

CSS Affinity by PVII



Select one of 5 **responsive** CSS layouts and then watch as Affinity creates your page instantly. It couldn't be any easier. We hope you enjoy using this product as much as we did making it.

Al Sparber & Gerry Jacobsen PVII

Table of Contents

CSS Affinity by PVII.....	1
Install the extension	5
Work in a defined Dreamweaver site.....	5
Specify local site location	5
Site Name	5
Local Site Folder.....	5
Creating a CSS Affinity page.....	6
Site Wide Management.....	6
Assets folders	7
Uploading to your server.....	7
Mobile-Ready and Responsive	8
Feature-rich, accessible, and usable.....	9
Managing a CSS Affinity Site	9
Avoid adding fixed-width elements.....	9
Dynamic Image Sizing with CSS	10

CSS Layout Columns	13
Default CSS Classes Assigned to Columns.....	13
Changing column attributes	14
Changing Column Width.....	15
Editing Text Attributes	15
CSS Affinity CSS fundamentals	16
The Viewport Meta Tag	24
CSS3 Essentials.....	25
CSS Border-Radius (rounded corners).....	25
border-radius syntax	25
CSS box-shadow.....	26
box-shadow syntax.....	26
Editing and determining RGBA colors	27
Making aero or glassy style effects using RGBA and box-shadow.....	28
CSS3 Transitions	29

Support and Contact info	30
PVII Knowledge Base.....	30
PVII Communities	30
RSS News Feeds.....	31
Before you Contact us.....	31
Snail mail.....	31

Install the extension

Look for the extension installer files **p7_Affinity_117.mxp** and **p7_DMM_119.mxp** (The Drop Menu Magic system) in the root of the zip archive you downloaded. Double-click the files to install the extensions. Dreamweaver's Extension Manager will open and you will be prompted to complete the installation.

OS X users: If, upon double-clicking the installer, Extension Manager does not properly launch, you have a file association issue. To remedy the problem locate the installer file from inside a Finder window and double-click it.

Work in a defined Dreamweaver site

Before you begin, make sure you are working inside a defined Dreamweaver web site. This is necessary so that Dreamweaver knows how to link required assets. If you are new to Dreamweaver or need to learn how to define a web site, follow these simple steps:

Choose Site > New Site

Specify local site location

The Site category of the Site Setup dialog box is the only one you need to fill out to begin working on your Dreamweaver site. This category lets you specify the local folder where you'll store all of your site files. When you're ready, you can fill out the other categories in the **Site Setup dialog box**, including the Servers category, where you can specify a remote folder on your remote server.

Site Name

The name that appears in the Files panel and in the Manage Sites dialog box; it does not appear in the browser.

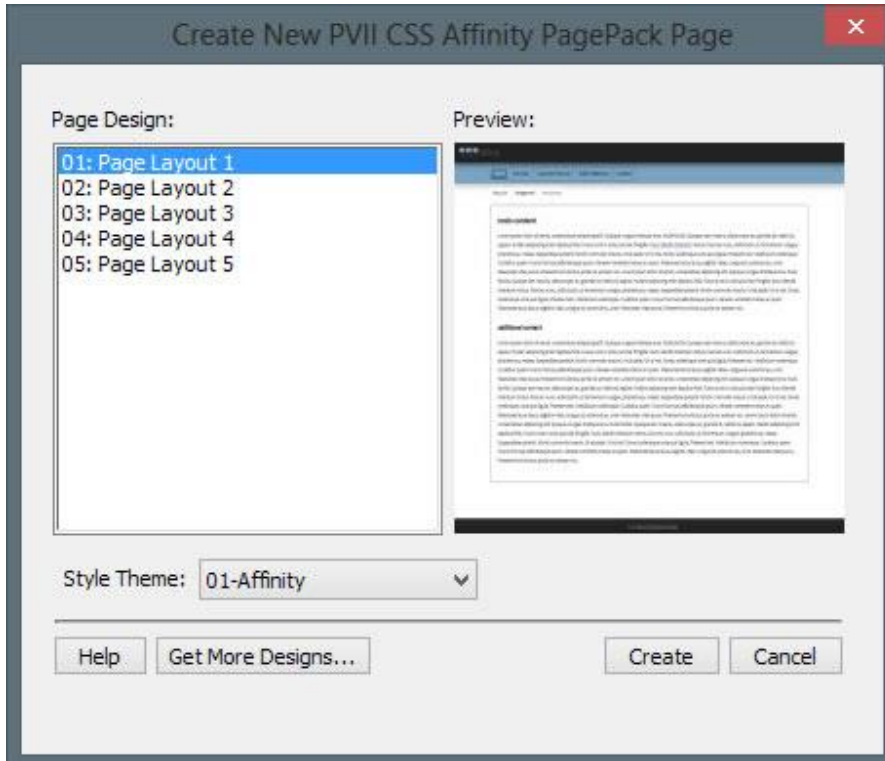
Local Site Folder

This is the name of the folder on your local disk where you store site files, templates, and library items. Create a folder on your hard disk or click the folder icon to browse to the folder. When Dreamweaver resolves site root-relative links, it does so relative to this folder.

Creating a CSS Affinity page

Let's see how easy it is to create a page. To create a page, choose: **File > New PVII PagePack > CSS Affinity**

The interface will open.



Select a layout from the **Page Design** column and the **Preview** image will change to reflect your choice. The Style Theme list will include any [Affinity Theme Packs](#) you might have purchased. Click the **Create** button. The **Save As** dialog will open. Browse to a folder within your current Dreamweaver-defined site so that it appears in the **Save in** box. Enter a **File name** and click **Save**.

Site Wide Management

Once you create, save, and customize your first page you should make a decision on how to manage repeating content, such as headers, menus, and footers. The two most popular methods would be Dreamweaver's Template utilities (DWT) or Server-Side Includes.

Assets folders

When your page is created, the Page Pack system generates the following folders

- **p7affinity**
This folder contains your *Affinity* CSS files
 - **img**
This sub-folder contains the Affinity logo
- **p7dmm**
This folder contains all of the assets used by the Drop Menu Magic system
 - **img**
This sub-folder contains the arrow images used by the Drop Menu Magic system when it is in accordion/responsive mode
- **p7ehc**
This folder contains the PVII Equal Height Columns script

The Page Pack system will create these folders in the same folder where you save your Page Pack page. This is a workflow feature that enables you to create multiple testing folders in which to *play*—but you must be mindful of where your relevant assets folders are when you ultimately publish your page or when you edit assets.

Uploading to your server

When you publish your page, you need to **upload the entire p7affinity, p7dmm, and p7ehc folders.**

Mobile-Ready and Responsive

CSS Affinity works seamlessly on desktop, laptop, smartphone, and tablet browsers—automatically.



You can test the responsive functionality without a mobile device. Simply open your page in a modern browser and adjust your browser window width. As you adjust the width, you will see your page respond.

Feature-rich, accessible, and usable

CSS Affinity uses modern CSS such as rounded corners, transitions, shadows, and **equal height columns**. This makes it extremely easy for you to maintain and customize your site. The advanced CSS we use is deployed in ways that enhance modern browsers while assuring an attractive and accessible presentation in older ones. The idea is to start with a solid foundation that is accessible to all browsers and devices and then enhance it for those that support advanced CSS. The result is that your page is accessible to all, attractively designed for all, with additional special effects available for modern devices. **Here are the key features:**

- Automatic Page-Creation interface
- **5 layouts** (choose from an assortment of fixed and flexible-width pages)
- Standards-based and accessible
- **Equal height columns without background images**
- Stable cross-browser compatibility
- **Included DMM Menu System**
- Compatible with PVII widgets
- Search engine friendly
- **Mobile-Ready** means you create one Web site that works on desktop, laptop, and mobile devices
- **Responsive** means that your page responds to the width of any device and any browser
- Logical **source code ordering** that enhances usability and search engine optimization
- **CSS 3 effects**, including CSS animations
- **PDF user guide**
- Free technical support

Managing a CSS Affinity Site

You chose CSS Affinity because you want a modern CSS layout that is responsive and mobile-ready. Here are the keys to keeping it that way:

Avoid adding fixed-width elements

When you add elements to your page, do not assign them fixed widths. Allow your elements to scale with the page. This is especially true of tables, forms, media objects, and images. Tables and forms are easy—*do not assign a width attribute*. Images and objects, however, have a natural width that must be countered. Here's how...

Dynamic Image Sizing with CSS

Setting images to dynamically resize allows you to use images in your pages without worrying about a perfect fit. The images will shrink to fit as you reduce the size of your browser window, and will expand as you increase the window size—while always maintaining its proper aspect ratio. The maximum size for your image will always be its natural size. We've taken care of the default banner image for you. In many (but not all) cases, when you add images to the content areas of your page, you need to make them scalable. We've included a special CSS class that you can assign to any image to make that image scalable. The name of the CSS class is **scalable**:

```
img.scalable {  
  height: auto !important;  
  width: auto !important;  
  max-width: 100%;  
}
```

Simply assign it to your image and it will scale.

```

```

We've also included a class called *fancy*, which you can assign to an image *in addition* to the *scalable* class to give your images rounded corners and a subtle shadow:

```
.fancy {  
  border: 1px solid rgba(255,255,255,.2);  
  box-shadow: 0px 0px 30px rgba(0,0,0,.65);  
}
```

```

```

What if I need to add an image gallery?

If you add an image gallery to your page, you will need to ensure that it scales. A perfect solution would be [PVII Image Gallery Magic 2](#), which is scalable and responsive by default. If you are using Image Rotator Magic, you can make your rotating images scalable by adding this rule to your page style sheet:

```
#p7IRM_1 img {  
  width: auto !important;  
  height: auto !important;  
  max-width: 100%;  
}
```

If you are using another image-handling script, you will need to contact its author to see if it can be made responsive.

Can my Video be responsive?

If you are deploying YouTube or Vimeo to serve your videos, you can easily make them responsive. Your video embed code will be an iframe. This method will also work if you are deploying your video using object and embed tags.

1. Wrap your video iframe inside a containing DIV with a class of *video-wrapper*:

```
<div class="video-wrapper"> <iframe width="640" height="360"
src="http://www.youtube.com/embed/xUQ_nBQZdtk?rel=0&hd=1" frameborder="0"
allowfullscreen></iframe></div>
```

2. Add the *video-wrapper* class to your CSS file:

```
.video-wrapper {
position: relative;
padding-bottom: 56.25%;
padding-top: 30px;
height: 0px;
overflow: hidden;
}
```

Note: The 56.25% padding bottom sets the aspect ratio to HD (16:9). Adjust as necessary.

3. Add this additional rule to your CSS file to position the iframe, object, or embed tags within the video-wrapper DIV:

```
.video-wrapper iframe, .video-wrapper object, .video-wrapper embed {
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
}
```

Can I make a Google Map responsive?

Sure. The technique is similar to the one above for making a video responsive.

1. Wrap your Google Map iframe inside a wrapper DV assigned the class *map-wrapper*:

```
<div class="map-wrapper">  
Google map embed code in here  
</div>
```

2. Add the following CSS classes to your CSS file:

```
.map-wrapper {  
position: relative;  
padding-bottom: 100%;  
padding-top: 0px;  
height: 0px;  
overflow: hidden;  
margin-bottom: 20px;  
}
```

Note: The 100% padding bottom sets the map box to a square shape.

```
.map-wrapper iframe, .map-wrapper object, .map-wrapper embed {  
position: absolute;  
top: 0;  
left: 0;  
width: 100%;  
height: 100%;  
}
```

Affinity CSS Notes

Though the resulting page designs are distinctly different, the Affinity default theme and [Garnet](#) theme use the same basic markup structure and CSS rules. The [Boardroom](#) theme, however, uses a totally different structure and CSS approach. If you are using Boardroom, please review the Boardroom Style and Snippets Guide:

[Boardroom Styles and Snippets Guide](#)

CSS Layout Columns

CSS Affinity creates one, two, and three-column designs. The columns are always the full height of your layout container. This is accomplished without using background images and enables you to change column widths and/or background with simple CSS edits—as opposed to having to edit background images.

Default CSS Classes Assigned to Columns

Page columns have one of three classes assigned:

1. **.sidebar**
2. **.main-content**
3. **.sidebar2**

In a 2-column layout, **sidebar** is either left or right of *main-content*. In a 3-column layout, **sidebar** is left and **sidebar2** is right of *main-content*.

Changing column attributes

Styling for your columns is assigned in the `.main-content .content`, `.sidebar .content` or `.sidebar-right .content` rules in your CSS Affinity style sheet:

```
.sidebar .content {  
padding: 20px 20px;  
font-size: .9em;  
line-height: 1.5em;  
background-color: #7BA4C1;  
background-image: linear-gradient( #7BA4C1, #FFF);  
border: 1px solid;  
border-color: #333;  
border-radius: 5px;  
}
```

Change the padding, font-size, and line-height to suit. Background-color is set for old browsers and a CSS3 linear gradient is used by modern browsers.

The gradient syntax is very easy to understand. The first value is the starting color. The last value is the ending color. Our gradient above is set to ramp from blue (`#7BA4C1`) to white (`#FFF`).

The background-color declaration is a fallback. That is, those older browsers that do not support gradients will use a traditional background color. If you edit the gradient, be sure to edit the background-color to a complimentary shade.

Border, border-color, and border-radius (for rounded corners) is also set on the column content classes.

Changing Column Width

Changing the width of a column is easily achieved. All Affinity columns in layouts 1-4 are styled using one or more of the following classes:

- **.main-content**
- **.sidebar**
- **.sidebar2**

For Layout 5 (the 3-column portal) we use these names:

- **.column-1**
- **.column-2**
- **.column-3**

The width is always declared as a percentage and the sum of the widths of all your columns, plus margins, is always 100%. If you change the width of one or more columns, simply make sure that when added together, the width of all your columns is 100%. Don't forget to add in the margins!

Tip: How can a fixed-width layout have column widths in percentages? In fixed-width layouts, the content-wrapper DIV is set to a fixed width. The percentages assigned to the columns, therefore, are essentially fixed since the wrapper width is fixed.

Editing Text Attributes

Text attributes are carried in the following rules:

- **.sidebar .content, .sidebar-right .content { }**
- **.main-content .content { }**
- **.column-1 .content** (Layout 5 only)
- **.column-2 .content** (Layout 5 only)
- **.column-3 .content** (Layout 5 only)

CSS Affinity CSS fundamentals

If your CSS skills are intermediate or advanced, you will find it very easy to customize your CSS Affinity pages. If you are a beginner, you should be able to undertake many customizations by opening the CSS files and reading them. The key rules contain comments. All 5 CSS files are very similar. Understand one and you will easily understand them all.

Note: Though the resulting page designs are distinctly different, the Affinity default theme and [Garnet](#) theme use the same basic markup structure and CSS rules. The [Boardroom](#) theme, however, uses a totally different structure and CSS approach. If you are using Boardroom, please review the Boardroom Style and Snippets Guide: [Boardroom Styles and Snippets Guide](#).

Let's look at the CSS file for Layout 2, `p7affinity_02.css`:

```
body {  
font-family: "Segoe UI", Optima, Helvetica, Arial, sans-serif;  
color: #000;  
background-color: #FFFFFF;  
margin: 0px;  
padding: 0px;  
font-size: 1em;  
padding-bottom: 4em;  
}
```

Use the `body` rule to change font attributes, background, and whitespace (margin and padding) globally for your page. The `padding-bottom` declaration (highlighted) leaves space at the bottom of the page for the fixed position footer. Edit the `padding-bottom` only if you add content to the footer that causes it to become taller.

```
img {vertical-align: bottom;}
```

When using a strict DOCTYPE, the image element is placed slightly above the bottom of its box. In other words, there is white space below the image. It's one of those silly things that the W3C did not really think through. To remedy this, we set all images to vertically align bottom.


```
.masthead {  
padding: 10px 10px;  
text-align: left;  
background-color: #222;  
position: relative;  
z-index: 10;  
box-shadow: 0px 0px 30px #000;  
}
```

Use the `text-align` property to align text or image elements inside the **masthead** left, center, or right. Adjust padding and other properties as desired. Do not edit the highlighted properties.

```
.content-wrapper {  
margin: 4em 1.25em 0em 1.25em;  
overflow: hidden;  
}
```

The **.content-wrapper** is the main layout container. Everything below the log and main menu is inside this element. The margins are set in shorthand to 4em top, 1.25em right, 0em bottom, and 1.25em left. The top margin is set to provide a buffer area for sub-menu items that drop down. **If you add a sufficient number of sub-menu items to cause them to wrap to 2 lines, increase the top margin to allow them room.**

```
.columns-wrapper {  
overflow: hidden;  
background-color: #FFF;  
position: relative;  
}
```

The **.columns-wrapper** is the containing element for the the main content columns.

```
.columns-wrapper, .menu-top-wrapper {  
max-width: 1260px;  
margin: auto;  
}
```

All Affinity layouts are set to a max-width of 1260 pixels. This rule makes it easy for you to customize your page width to whatever you need. Simply adjust the max-width value to suit or, if you want your page to be fully flexible, remove the max-width property. If you edit the width make sure you edit the width in your media queries so you do not affect the page presentation in mobile devices. If you change the max-width value you must also change the max-width value in the second media query:

```
@media only screen and (min-width: 700px) and (max-width: 1280px) {
```

The max-width in the media query above should always be 20px greater than the ax-width set on the **.columns-wrapper, .menu-top-wrapper** rule.

```
.main-content {  
float: left;  
width: 76%;  
margin-right: 1%;  
}
```

The main content column is floated left, assigned a width of 78%, and a right margin of 1%.

```
.sidebar {  
width: 23%;  
float: left;  
}
```

The sidebar column is floated left and assigned a width of 23%. The sum of the main-content, the main-content right margin, and the sidebar DIVs equals 100%.

```
.sidebar .content {  
padding: 20px 20px;  
font-size: .9em;  
line-height: 1.5em;  
background-color: #7BA4C1;  
background-image: linear-gradient(#7BA4C1, #FFF);  
border: 1px solid;  
border-color: #333;  
border-radius: 5px;  
}
```

Change the padding, font-size, and line-height to suit. Background-color is set for old browsers and a CSS3 linear gradient is used by modern browsers. The gradient syntax is very easy to understand. The first value is the starting color. The last value is the ending color. Our gradient above is set to ramp from blue (#7BA4C1) to white (#FFF). The background-color declaration is a fallback. That is, those older browsers that do not support gradients will use a traditional background color. If you edit the gradient, be sure to edit the background-color to a complimentary shade. Border, border-color, and border-radius (for rounded corners) is also set on the column content classes.

```
.main-content .content {  
padding: 20px 30px;  
font-size: 1em;  
line-height: 1.5em;  
border: 1px solid;  
border-radius: 5px;  
border-color: #666;  
}
```

This container is inside the **main-content** wrapper and serves as a container for the content inside, allowing you to adjust whitespace (padding) without altering the CSS box model. This is the rule to use to set text attributes for your main content and border effects.

```
h1, h2, h3, h4 {  
margin: 30px 0px 0px 0px;  
font-family: Federo, "Segoe UI", Optima, Helvetica, Arial, sans-serif;  
font-weight: normal;  
}  
h1 { font-size: 1.5em; }  
h2 { font-size: 1.35em; }  
h3 { font-size: 1.25em; }  
h4 { font-size: 1.1em; }
```

We first use a multiple CSS rule for heading levels h1-h4, setting margin and a Google font. Then we add separate rules for each heading level to set specific font sizes.

```
.content h1:first-child,  
.content h2:first-child,  
.content h3:first-child,  
.content h4:first-child {  
margin-top: 0px;  
}
```

A utility rule that zeroes top margin for the first heading in a column to eliminate its top margin.

```
img.scalable {  
height: auto !important;  
width: auto !important;  
max-width: 100%;  
border-radius: 5px;  
}
```

This special CSS class can be assigned to any image that you wish to have scale dynamically based on browser window width. If you don't like rounded corners, delete the border-radius property.

```
.fancy {  
border: 1px solid rgba(255,255,255,.2);  
box-shadow: 0px 0px 30px rgba(0,0,0,.65);  
}
```

This utility class can be assigned to any element to give it a subtle shadow effect.

```
.sidebar ul, .sidebar ol, .sidebar2 ul, .sidebar2 ol {  
margin: 0 0 0 .5em;  
padding: 0 0 0 .5em;  
line-height: normal;  
}
```

```
.sidebar li, .sidebar2 li {  
margin-bottom: 4px;  
}
```

We reduce normal indentation for ordered and unordered lists in your sidebars. Then we set a small bottom margin on all sidebar list items.

```
.content a {  
color: #002953;  
text-decoration: none;  
border-bottom-width: 1px;  
border-bottom-style: dotted;  
}  
.content a:hover, .content a:focus {  
color: #000;  
border-bottom-style: solid;  
}  
.footer a {  
color: #8C8C8C;  
}  
.footer a:hover, .footer a:focus {  
color: #FFF;  
}
```

Ordinary Link Styles. We set text link styles for the various content areas of your page. These link styles are separate from the navigation menu styles.

```
.top-navigation {  
background-color: #7BA4C1;  
border-bottom: 1px solid #FFF;  
box-shadow: 0px 0px 30px #999;  
position: relative;  
z-index: 9;  
}
```

This is the wrapper that contains the menu's background, border, and shadow effects. Do not edit the highlighted properties. When the menu is centered or otherwise aligns, this element allows the underlying background to be rendered full page width.

```
.top-navigation:after {  
content: "\0020";  
font-size: 0px;  
display: inline;  
overflow: hidden;  
line-height: 0;  
clear: both;  
}
```

This rule serves to clear the floats inside the menu. Do not edit this rule.

```
.menutop-wrapper .p7DMMo1 {  
background-color: transparent;  
border: none;  
box-shadow: none;  
}
```

The inner wrapper for the horizontal menu bar at the top of your page. This rule and its associated DIV allows us to assign a width to the menu so that it aligns with your page content. We turn off the default background, border, and box-shadow for the Drop Menu Magic menu. If you use Drop Menu on a non-Affinity page, it will carry its own border, background, and shadow effects.

```
.footer {  
background-color: #222;  
box-shadow: inset 0px 16px 2px rgba(255,255,255,.025);  
border-top: 1px solid #000;  
padding: 15px 15px;  
font-size: 0.7em;  
color: #8C8C8C;  
text-align: center;  
clear: both;  
position: fixed;  
top: auto;  
bottom: 0px;  
width: 100%;  
}
```

```
.footer p.copyright {  
text-transform: uppercase;  
margin: 0px;  
}
```

The footer is assigned background, a shadow, top border, padding, font-size, color, and alignment. The fixed value for position plants the footer at the bottom of your page and allows the page to scroll below it.

The Media Queries

At the bottom of your style sheet you will find media queries, which are what make your Affinity page responsive and mobile-ready. A media query is a special kind of CSS rule called an **@media** rule. The rule starts with an **@media** statement (the query) followed by an opening and closing curly brace. Inside those curly braces are CSS rules that modern browsers, and mobile devices, process based on the query. CSS Affinity CSS files contain 2 media queries. The first one is for browsers and devices whose width falls between 0 and 700 pixels. This would include conventional browsers when their window is (or is made) narrow, as well as all smartphones.

```
/*1 Column for Narrow Browser Windows and Smartphones in both orientations*/
@media only screen and (min-width: 0px) and (max-width: 700px) {
```

```
.content-wrapper, .columns-wrapper {margin: 0px; border: none;}
```

This first rule turns off margins and borders for the content containers.

```
.sidebar, .main-content {
width: auto;
float: none;
position: static;
}
```

This rule resets the width of the main content and sidebar(s) to auto, turns float off, and position to static, allowing the columns to linearize into single, vertically-stacked elements.

```
.main-content .content, .sidebar .content {
padding: 20px 10px;
border-radius: 0px ;
border: none;
height: auto !important;
max-height: 888678px;
}
```

This rule reduces the padding inside the content areas, turns off borders, sets height to auto and max-height to the odd value 888678px. The height and max-height settings are flags that are read by the PVII Equal Height Columns script, telling the script to ignore those elements. Because since they are no longer side-by-side, they do not need to be set to the same height.

```
.p7DMM01 ul a, .p7DMM01 ul a:hover, .p7DMM01 ul a:focus, .p7DMM01 ul a.open {
border-bottom: 1px solid;
border-color: #89AEC9;
}
```

This rule sets a bottom border on the main menu items when the menu is in responsive mode.

```
.p7DMMo1 ul li:last-child a {  
border-bottom: none;  
}
```

This rule turns off the bottom border for the last item in the menu.

```
}
```

This last curly brace is the one that closes the media query.

The second media query is for browsers and devices whose width falls between 700 and 1280 pixels. This applies to conventional browsers when their window is (or is made) narrow.

```
/*Medium windows - reduce padding content padding*/  
@media only screen and (min-width: 700px) and (max-width: 1280px) {  
  
.main-content .content, .sidebar .content {  
padding: 20px 15px;  
}
```

We reduce the padding inside the content areas.

```
.top-navigation {padding: 0px 10px;}
```

We reduce the padding around the main menu.

```
.content-wrapper {margin-left: 10px; margin-right: 10px;}
```

We reduce the margin around the content area.

```
}
```

This last curly brace closes the media query.

The Viewport Meta Tag

In order for media queries to function properly in mobile devices, we need to instruct the mobile device to set its viewport to the actual device width. This will prevent the device from trying to shrink the page. We do this with the viewport meta tag, which is automatically inserted by CSS Affinity into the head of your document:

```
<meta name="viewport" content="width=device-width">
```


CSS3 Essentials

A primer on Border-Radius, Box-Shadow, and RGBA color methods.

CSS Border-Radius (rounded corners)

Note: CSS3 border-radius is supported by modern browsers including Firefox 4, Opera 11, Safari 5, Chrome 2, and IE9 (and higher). Older browsers default to rectangular corners

border-radius syntax


Consider the declaration:

```
border-radius: 6px;
```

It creates a 6px radius curve at all 4 corners, which is expressed with a single value.


Tip: To increase the amount of curve, increase the radius value. To decrease the curve, reduce it.

You can also set a unique radius for each corner of an element. The border-radius property allows for 4 separate values, one for each corner. The values are expressed starting with the top-left and moving clockwise to the top-right, the bottom-right, and the bottom-left. While we don't use this technique in Affinity, it can come in handy.




This is the DIV

```
border-radius: 10px 20px 0px 40px;
```



This is the DIV

```
border-radius: 10px 10px 0px 0px;
```



This is the DIV

```
border-radius: 30px 0 0 30px;
```

CSS box-shadow

Note: CSS3 box-shadow is supported by modern browsers including Firefox, Opera, Safari, Chrome, and IE9 (and higher).

box-shadow syntax

The box-shadow property is extremely powerful and far more nuanced than border-radius. We'll cover some basics and how it's used in Affinity. For additional insight, we recommend reading the [W3C specification](#).

Consider this simple rule:

```
.theDIV {  
box-shadow: 10px 10px 20px #000;  
}
```



The values represent:

box-shadow: **horizontal offset** | **vertical offset** | **blur** | **color**

Don't forget the browser-specific property name for *Safari* browsers.

You can also declare multiple shadows for an element. For instance, use a regular shadow, followed by an inset shadow to create a bevel effect.

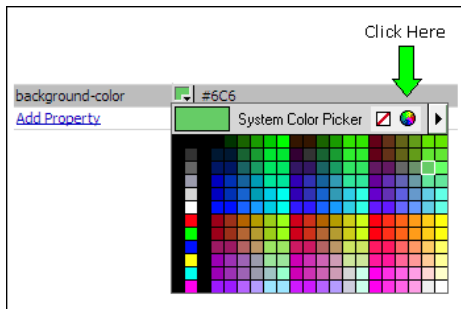
The 2 shadows are separated by a comma:

```
box-shadow: 0px 0px 50px #999, inset -3px 20px 40px #C05A5A;
```

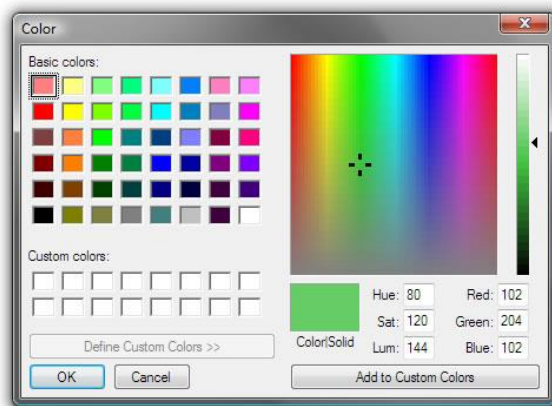
Editing and determining RGBA colors

The following procedures will enable you to convert HEX colors to RGBA in Dreamweaver CS5 and under. Dreamweaver CS5.5 and up have native support for RGBA built in. RGBA stands for **Red-Green-Blue-Alpha**. Each value is separated by a comma. The alpha value allows you to set a transparency level for the element being colored—be it a background color, a text color, or a border color. White (255, 255, 255) and black (0, 0, 0) are easy. Dreamweaver can help you to determine the RGB values using its standard color picker tool.

In your CSS Styles panel, select the rule you want to edit. Start by assigning a Hex color. Let's say #6C6. Open the color picker. Then click the *System Color Picker* icon:



The System Color Picker will open:



Your RGB numbers are listed as the Red, Green, Blue values: 102, 204, 102. So change your color value to:

rgba(102,204,102,1);

Remember, the "1" at the end is the alpha value and means your color is fully opaque (no transparency). To set transparency, choose a decimal value between 1 and 0. For example, to set 50% transparency, your values would be:

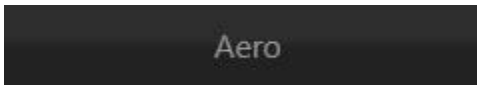
rgba(102,204,102,.5);

Making aero or glassy style effects using RGBA and box-shadow

It is possible to make a really neat aero or glassy effect. In fact, the menu links and the footer in Affinity deploy this technique. Consider this rule:

```
.aero {  
background-color: #222;  
box-shadow: inset 0px 20px 5px rgba(255,255,255,.15);  
padding: 10px;  
font-size: 1em;  
color: #999999;  
text-align: center;  
}
```

In a modern browser, this is the effect you will see:



By simply changing the background color, you can retain the effect without having to adjust the shadow color:

```
.aero {  
background-color: #CC3333;  
box-shadow: inset 0px 20px 5px rgba(255,255,255,.15);  
padding: 10px;  
font-size: 1em;  
color: #000000;  
text-align: center;  
}
```



CSS3 Transitions

The transition property sets up a pure CSS animation for modern browsers. Let's do a simple fade from transparent to a gray background-color on a link. The CSS:

```
a.fading-link {
  color: #000;
  background-color: transparent;
  transition: background-color linear .5s .1s;
}
```

```
a.fading-link:hover {
  color: #000;
  background-color: #DDD;
  text-decoration: none;
}
```

The values define the animation to occur on **background-color**, in a **linear** progression, for a duration of **.5** seconds, after a delay of **.1** seconds. CSS transitions are triggered by the hover event. The browser will compare all like properties between this rule and the hover rule below and animate the differences.

If you want to change both the background color and the text color, you would set the property attribute to **all**, like this:

```
a.fading-link {
  color: #000;
  background-color: transparent;
  transition: all linear .5s .1s;
}
```

```
a.fading-link:hover {
  color: #FFF;
  background-color: #999;
  text-decoration: none;
}
```

There are many, many more possible effects that can be animated with CSS. For additional insight, we recommend reading the [W3C specification](#).

Support and Contact info

PVII quality does not end with your purchase - it continues with the best customer support in the business.

PVII Knowledge Base

The PVII Knowledge Base is an online PVII application allowing you to access dozens of Tech Notes, tips, and techniques relating to our products, as well as to general web development issues..

[Open the Knowledge Base](#) | [View the 10 Most Recent Additions](#)

PVII Communities

PVII maintains separate and distinct Web Forum and Newsgroup communities. We do this to ensure you always have access via your preferred medium and device.

1. The PVII Web Forums

The PVII Web Forum is a modern browser-based community, accessible to both desktop and mobile devices and requires no additional software or plugins.

[Visit The PVII Web Forum community now...](#)

2. PVII Newsgroup forums

The Project VII Newsgroup is available via private and secure NNTP servers. In order to subscribe to a newsgroup you must have a newsgroup-capable client installed such as:

1. Mozilla Thunderbird
2. Windows Live Mail
3. Windows Mail/Outlook Express
4. Opera Mail
5. Entourage

Server: forums.projectseven.com

Note: PVII newsgroups are private and have nothing to do with usenet feeds that may be provided by your ISP. That is, you will not find our newsgroups in a list of newsgroups distributed by your internet service provider. You must set up our news server as a new account.

RSS News Feeds

Keep up with the latest news the minute it's released by subscribing to our RSS news feed. If you are not sure how to subscribe, please check this page:

[PVII RSS Info](#)

Before you Contact us

Before making a support inquiry, please be certain to have read the documentation that came with your product. Please include your Dreamweaver version, as well as your computer operating system type in all support correspondence.

E-Mail:

support@projectseven.com

Phones:

330-650-3675

336-374-4611

Phone hours are 9:00am - 5:00pm Eastern Time U.S.

Snail mail

Project Seven Development
339 Cristi Lane
Dobson, NC 27017