the site to plan for any possible pages you'll need. It is far easier to make room in the site for a page to be added in the future, than to go back and add links everywhere for a page that you forgot.*



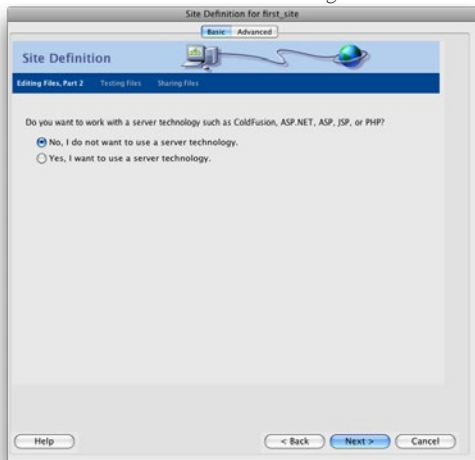
Site Setup

I'll start by choosing **Site > New Site**. The Site Definition dialog will now appear, guiding you through your initial site setup. There is a Basic & Advanced tab across the top. We'll be setting up a Basic site.

1. We'll start with **Site > New Site** and name the site **first_site**, then click **Next**.



2. For this site we won't be using a server technology, so select **No**, then click **Next**.



Note: Naming Conventions

My First Website.html is not web-compliant. In addition, many Unix servers, may have a problem with illegal characters such as / & ; ; and may be sensitive about the use of a period. To build good work habits, you should always name files with lower case characters, instead of a space use underscore_ (unless you are doing JavaScript). JavaScripters will typically change case in order to separate words. Never use more than one period in a file name, separating it from it's proper 2, 3 or 4 letter extension.

Example:

`my_first_website.html` (correct)

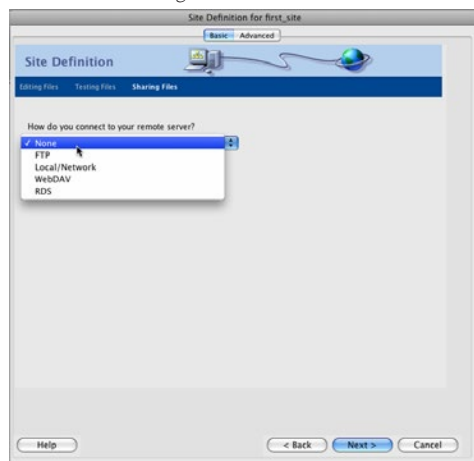
`MyFirstWebsite.html` (correct)

`My First Website.html` (incorrect)

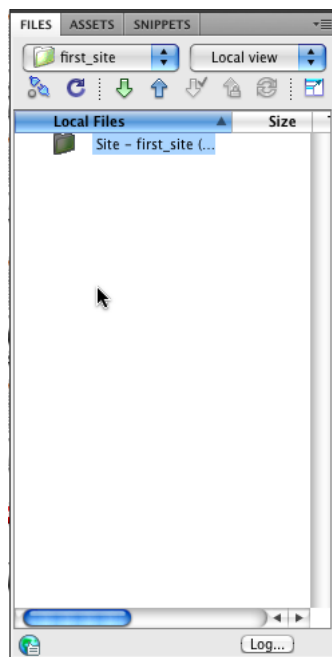
3. Edit local copies on my machine should be selected, click **Next**.



4. Under Sharing Files click Local/Network and choose **None**, click **Next**.

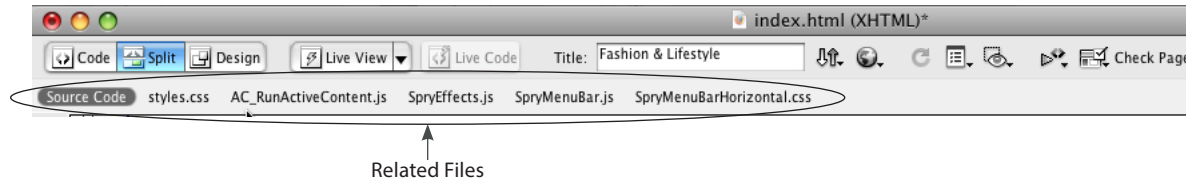


5. When you are finished you will see your site folder appear at the right of your screen in the **Files** panel.



CS6 Features

Features in CS6 such as the **Related Files** toolbar allow designers to see and edit all necessary external content associated with their HTML page. All referenced content will be displayed here such as CSS, JavaScript, PHP (*Personal Home Page*) and ASP (*Active Server Pages*).



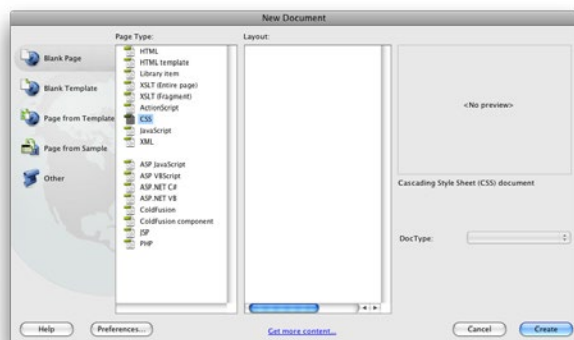
Other features we will cover include the **Code Navigator** to help you find and edit your Cascading Style Sheets and the **Live View** to dynamically see changes made in your CSS without having to launch a browser.

When it comes to **Cascading Style Sheets**, Dreamweaver has made it easier to implement CSS best practices without writing code. You can now create **New CSS Rules** in the **Properties panel** and get clear, concise explanations of where each property fits into the cascade of styles.

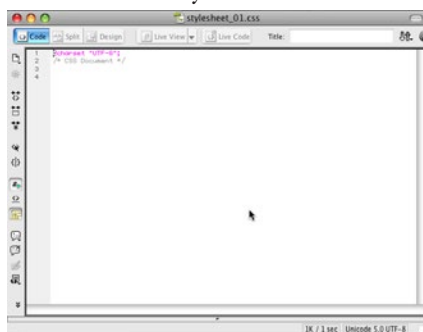
Building & Applying External CSS in Dreamweaver

Now that you have a good fundamental understanding of CSS, it's time to let Dreamweaver guide you through making and applying those styles. For the first example, we'll start by creating a blank, External CSS document.

1. In Dreamweaver choose **File > New** (**Command N** or **Ctrl N**)
2. Under **Page Type** click on **CSS**



3. Click **Create**
4. Save the file to your folder on the desktop as **stylesheet_01.css**



Building a Site

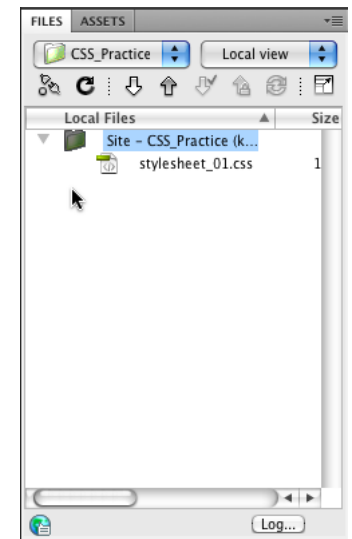
As a teacher, I find there is no substitute for building from scratch, so we'll start with a new, blank website.

1. Go to **Site > New Site...**



Note: If there are files in your folder, they will be automatically added to the Files panel on the right side of your screen.

2. Name the site **CSS_Practice**, you can leave the HTTP address blank
3. Click **Next**
4. Select **No, I don't want to use a server technology**, click **Next**
5. Select, Edit local copies on my machine, then upload to server when ready, click **Next**
6. Click the **folder** on the right of the path to store your files and select your folder on the Desktop
7. Click **Next**
8. Choose **None** in How do you connect to your remote server, click **Next**
9. Click **Done**



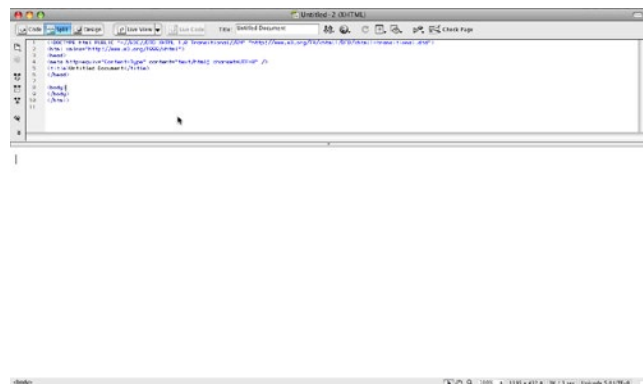
Creating a Simple Page

Now we are ready to start building.

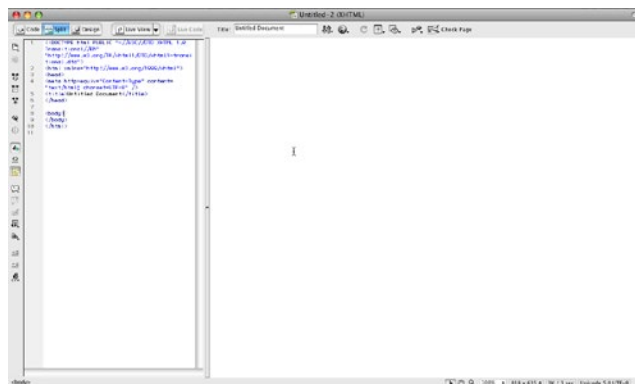
1. Chose **File > New...**
2. Select **Blank Page** at the top left, **HTML** in the middle and **<none>** under **Layout**
3. Click **Create**
4. Title this page **Fun with CSS** in the Title bar across the top
5. Save the document as **index.html** (*this is typically the name of any home page*)

One of my favorite new features of Dreamweaver CS4 is the ability to split your Design & Code view **Vertically**, instead of **Horizontally** (*the default*). To do this, be sure that Split is selected at the top left, then go to **View > Split Vertically**.

Horizontal View

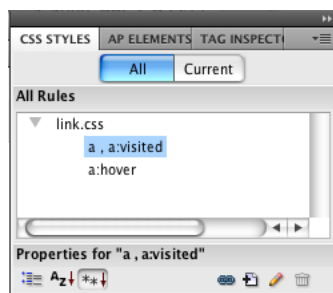


New Vertical View



Creating a CSS for Anchor Tags

When building an `<a>` tag CSS, setup is quite simple and you can follow the instructions from the previous pages. To get more specific and control the rollover, right-click or Ctrl-click (Mac) and choose **Edit Selector**. Rename the a CSS you built to `a, a:visited`. This will specify that the static anchor and one that has been clicked have the same appearance.



```
1 @charset "UTF-8";
2 /* CSS Document */
3
4 a, a:visited {
5     font-family: Verdana, Geneva, sans-serif;
6     font-size: 0.8em;
7     color: #00F;
8 }
9
10 a:hover {
11     font-family: Verdana, Geneva, sans-serif;
12     color: #900;
13 }
```

Cascading Style Sheets (CSS)

Cascading Style Sheets give us a better method than HTML for specifying how content should be presented on a web page. The main reason to use CSS is to separate **Structure** and **Presentation**. CSS can control many things, including: fonts, colors, background images and page layout. Another important reason to use CSS, is to have custom stylesheets for different devices. You could use one stylesheet for **Print**, one for **On-screen** and a third for **Mobile** devices. As you make a change to the CSS, when using External styles, those changes **Cascade** down to all the pages referencing that CSS document.

One important reminder, as with HTML, different browsers interpret Cascading Style Sheets differently. It is essential to test your pages on as many browsers and platforms as possible. Currently these are the most popular browsers, at a minimum you should keep 2 or 3 on your hard drive:

- Internet Explorer
- Firefox & Mozilla
- Netscape
- Safari
- Opera
- Mobile browsers (Safari for iPhone)

Three Methods for Applying CSS

There are 3 methods for applying Cascading Style Sheets to your web pages: Inline, Embedded or External.

Inline

Inline Cascading Style Sheets occur directly in the HTML and are supported by every tag. They are easy to understand, but not very practical since you'd have to keep repeating common style changes throughout the HTML.

Example:

```
<body>
  <h1 style="color: #669966; font:Verdana, Geneva, sans-serif;">Welcome to CSS</h1>
  <p style="font:Verdana, Geneva, sans-serif;">This is the first paragraph.</p>
  <p style="font:Verdana, Geneva, sans-serif; font-size: .7em; text-align:center">
    Copyright 2008 Kelly McCathran</p>
</body>
```

Embedded

Embedded Cascading Style Sheets specify all style information in the header of the HTML, using a style tag. This gives the page "rules".

Example:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>CSS Practice</title>
<style type="text/css">
  <!--
    h1 {
      font-family:Verdana, Geneva, sans-serif;
      color:#669966;
    }
  -->
</style>
</head>
```

Note: Typically for Embedded CSS, older browsers need HTML comments wrapping the CSS. This way if it can't interpret the styles, the code won't be displayed.

External

External Cascading Style Sheets link to a separate CSS document; thereby allowing web browsers to only have to download the style sheet once and use it multiple times.

Example:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>CSS Practice</title>
<link rel="stylesheet"
      type="text/css" href="CSS_practice.css"/>
</head>
```

CSS Document "CSS_practice.css"

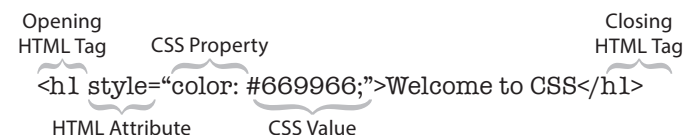
```
@charset "UTF-8";
/* CSS Document */

h1 {
    font-family:Verdana, Geneva, sans-serif;
    color:#669966;
}
p {
    font-family:Verdana, Geneva, sans-serif;
    font-size:1em;
}
```

Note: You can also have multiple external CSS docs referenced in the same HTML file.

CSS Vocabulary

Although Dreamweaver aids in the building of CSS styles, every good web designer should be able to “*talk the talk*”.



Opening
HTML Tag CSS Property Closing
HTML Tag

`<h1 style="color: #669966;">Welcome to CSS</h1>`

HTML Attribute CSS Value

CSS **Properties** are followed by colon and the value or values are listed after.

CSS **Values** are listed after properties. To list more than one value a comma is used. Semicolons are used to end a line (called a *declaration*). Also, the semicolon is equivalent to a return in UNIX.

CSS **Selectors** have 3 categories: Element, ID, or Class (*discussed later in this chapter*).

CSS **Rules** apply to a selector or style being reformatted with properties and values in the header of the HTML. A stylesheet is a group of rules.

Example:

```
property:value;  
color:#696;
```

Note: Stylesheet Rules are applied from the top-down. Rules that occur later in the Stylesheet override or update earlier rules.

Types of CSS Selectors

Element

CSS Rules can be applied to an **HTML Element** (*tag*) such as the `<p>` or `<h1>` tags and **Selectors** are used to identify that rule. These are my favorite way to define the formatting for large portions of text and they redefine all occurrences of that element on the page. In the CSS you identify the Element and re-define it's style settings.

Example:

```
h1 {
    font-family:Verdana, Geneva, sans-serif;
    color:#669966;
}
```

ID

The 2nd type of Selector is **ID Selector**, as mentioned earlier. With ID selectors you are identifying unique areas of the page that are named within any tag. IDs can only call to one element on a page.

IDs aren't limited to DIV & SPAN, but they are good examples of how IDs can be used. SPAN tags are more finite than DIV, they can specify a formatting change on a single character. The DIV tag defines a **division/section** in a document. Standard DIV attributes include: id, class, title, style, dir, lang, xml:lang

Example:

```
<div id="copyright">Copyright 2008 - Kelly McCathran</div>
```

Example:

```
<style type="text/css">
    #copyright {
        font-family:Verdana, Geneva, sans-serif;
        font-size:.7em;
        text-align:center;
        color:#999;
    }
</style>
<div id="copyright">Copyright 2008 - Kelly McCathran</div>
```

Class

The 3rd type of Selector is a **Class selector** which can be applied to any text in the HTML, regardless of the tags used to format the text. All Class selectors start with a period (.) and can be used over and over again in the same page.

Example:

```
<style type="text/css">
.quote {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-style: italic;
  color: #666;
}
-->
</style>

<p class="quote">If you can't lead by example, at least be a horrible warning.</p>
```

Note: HTML Elements can have both Class & ID properties assigned to them:

- *Class settings override default Element properties*
- *ID settings override Class and default Element properties*

3-Digit Hexadecimal Values

The three-digit hexadecimal color value can be listed in the form #RGB, where RGB is a three-digit number that can be expanded to define the six-digit color. In this usage, each digit is repeated once.

Example:

#RGB maps to the color #RRGGBB

#696 maps to the color #669966

Resource Web Sites

World Wide Web Consortium, Cascading Style Sheets home page:

<http://www.w3.org/Style/CSS/>

Tutorial for Cascading Style Sheets:

<http://www.w3schools.com/css>

See what CSS Can do:

<http://CSSzengarden.com>

On the Windows platform you can typically only run one version of Internet Explorer, here is a website to test your page with multiple versions:

http://tredosoft.com/Multiple_IE

HTML5



HTML5 is the latest iteration of HTML, but it is so much more than that. HTML5 builds on the W3C's (*World Wide Web Consortium*) open web platform. HTML5 attempts to foster development with the full potential of the web.

HTML5 supports a richer (and new) set of tags, microdata and microformats. RDFa (Resource Description Framework in Attributes) technology is also part of HTML5. RDFa provides a set of markup attributes that has machine-readable hints. Here is a real world example: If I see the website zappos.com and I click the "Like" button on Facebook™ my news feed can change to display more catered information, based on the knowledge that I like Zappos products. This means that we have an evolving, data-driven web experience.

The Need for New HTML Elements

When HTML5 was being developed, they researched the most commonly used ID and Class names (in CSS). Some of the most popular results were:

- Header
- Footer
- Nav
- Section
- Article
- Figure
- Audio
- Video
- Embed
- Time
- Progress

For this reason these new HTML elements are built-into HTML5. Think of them as an addition to the standard `<h1>` and `<p>` tags.

HTML5 Resources

List of New Elements in HTML5

<http://www.w3.org/TR/html5-diff/#new-elements>

Adobe Developer Connection | HTML5 Info

<http://www.adobe.com/devnet/html5.html>

Free HTML5 Templates

<http://freehtml5templates.com/>

Browser Testing of HTML

Adobe BrowserLab for Desktop

Adobe Shadow for Devices

Download Shadown from Labs.Adobe.com:

<http://labs.adobe.com/technologies/shadow/>

Shadow Instructions

<http://www.adobe.com/devnet/shadow/articles/browser-testing-across-devices-with-shadow.html>

CSS3

CSS3 along with HTML5 provides a more powerful way to adapt print publications to a rich digital format, far beyond the original capabilities of the first generation ePub (for eReading devices).

Rounded CSS Corners

A popular request when styling a page is to add CSS rounded corners or “*the CSS corner*.” CSS corners that aren’t square are in high demand for essential elements in web page design and app design. The elements that absolutely need rounded corners are commonly the “*chrome*” of a web application or page. Chrome is not only the name of Google’s browser, but it is also a generic term referring to outer parts of a window or app. Items that can be classed as chrome are: scroll bars, buttons, visually attractive tabs and dialog boxes. Rounding the corners give a more visually-polished look and feel to your UI (*user interface*) and enhance the overall UX (*user experience*).

Until CSS3, rounded corners had to be created by using tables and tiny rounded images that were placed in each corner. While this worked in EVERY browser, it was cumbersome to write the code (to say the least). From this design need many alternate methods sprung up, blog posts claiming “this is the way to do it”, dedicated JavaScript libraries and many JQuery plug-ins.

Today we use the `border-radius` property in CSS3—Curve Radii to accomplish rounded corners. Example:

```
p {  
    border-radius: 20 px;  
    background: #696;  
    margin-left: 80;  
    margin-right: 80;  
}
```

Rounded Corners for Multiple Browsers

To compensate for the difference among browsers, many people create a CSS Class with selectors for multiple browsers.

Example

```
.rounded-corners {
  -webkit-border-radius: 30px;
  -moz-border-radius: 30px;
  -o-border-radius: 30px;
  border-radius: 30px;
}
```

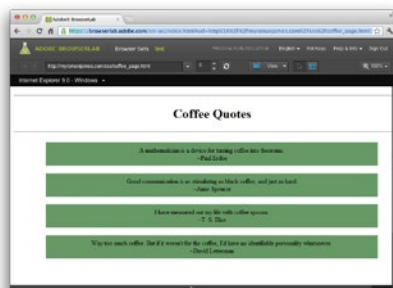
CSS Border-radius Prefix

-webkit- is for Chrome & Safari
 -moz- is for Firefox
 -o- is for Opera

In the example above the prefix wrapped in - - is for additional browser support. Most browsers released or updated after 2011 don't require this selector. Many coders leave this in, to compensate for large government agencies or corporations that only upgrade their software every 5 years.

Rounded Corners for IE9

IE9 in Adobe® BrowserLab



IE8 and older simply do not support border-radius. IE9 is reported to, it doesn't work with the above CSS. There are many recommendations for extra code to get this to work (adding to the CSS, HTML, using a plug-in, or writing JavaScript). None of which I believe is worth the time here.

