

CSS Page Builder Magic 2 by PVII



Select one of 16 **responsive** CSS layouts and then watch as Page Builder Magic 2 creates your page instantly.

We hope you enjoy using this product as much as we did making it.

Al Sparber & Gerry Jacobsen PVII

Contents

CSS Page Builder Magic 2 by PVII	1
Install the extension	4
Work in a defined Dreamweaver site	4
Specify local site location	4
Site Name	4
Local Site Folder.....	4
Creating a Page Builder Magic 2 page	5
Site Wide Management	6
Assets folders	6
Uploading to your server.....	6
Mobile-Ready and Responsive	6
Feature-rich, accessible, and usable.....	7
Key features:	7
New Features in Page Builder Magic 2:	7
Managing a Page Builder Magic 2 Site	7
Avoid adding fixed-width elements.....	8
Dynamic Image Sizing with CSS	8
CSS Layout Columns	11
Default CSS Classes Assigned to Columns.....	11
Changing column attributes	11
Changing Column Width.....	12

Editing Text Attributes	12
Page Builder Magic 2 CSS fundamentals	13
The Viewport Meta Tag	21
CSS3 Essentials.....	22
CSS Border-Radius (rounded corners).....	22
border-radius syntax	22
CSS box-shadow	23
Box-shadow syntax.....	23
Editing and determining RGBA colors	24
Making aero or glassy style effects using RGBA and box-shadow.....	25
CSS3 Transitions	26
Support and Contact info.....	27
PVII Knowledge Base.....	27
PVII Communities	27
RSS News Feeds.....	28
Before you Contact us.....	28
Snail mail.....	28

Install the extension

Look for the extension installer file **p7_PBM2_204.mxp** or **p7_PBM2_204_upg.mxp** (if you are upgrading from Page Builder 1) in the root of the zip archive you downloaded. Double-click the file to install the extension. 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.

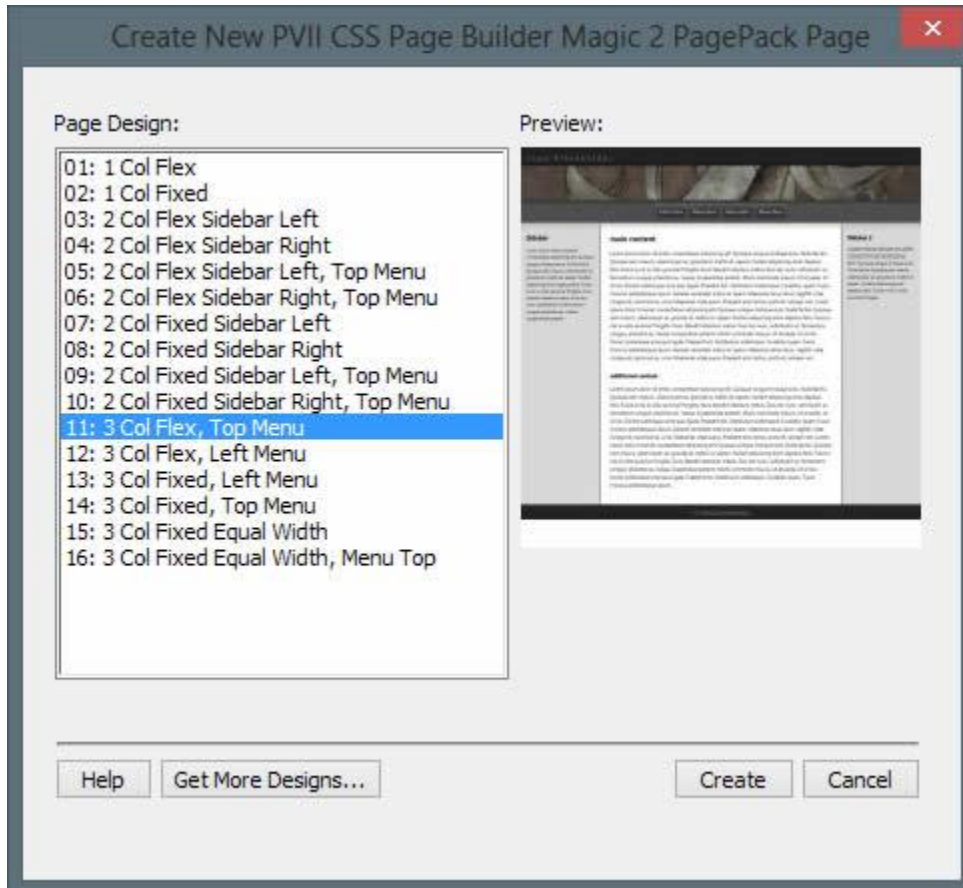
Local Site Folder

This is the 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 Page Builder Magic 2 page

Let's see how easy it is to create a page. To create a page, choose: **File > New PVII PagePack > CSS Page Builder Magic 2**

The interface will open.



Select a layout from the **Page Design** column and the **Preview** image will change to reflect your choice. 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:

p7csspbm2: This folder contains your *Page Builder Magic 2* CSS files

img: A sub-folder contains images used in all Page Builder Magic 2 pages.

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 p7csspbm and p7ehc folders**.

Mobile-Ready and Responsive

CSS Page Builder Magic 2 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

Page Builder Magic 2 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.

Key features:

- Automatic Page-Creation interface
- **16 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
- Compatible with PVII widgets and menu systems
- Search engine friendly
- **PDF user guide**
- Free technical support

New Features in Page Builder Magic 2:

- **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 pure CSS animated menus

Managing a Page Builder Magic 2 Site

You chose Page Builder Magic 2 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.

```

```

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 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%;  
}
```

CSS Layout Columns

Page Builder Magic 2 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 **.sidebar .content** or **.sidebar2 .content** rules in your CSS Page Builder Magic 2 style sheet:

```
.sidebar .content {
padding: 30px;
font-size: .85em;
line-height: 1.5em;
background-color: #D8D8D8;
background-image: -ms-linear-gradient(top, #FFF, #AAA);
background-image: -moz-linear-gradient(top, #FFF, #AAA);
background-image: -webkit-linear-gradient(top, #FFF, #AAA);
background-image: linear-gradient(top, #FFF, #AAA);
}
```

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 is assigned 4 times. The first 3 iterations use browser-specific prefixes: -ms- (Microsoft IE10), -moz- (Firefox), and -webkit- (Chrome and Safari). The last iteration is the W3 standard syntax, which will be supported by future versions of the aforementioned browsers.

The gradient syntax is very easy to understand. The first value is **top**, which instructs the browser to render the gradient from top to bottom. The second value is the starting color. The last value is the ending color. Our gradient above is set to ramp from white (#FFF) to medium-light gray (#AAA).

The background-color declaration is a fallback. That is, those older browsers that do not support gradients will use a traditional background color. We set the color to light gray (#D8D8D8), which falls between the starting and ending gradient colors. If you edit the gradient, be sure to edit the background-color to a complimentary shade.

Changing Column Width

Changing the width of a column is easily achieved. All Page Builder Magic columns are styled using one or more of the following classes:

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

The width is always declared as a percentage and the sum of the widths of all your columns 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%.

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, .sidebar2 .content { }**
- **.main-content .content { }**

Sidebar and sidebar2 are grouped together in a single rule. Should you want separate attributes, create a separate rule for each:

- **.sidebar .content { }**
- **.sidebar2 .content { }**

Page Builder Magic 2 CSS fundamentals

If your CSS skills are intermediate or advanced, you will find it very easy to customize your Page Builder Magic 2 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 16 CSS files are very similar. Understand one and you will easily understand them all.

Let's look at the CSS file for Layout 5, **p7csspbm_05.css**:

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

Use the **body** rule to change font attributes, background, and whitespace (margin and padding) globally for your page.

```
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 W3 did not really think through. To remedy this, we set all images to vertically align bottom.

```
.masthead {  
padding: 0px;  
text-align: left;  
background-color: #222;  
}
```

Use the text-align property to align text or image elements inside the **masthead** left, center, or right. Adjust padding as desired.

```
.masthead .logo {position: relative;}
```

The logo is positioned relatively so that it can be made to render below the box-shadows of adjacent elements. *It just looks better that way.*

```
.masthead h1 {  
margin: 0;  
color: #8C8C8C;  
padding: 20px 20px 10px 20px;  
font-size: 1.35em;  
font-weight: normal;  
letter-spacing: .35em;  
}
```

The default logo we provide is simply a level 1 heading, which may be replaced with an image of your logo. If you retain the heading, edit the **.masthead h1** rule to suit.

```
.masthead .banner {  
position: relative;  
z-index: 20;  
box-shadow: 0px 0px 20px #000;  
}
```

Positioning is set to ensure that the box-shadow renders above the logo DIV.

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

The masthead banner image is set to scale with the page.

```
.content-wrapper {  
border: 1px solid #222;  
margin: 0px;  
overflow: hidden;  
}
```

The **.content-wrapper** is the main layout container. The entire page is inside this element. Since layout 05 is flexible there is no width assigned. If this were a fixed-width layout, this is the element that would carry the width.

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

The **.columns-wrapper** is the containing element for the main content columns. We set this area to have a white background. This means that any individual column that is not assigned a specific background-color, will be white.

```
.main-content {  
float: right;  
width: 78%;  
}
```

The main content column is floated right and assigned a width of 78%. The main-content column comes before the sidebar column in the page source code but since both columns are floated right, the sidebar is visually rendered to the left.

```
.sidebar {  
width: 22%;  
float: right;  
}
```

The sidebar column is floated right and assigned a width of 22%. The sum of the main-content and sidebar DIVs equals 100%.

```
.sidebar .content {  
padding: 30px;  
font-size: .85em;  
line-height: 1.5em;  
background-color: #D8D8D8;  
background-image: -ms-linear-gradient(top, #FFF, #AAA);  
background-image: -moz-linear-gradient(top, #FFF, #AAA);  
background-image: -webkit-linear-gradient(top, #FFF, #AAA);  
background-image: linear-gradient(top, #FFF, #AAA);  
}
```

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 is assigned 4 times. The first 3 iterations use browser-specific prefixes: -ms- (Microsoft IE10), -moz- (Firefox), and -webkit- (Chrome and Safari). The last iteration is the W3 standard syntax, which will be supported by future versions of the aforementioned browsers. The gradient syntax is very easy to understand. The first value is **top**, which instructs the browser to render the gradient from top to bottom. The second value is the starting color. The last value is the ending color. Our gradient above is set to ramp from white (#FFF) to medium-light gray (#AAA). The background-color declaration is a fallback. That is, those older browsers that do not support gradients will use a traditional background color. We set the color to light gray (#D8D8D8), which falls between the starting and ending gradient colors. If you edit the gradient, be sure to edit the background-color to a complimentary shade.

```
.main-content .content {
padding: 30px;
font-size: 1em;
line-height: 1.5em;
border-left: 1px solid #000;
}
```

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 a left border.

```
.page-topper { margin-top: 0px; }
```

A utility rule assigned to the first heading in a column to eliminate its top margin. This class can be used on any element for which you would like to remove top margin.

```
h1, h2, h3, h4 { margin: 30px 0px 0px 0px;}
```

```
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. Then we add separate rules for each heading level to set specific font sizes.

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

This special CSS class can be assigned to any image that you wish to have scale dynamically based on browser window width.

```
.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: #0066CC; }
.content a:hover, .content a:focus { color: #C30; }
.footer a { color: #689BF0; }
.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.

```
.menutop-wrapper {
background-color: #6E6D68;
border-bottom: 1px solid #FFF;
box-shadow: 0px 0px 30px #999;
position: relative;
z-index: 10;
text-align: center;
padding: 10px;
background-image: url(img/pbm-menu-bg.jpg);
overflow: hidden;
}
```

The DIV that contains the horizontal menu bar at the top of your page. The white bottom border along with the gray shadow serve to create a neat bevel effect. Feel free to remove the background image or substitute your own. The text-align value is used to align your entire menu left, **center**, or right. By default, we have centered the menu.

```
ul.menutop {
margin: 0px;
padding: 0px;
display: inline-block;
vertical-align: bottom;
}
```

The Top Menu Bar **ul**. A display value of **inline-block** allows the menu to be positioned based on the text-align value of its parent container. A vertical-align value of bottom eliminates unwanted white space below the menu.

```
.menutop li {
list-style-type: none;
margin-right: 5px;
float: left;
}
```

The menu list items. Margin-right provides some separation between each item and may be edited.

```
.menutop li:last-child {margin-right: 0px;}
```

This rule resets the margin-right value to zero for the last menu item in the list.

```
.menutop a {
display: block;
color: #CCC;
text-decoration: none;
padding: 12px 12px;
font-size: 1em;
border: 1px solid;
background-color: #333;
box-shadow: inset 0px 12px 6px rgba(255,255,255,.15), 0px 0px 10px rgba(0,0,0,.5);
border-radius: 5px;
border-color: rgba(255,255,255,.5);
-webkit-transition: all linear .35s .1s;
transition: all linear .35s .1s;
}
```

The menu links. Display block is required. Padding provides white space within the link box. The first padding value controls the height of the box. The second value controls width. The box shadow carries both an inset and an outer shadow. Using RGBA colors with transparency for the shadow colors means that this shadow effect will work with virtually any background-color. The transition property sets up a pure CSS animation for modern browsers. The values instruct the animation to occur on **all** properties, in a **linear** progression, for a duration of **.35** 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.

```
.menutop a:hover, .menutop a:focus {
color: #FFFFFF;
background-color: rgba(51,51,51,0.2);
outline: 0;
}
```

The menu links hover and focus states. The outline value of 0 turns off browser-default focus outlines. The RGBA background color will be processed by modern browsers for a near-transparent effect. The change in color and background will, for modern browsers, be neatly animated according to the transition property in the default link state rule. Older browsers will simply switch colors.

```
.footer {  
background-color: #222;  
box-shadow: inset 0px 16px 2px rgba(255,255,255,.025);  
border-top: 1px solid #000;  
padding: 15px 30px;  
font-size: .7em;  
color: #8C8C8C;  
text-align: center;  
}  
  
.footer p.copyright {  
margin: 0px;  
}
```

The footer is assigned background, a shadow, top border, padding, font-size, color, and alignment. Then we make a special class called **copyright** assigned to a paragraph inside the footer.

The Media Queries

At the bottom of your style sheet you will find media queries, which are what make your Page Builder Magic 2 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. Page Builder Magic 2 CSS files contain 2 media queries. The first one is for browsers and devices whose width falls between 0 and 699 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: 699px) {
```

```
ul.menutop {display: block;}
```

This first rule sets the display value for the top menu to **block**, allowing it to span the full width of the layout.

```
.menutop li {
display: block;
float: none;
margin-right: 0px;
margin-bottom: 5px;
}
```

This rule sets the display value of the individual menu items to **block** and turns off the **float** property, allowing each menu item to display full-width on its own line.

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

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.

```
.content {
padding: 10px 10px !important;
border: none !important;
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.

```
}
```

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 900 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: 900px) {  
  
  .content {  
    padding: 10px 10px !important;  
  }  
}
```

We reduce the padding inside the content areas. Note that if we were looking at the CSS file for a fixed width layout, this query would also reset the content-wrapper's width to auto.

```
}
```

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 Page Builder Magic 2 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 Page Builder Magic, 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 Page Builder Magic. For additional insight, we recommend reading the [W3 specification](#).

Consider this simple rule:

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



This is the DIV

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:

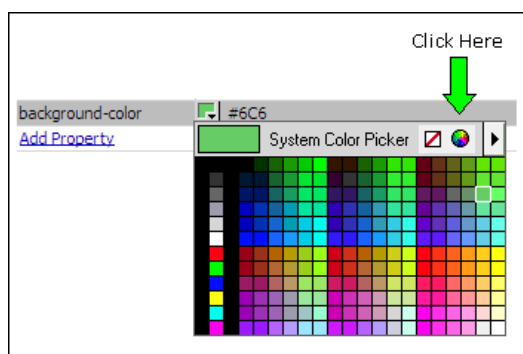
```
-webkit-box-shadow: 0px 0px 50px #999, inset -3px 20px 40px #C05A5A;
box-shadow: 0px 0px 50px #999, inset -3px 20px 40px #C05A5A;
```

Editing and determining RGBA colors

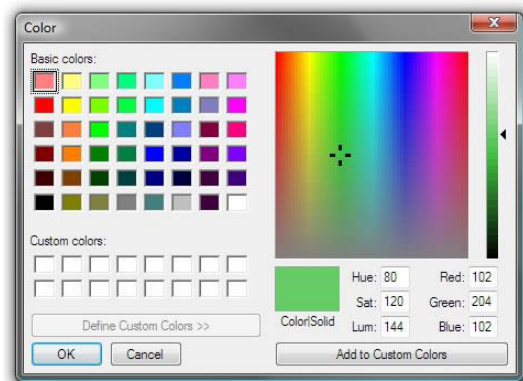
While we do not use RGBA colors in Page Builder Magic, you can certainly use them to create easy transparency effects. The following procedures will enable you to convert HEX colors to RGBA in Dreamweaver CS5 and under. Dreamweaver CS5.5 and up have native support for RGBA built in.

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:

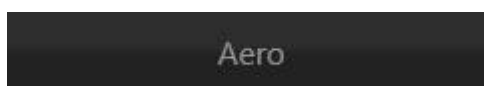
`rgba(102,204,102,.5);`

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 Page Builder Magic 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 instructs 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 [W3 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**

If you have another newsgroup-capable program that you are using, please see its documentation to learn how to add a new newsgroup account.

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