

CSS

Customizing SmugMug Simplified

—

Your Guide to a Stylin' SmugMug Site

written by: Ivar Borst

edited by: David Rosenthal

CSS

Customizing SmugMug Simplified

—

Your guide to a stylin' SmugMug site

written by: Ivar Borst

edited by: David Rosenthal

version 1 – March 6, 2007
version 1.1 - July 15, 2007

This material and its content may be freely distributed.
It may not be altered, amended or distributed for profit.

Warning: If you just want to know how to make some simple, specific changes to your site, and you don't really want to know anything about customizing, please go to the Digital Grin Customization forum (www.dgrin.com/forumdisplay.php?f=31), and ask your questions there. There are plenty of helpful folk there, waiting to help you make your site the way you want it. You do not need to read this guide to get help.

Now, if you got past that sentence, and you are still here, then let's move on.

Table of Contents

Preface	4
HTML	5
CSS	8
Firefox and the Web Developer Extension	10
Example: a Banner & the "margin" Property	12
WebDev: Display Element Information	16
Multiple Classes and divs (divs in divs)	19
Text & Font Properties	21
General Description of the SmugMug Pages	24
Clean Code and Comments	26
Problems, What To Do?	30
Advanced: Text Formatting & span	32
Index	34
APPENDIX A - CSS Properties and Values	35
APPENDIX B - Colors	43
APPENDIX C - CSS Pseudo Classes & Elements	44
APPENDIX D - Information and Links	45
APPENDIX E - Browser Specific Problems	46
APPENDIX F - CSS Cheatsheet	48

Preface

What this is: This is a basic introduction to customizing your SmugMug site, using CSS and HTML. We have tried to write it in such a way that after reading this you will be able to understand the basics of customizing, and you will be ready to tackle a lot of customization issues on your own. It covers and explains several examples of customization that are possible with your Power account or Pro account. You should be able to do a lot more than just the examples we give you after reading it. In other words, it's intended to "teach you how to fish".

Who this is for: This is for people who are looking to customize their site, and make it personal. This is for people who don't want to have to ask how to do things, but would rather be able to figure things out for themselves. Or, maybe they just want to better understand the answers they are given. Even if you are only doing a little bit of customizing, it may make things easier as it will help you better understand what you are doing. It's not required though. See the warning, on the first page.

If you decide to read this, but you don't understand it, do not be afraid to ask clarification. There is a separate Digital Grin thread about this, here:
<http://www.dgrin.com/showthread.php?t=54833>.

This information is in no way a complete and whole description of HTML and/or CSS. Some things are highly simplified, and some things are not mentioned at all. It's also written specifically for learning to understand HTML/CSS for use with SmugMug.

Ready to move on? Let's go!

HTML

HTML stands for **H**yper**T**ext **M**arkup **L**anguage, something you can forget right away. It's a language that is understood by your browser (Internet Explorer, Firefox, Safari, etc.).

An **internet page** is a file containing HTML. HTML can refer both to the language itself, as in "This was written in HTML", and to the actual code you have written for your site, as in "Go check your HTML".

The HTML usually consists of: the **head** and the **body**. The head can contain a lot of nice code, but it is not very important to us at this stage. What is important for us, is the body. Why? Because that's where everything we see, and some things we don't see, are created.

The body in SmugMug-land, consists of 3 parts. From top to bottom:

- the header
- the middle section (the part with the photos)
- the footer

The body can contain **elements**. An element is a bit of code that tells the browser to do something, or behave in a certain way. For instance, you can use an element to show an image, or to create a link to another website.

A special type of element is called a **div**. This comes from *division*. It can contain text and other elements. See it as being a container that you can put stuff in. You can specify the size and color of the container, among other things.

An element can be part of a **class** (from *classification*), and/or have an **id** (from *identifier*). These are two ways of identifying and addressing an element.

Think about class as being a type of car, and id being the license plate.
There can be more than one Jeep, but there can be only one with a certain license plate.

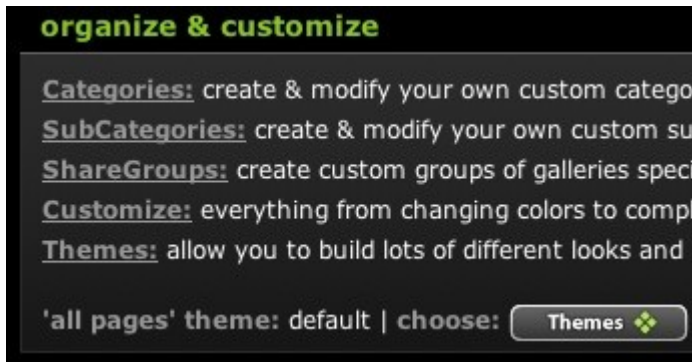
A class can be shared with other elements (the car model), while an id is specific for that element (the license plate).

A single element can have both an id and a class (or classes).

There can be a class "Jeep" and a class "4wd" and one car can be a member of both of them.

Later on, we will be using the classes and ids to style the elements.

Now, it's time to see how it actually works in your SmugMug account. First, make sure that you are logged in. Then go to your Control Panel. In your Control Panel there is a link that says "Customize". Click on that link.



The page you will now see has several fields where text can be added. Look at the 2 big boxes at the bottom of the page. They are labeled "header" and "footer".

Remember the header, middle section and footer we discussed earlier? These are the same header and footer. The middle section is created by SmugMug, so there's no box to edit it. The header and footer boxes are also the only boxes to which we will be adding our HTML code.

Let's create a new element. Okay, so this is where it all starts. Our very first element. Our first step to total customization. Are you nervous yet?

We'll start by creating a "div". Remember the div? It's like a container. You put things in it.

- Open your customize page (Control Panel > Customize).
- On your customize page, in the header box, type: `<div>`

Don't do anything else yet. That bit of code is HTML.

Now our browser will know that that this is where the element starts.

Hey, hold on. It doesn't know where it ends yet.

- Add the code: `</div>`

The slash is like the period at the end of a sentence. It indicates the end of that particular element.

Okay, so this is what you have now: `<div></div>`

Our browser now knows where the element starts, and where it stops.

(It's also referred to as opening, and closing). A div, must ALWAYS be closed.

- Let's add a name too. Change `<div></div>` into: `<div id="my_first_div"></div>`

header - optional

The header can contain words and graphics to replace smugmug's logo and menubar with your own. This is not the code that goes inbetween the `<head>` tags. Use HTML.

```
<div id="my_first_div"></div>
```

- Save this by clicking on 'update' at the bottom of the page.

Update

Congratulations! We just made and named your very first element!

- It's a container into which we can put other things.
- It has an id, it's called "my_first_div".

At this time the container is not visible and we didn't put anything in it yet, but don't worry. We will get back to this element in a little bit.

REVIEW:

- HTML is a language that creates and defines elements on a page.
- A class is a group that many elements can belong to.
- An id is the specific name for a specific element on that page, and is unique to that element on that page.
- One type of element is a div.
- A div is a container that can be styled, and can be filled with text and/or other elements.

CSS

CSS stands for **Cascading Style Sheet**. As you have seen before, HTML is saved in a file. CSS is also saved in a file. Although HTML and CSS are two different languages, they work together closely. The HTML creates the elements, and the CSS styles them. The CSS contains instructions on how to style existing elements only. Just like HTML, CSS can refer both to the language itself, as in "This was written in CSS," and the actual code you have written for your site, as in, "Go check your CSS."

How does that work? I'm glad you asked.

CSS works this way; First you identify what it is you want to style. Second, you tell it how you want it to be styled. Think about the car again. First we are going to say that we want to style all Jeeps. Then we will say that we want to paint them green, for example. In CSS that would look like this:

```
.jeep {  
  background-color: green;  
}
```

Notice the **period** (.) before the word 'jeep'? That's to indicate we are addressing a class here. So in this case, all jeeps will be getting the color "green".

What we want to style, in this case 'background-color', is called a **property**. What we change it to, in this case 'green', is called a **value**.

Now if we want to change just a single element, we will use the id (the "license plate" of the element). Let's change the div that we have created in the first chapter, and paint it red.

```
#my_first_div {  
  background-color: red;  
}
```

This time, you see a **pound-sign** (#) before the word. That is to indicate that we are talking about an id. Only that particular element will get a different color. Even though it may be a car, only this car with this license plate is painted red. All the other cars are unaffected.

Besides giving our car a color, you can also select a size for our car. (if only we could do that in real life!) Let's make our div 300 pixels wide, and 100 pixels high, just for practice:

```
#my_first_div {  
  width: 300px;  
  height: 100px;  
  background-color: red;  
}
```

Do you see the **semi-colon** (;) right after each value? That's to indicate that the styling of that property has finished, and the next property can be styled. You can add as many styling lines as you like, as long as they stay between the brackets.

Even though they are two separate things, both the CSS and the HTML are edited and saved at the same place in your SmugMug Control Panel (Control panel > Customize). The CSS is entered in the box on the customize page, labeled "CSS". You can find it all the way at the top.

Let's style our new element.

- Open your customize page (Control Panel > Customize).
- In the CSS box, enter:

```
#my_first_div {  
  width: 300px;  
  height: 100px;  
  background-color: red;  
}
```

CSS - optional

The Stylesheet section will be placed between <style> tags (so please don't add them below). Uses standard CSS syntax.

```
#my_first_div {  
  height: 100px;  
  width: 300px;  
  background-color: red;  
}
```

- Save this by clicking on 'submit' at the bottom of the page.

NOTE: This will actually style the box on your SmugMug pages, and you will see a red box on them. We will be removing it soon, though!

REVIEW:

- CSS styles the elements that are created by the HTML.
- First you identify an element, and then you style an element.
- In the CSS code, a class is preceded by a period (.) and an id is preceded by a pound-sign (#).

Firefox and the Web Developer Extension

So far, so good, eh? Well, now we're going to tell you the truth: using the customize page to edit your CSS and HTML is clunky and slow. Sure, that's where it needs to wind up in the end, but in order to edit your CSS effectively, you really need two things: **Firefox** (<http://www.mozilla.com>) and the **Web Developer Extension** (addons.mozilla.org/firefox/60/) We will be calling that **webdev** from now on. Firefox is a browser, and webdev is an add-on for Firefox. It is a tool that you access from within Firefox.

Before you take your next step in learning how to code CSS, you need to download and install these two applications. First, if you don't already have Firefox, visit the link above, and download it. Once you have downloaded and installed Firefox, open it, and visit the webdev link above. When you download the extension in Firefox, Firefox will recognize it as an extension and will install it so that the two will work properly together. Once installed, you must restart Firefox in order to access webdev. Do not download webdev from another browser (IE, Safari, etc.).

Ok, so now you've got the tools, how do you use them?

You'll notice, once you've restarted Firefox, that there is now a new toolbar in the browser:



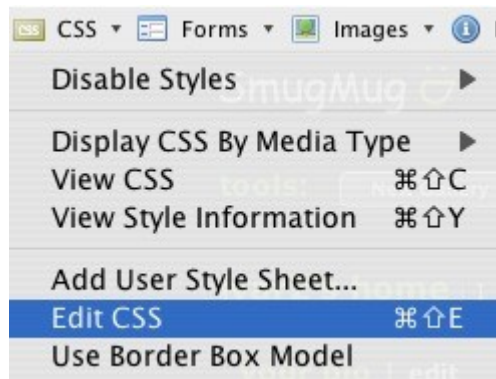
Don't worry about the amount of menus in the toolbar. It can be kind of intimidating, but we'll only be using 2 functions in webdev for now.

The web developer has a function called **Edit CSS** in the CSS menu. It will allow you to change your CSS, and to view the changes immediately.

Here's the best part: although you'll see your changes in the browser window, it doesn't actually apply it to your SmugMug stylesheets. Remember, your changes are only saved in Control Panel> Customize when you make the changes, and click on "submit".

Why would you want to do something that isn't saved to your SmugMug stylesheets? — So you can play around, make mistakes along the way, and no one else will see the changes until you save them. Now that's cool, eh? I wish I could make mistakes that no one could see all the time.

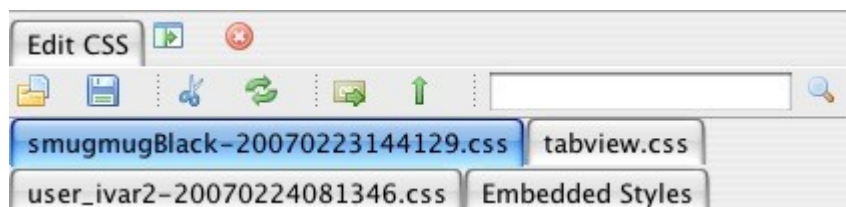
So go ahead select 'Edit CSS'.



The first thing you'll notice is that the whole layout of your browser window changed. You've now got the browser on the right, a notepad area on the left, and above the notepad are several tabs.

The number and names of the tabs will change depending on the page you're viewing. The only tab that you need to pay attention to is the one that has your nickname in it. The rest you can just ignore. (For now, at least!)

OK, one last thing to tell you before we move on to actually using the webdev. Above the tabs you'll see an area that looks like this:



There are two things here that you need to know about.

1. The X will close the Edit pane.
2. The green arrow pointing up or down, is for "sticking" your CSS changes.

Closing the pane is self-explanatory, but what do we mean by sticking your CSS?

The edit pane can have two types of behavior. By default it shows you the CSS for the active page. That means that if you navigate to another page, the CSS will update. That's all well and good, but what if you are in the middle of editing your CSS and you want to see how it's affecting another page -- you go to that page, and *poof!*, all the changes you've been working on go away into the ether. To avoid that, just click the green arrow. It will point upwards, indicating that the CSS in the edit pane is "stuck", overriding the CSS on any page you navigate to.

One of the tabs will have a name starting with "user...". This is your stylesheet. It's the same stylesheet where you have saved the CSS code in the exercise of the previous chapter. Click on that. In the text area, you will now see the code:

```
#my_first_div {  
  width: 300px;  
  height: 100px;  
  background-color: red;  
}
```

That's the code from the previous exercise. Now, remove that code from the text-area. See any changes? That's right, instant result! Now, close the edit css window by clicking on the small X of the 'Edit CSS' pane. See? The red container is back! The Edit CSS window doesn't actually apply the code, it just shows you what would happen.

Example: a Banner & the "margin" Property

Let's create a banner, you know, the pretty logo or name that identifies you at the top of your site; While doing that, we'll look at the property **margin** as well.

The first thing we need is a place to put our banner. We'll place it in a div.

Before we go work on the HTML and CSS code, first make sure you have an image you can use as the banner. If you don't have one, you can use this:
<http://www.smugmug.com/photos/132259920-O.jpg>.

If you are using your own banner, save the file, and upload it to one of your SmugMug galleries. If you are using the example-banner, there is no need to save the image and upload it to your galleries.

- Open up the customize page, and create a div by adding: `<div id="my_banner"></div>` in the box labeled "header".
- Click on 'submit'.

We now have the div that will contain the banner, and we have named it. It's called "my_banner".

Now, go to your SmugMug homepage, and then open the edit CSS window from the webdev menu.

- Start by adding:

```
#my_banner {  
}
```

This addresses our div. All of the properties and values will go between the braces.

Since our div doesn't have any dimensions yet, let's give it some. The sample banner has a height of 150 pixels and a width of 750 pixels. If you are using your own banner, substitute the width/height with that of your own banner.

- add the height and width to your CSS code
This is what you should have ended up with:

```
#my_banner {  
  width: 750px;  
  height: 150px;  
}
```

Now that we've given it dimensions, but it still doesn't show. It is located in the top-left corner.

Let's set your banner as the background of our div. We will use the property **background** for it.

- Add the location of your banner image by changing the code to the following:

```
#my_banner {  
  width: 750px;  
  height: 150px;  
  background: url(http://www.smugmug.com/photos/12345678-O.jpg) no-repeat;  
}
```

Have a look at the background line; The first thing that is mentioned is "url(background image location)". I added "no-repeat" behind it as well. This will only show the image once. If the width/height that you have given to the div matches that of the image exactly, it would only show once, anyway. Adding 'no-repeat' is good practice though.

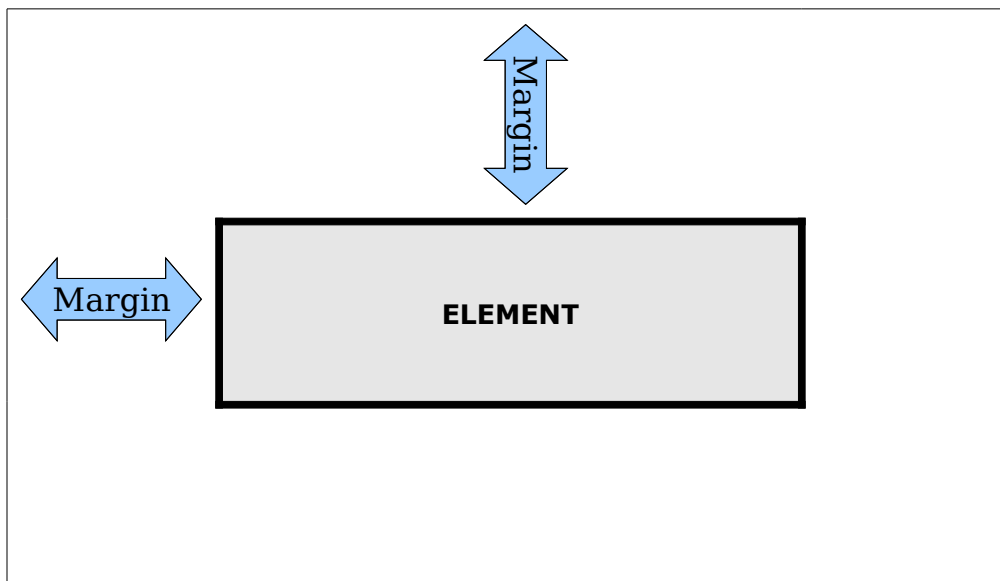
Other options with regards to the repeat value, are

- "repeat" (repeats the image in both horizontal and vertical direction),
- "repeat-x" (repeats the image horizontally only) and
- "repeat-y" (this repeats the image in the vertical direction only).

If you look at your banner now, it will show in the top-left corner. The goal is to get it horizontally centered on your page. There are several ways of moving the banner around.

The **margin** property sets the space around (on the outside) of the element.

You can use that to position the div.



The margin property

Using the margin property, you can set all margins individually, like so:

```
margin-top: 10px;  
margin-left: 10px;  
margin-right: 10px;  
margin-bottom: 10px;
```

but this shorthand version will do the same:

```
margin: 10px;
```

If you want to set margin to 10px on all sides, but 20px on the bottom side, you can do this:

```
margin-top: 10px;  
margin-left: 10px;  
margin-right: 10px;  
margin-bottom: 20px;
```

or this:

```
margin: 10px;  
margin-bottom: 20px;
```

Although both examples are correct and will work, the second example is preferred as less code takes less time to be loaded.

You can also use a shorthand version of the margin property using 2, or 4 values.

2 values:

```
margin: 20px 10px;
```

where the margins are in this order: top&bottom, left&right

4 values:

```
margin: 10px 10px 20px 10px;
```

where the margins are in this order: top, right, bottom, left

Although slightly less code to load, I personally don't prefer the 4 shorthand values, as it makes me need to think about what value goes where and I don't like to think :)

Besides being set to a fixed size, the 'margin' property can also be set to a percentage, or to a the value 'auto'. If you set both the left and right margin to 'auto', it will automatically select the margins on those sides as to keep the object in the middle, providing a width has been set.

Let's use the 'auto' value to align the banner we created. We'll use the following: `margin: 0 auto;` This shorthand version sets the margin on the top & bottom to 0, and on the right & left to 'auto'.

- Add the margin property to the code you already have:

```
#my_banner {  
  width: 750px;  
  height: 150px;  
  background: url(http://www.smugmug.com/photos/12345678-O.jpg) no-repeat;  
  margin: 0 auto;  
}
```

Now, the banner will show nicely centered. This is the final code you need to show it on all pages.

What if we only want the banner to show on the homepage, but not on any other pages?

We can 'hide' the banner by using the property and value: `display: none;`

- Let's add that to our code:

```
#my_banner {  
  width: 750px;  
  height: 150px;  
  background: url(http://www.smugmug.com/photos/12345678-O.jpg) no-repeat;  
  margin: 0 auto;  
  display: none;  
}
```

The banner now disappeared from all the pages, but we want to show it on the home page of our SmugMug account. How do we do that? We tell it to, using the following code.

- Add this to the Edit CSS window, below the existing code:

```
.homepage #my_banner {  
  display: block;  
}
```

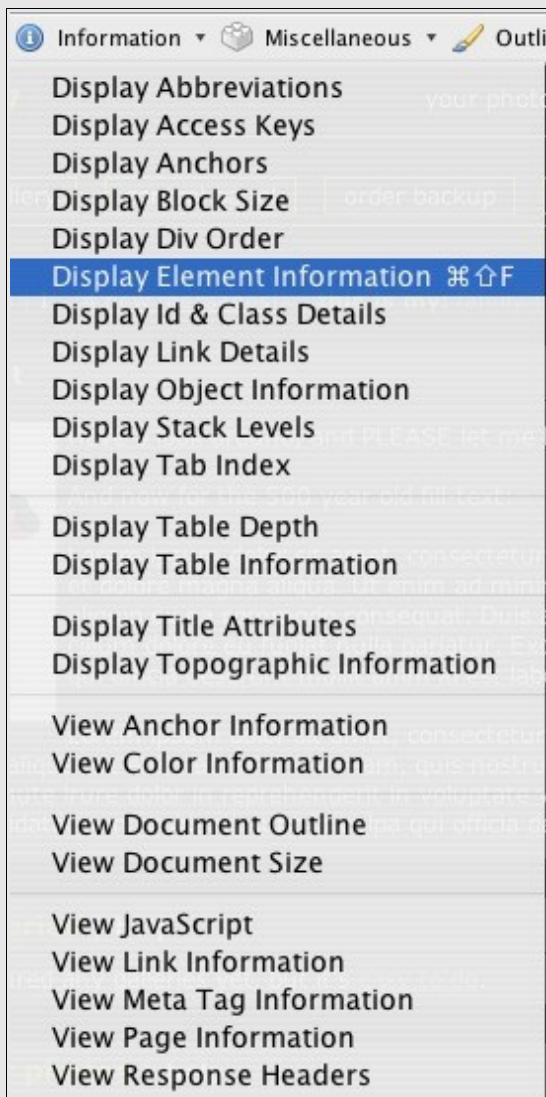
This CSS code first looks for a thing that is classed 'homepage'. After it's found the homepage class, it looks for a div called "my_banner". Only if BOTH the class AND the id are there, will the code have effect.

The big question is, how did I know that SmugMug has given the homepage the class 'homepage'? For that, I used the 'display element information' function from the Web Developer Extension. Let me show you how.

WebDev: Display Element Information

In order to tell your CSS what element to change, you need to know the name of the element. This is where the '**Display Element Information**' function from the webdev comes in handy.

- Find your 'Information' menu, and select 'Display Element Information'



After you select Display Element Information (either by the menu, or the shortcut listed), your cursor changes to a crosshair, and whatever element you put your cursor over gets a thin red border.

Clicking on the element, will show a lot of information in the yellow floatie.

Both the id and the class of the element selected show on the top row of the yellow floatie.

- At this point you should be able to tell what the id and class are. Take a moment to look at the top line of the floatie and see if you can name the id and class of this object.

div #breadcrumb .nav	
Attributes	
class = nav	
id = breadcrumb	
Position	
Left: 273px	Top: 102px
Width: 750px	Height: 30px
Other	
Font Family: verdana,tahoma,Helvetica,arial,sans-serif	
Font Size: 16px	
Ancestors	
html	
body .homepage loggedIn bodyColor_Black	
div #bodyWrapper	
div #homepage .homepage loggedIn bodyColor_Black	
Children	
div #all Views	
span #userName .title	
span #userHome .title	
span .note	
span .note	
span .note	
span .note	
div .spacer	

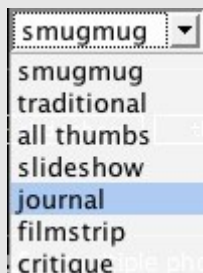
Answer:
id = breadcrumb
class = nav

A couple of things to know about using the floatie:

- If the floatie is covering something you want to see, click and hold the top bar of it, and drag it anywhere you want.
- *HELP, I don't see an ID or a class on the top row!!* Some elements are simply not named. Some elements are contained in another element that has an ID. These elements are called **Children**, and don't have an ID of their own. Not to worry, the ID of the **Ancestor** is also shown on the yellow floatie. Look under Ancestors and find the last line which starts with a 'div#'.

Okay, so now that we know how this works, sorta, let's do an exercise. Let's take the 'journal' style, and change the divider lines, and the background color.

- First, go to a gallery and set your viewstyle to 'journal'. (Make sure that you have not forced your viewstyle to something else in the customize gallery page).



- Now, let's use the 'display element information' function on one entire photo plus surroundings. Hold your mouse over an image so the red bounding box shows like below, and then left-click once.

One of my favorite pictures :)

div .journal_entry	
Attributes	
class = journal_entry	
Position	
Left: 273px	Top: 2577px
Width: 750px	Height: 332px
Other	
Font Family: verdana,tahoma,Helvetica,Arial,sans-serif	
Font Size: 12px	
Ancestors	
html	
body .journal_galleryPage gallery_1426716 category_Photography	
subcategory_Pro_Work bodyColor_Black	
div #bodyWrapper	
div #journal .journal_galleryPage gallery_1426716 category_Photography	
subcategory_Pro_Work bodyColor_Black	
Children	
div .photo_left	
div #caption96678156 .caption	
div .spacer	

You'll see that the element that we've selected is classed as 'journal_entry'.

- Now, open the Edit CSS window, and let's give the journal entry a gray background, and a red border.
- Enter the following in the CSS window:

```
.journal_entry {
    background: gray;
    border: 2px dashed red;
}
```

See the difference? Pretty simple and cool, huhh?
You can close the Edit CSS pane now :-)

REVIEW:

- Using Firefox and the Web Developer Extension can make customizing your site much easier.
- You can identify elements by using the Display Element Information function.
- The id and class appear in the top of the floatie.

Multiple Classes and divs (divs in divs)

OK, we're going to start moving into more advanced territory here, but it will be fine. We'll go nice and slow.

Well, our containers are nice, and we can do cool stuff with them, but what if we want to have a container in a container?

One example of where this can come in handy, is to have one container that contains all the rest of the page. That way we can set the background portion of everything that is seen on the page with one command, instead of addressing all the individual containers.

If `<div id="container1"></div>` creates a container, then `<div id="container1"><div id="container2"></div></div>` creates a container within a container. In this case, "container2" is within "container1".

"container2" is a **child** of "container1".

Since linebreaks have no effect on code,

`<div id="container1"><div id="container2"></div></div>` is the same as:

```
<div id="container1">
  <div id="container2"></div>
</div>
```

This has no effect on the behavior, but may make it a lot easier to look at, and understand what you have written.

Once you start using a lot of containers, it will become a mess quickly, and troubleshooting is a lot easier if you have a better overview of your code.

Consider the difference between this:

```
<div id="container1"><div id="container2"></div><div id="container3"></div><div
id="container4"></div><div id="container5"></div><div id="container6"></div></div>
```

and this:

```
<div id="container1">
  <div id="container2"></div>
  <div id="container3"></div>
  <div id="container4"></div>
  <div id="container5"></div>
  <div id="container6"></div>
</div>
```

Let's imagine the following:

```
<div id="container1">
  <div class="example"></div>
  <div class="example"></div>
  <div class="example"></div>
  <div class="example"></div>
  <div class="example"></div>
</div>

<div id="container2">
  <div class="example"></div>
  <div class="example"></div>
  <div class="example"></div>
  <div class="example"></div>
  <div class="example"></div>
</div>
```

Remember the difference between an id and a class? In the above example some of the divs have a class, and some have an id. The two with the ids are addressed using `#container1 { }` and `#container2 { }`. The ten divs classed 'example' are addressed all at once, by `.example { }`.

These are 5 containers classed "example" inside of a container with the id "container1", and 5 containers inside a container with the id "container2".

If we want to set all ten divs (not container1 and container2) to be 100 pixels by 100 pixels and have a red background, we could use this CSS:

```
.example {
  margin: 100px;
  width: 100px;
  background: red;
}
```

We will set all 10 divs (not container1 and container2) to be 100 by 100 pixels, with a red background. But what if we only want to have the 5 divs in container1 to be that size and color?

In that case, you can use the following.

```
#container1 .example {
  margin: 100px;
  width: 100px;
  background: red;
}
```

It will read your elements from left to right. First, it looks for the id container1, then it will look for the class example. The elements classed example inside of container2 will not be touched.

Text & Font Properties

Styling text is done in a similar way as styling other elements.

In this example, we are going to add a copyright statement near the bottom of your pages.

You start with a div: `<div id="my_footer_note"></div>`. The name "my_footer_note" is a personal choice. It usually makes sense to give the element an id that somehow describes its function and/or location, though.

The text goes inside of the element, like this:

```
<div id="my_footer_note">All photos are mine! Don't use them illegally please.</div>
```

Considering that you want this text to show below all your photos, where would you put this code?

That's right, in the box labeled "footer" on the customize page.

Now that we have created the div in HTML, it's time to style the footer using CSS. Addressing the text is done in the same way we've been addressing elements so far.

This code will change the text into bold text:

```
#my_footer_note {  
  font-weight: bold;  
}
```

now you also want the text to show up nicely in the middle, I'm sure. This is what we could use:

```
my_footer_note {  
  font-weight: bold;  
  text-align: center;  
}
```

font-family

font-family is a property that specifies the font.

It's important to realize that the CSS specifies the font. The browser that is viewing the page, will read the CSS and say "Hey, whoever wrote this, wants to use font X". Then, it will go find the font in the list of fonts that are installed on that computer and use that.

What happens if the font that you've specified isn't on the computer that is displaying your page? The computer will replace your font with a default font. The default font may be different on different computers. This can result in unexpected changes in the way your page is displayed, and they may not look as you want them to.

Solution: Choose a font that is installed on all or most computers. See <http://typetester.maratz.com> for a list of 'safe' fonts and examples.

Another thing you can do, is specify multiple fonts. The first font will be used, unless it is not available. It will then use the second font specified, and so on.

Example:

```
#my_footer_note {  
  font-family: Arial, Courier, Times;  
}
```

The font will be Arial, unless that is not available at the computer that is used to view the page. Then it will display the Courier font. If that font is also not available it will display the Times font.

Here are some more examples of what's possible. See appendix A for the complete list.

font-size

specifies the size of your font. The size can be given in a variety of different units.

Example:

```
#my_footer_note {  
  font-size: 18px;  
}
```

font-style

sets the style of the font. Possible values are "normal", "italic" and "oblique".

Example:

```
#my_footer_note {  
  font-style: italic;  
}
```

font-weight

sets the weight of the font. Possible values are "bold", "normal", "lighter" and others.

Example:

```
#my_footer_note {  
  font-weight: bold;  
}
```

text-decoration

can add some cool decoration to your text. Possible values are "none", "underline", "overline", "line-through" and "blink".

Example:

```
#my_footer_note {  
  text-decoration: underline;  
}
```

So, if you want your footer note to look something like this:

All photos are mine! Don't use them illegally please.

you could use:

```
#my_footer_note {  
  font-family: Verdana, Arial;  
  font-weight: bold;  
  text-decoration: underline;  
  font-style: italic;  
  text-align: center;  
}
```

General Description of the SmugMug Pages

Okay, so what does the SmugMug layout look like?

body classes

First, the bodies of all the pages are classed with the name of the style/type page. For instance, the homepage has a class called "homepage". You can use that, to address elements on the homepage, even though those elements may exist on other pages as well.

For instance, to address the header on the homepage only:

```
.homepage #header {  
  
}
```

Here is a list of body classes that SmugMug has already defined:

- homepage
- category
- subcategory
- galleryPage
- smugmug
- smugmug_small
- allthumbs
- traditional
- journal
- slideshow
- filmstrip
- critique
- popularGallery
- datePage
- keywordPage (keyword_loggedIn)
- passwordPage (userPassword & galleryPassword)

loggedIn

SmugMug also gives the class "loggedIn" to all pages when you are logged in. This allows you to style things differently when you are logged in as the account owner, as opposed to when you are logged out. Example:

```
#header {  
  
}
```

to address the element 'header' all the time, versus

```
.loggedIn #header {  
  
}
```

which addresses the element 'header' when you are logged in only.

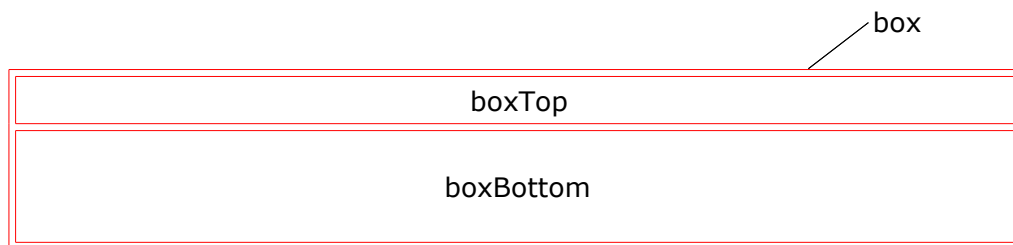
Each page has the same default layout. There is one div that contains all the rest of that page. It has the id 'bodyWrapper'.

Inside the bodywrapper there are 3 divs that contain the rest of the page. From top to bottom:

- header_wrapper
- an element with the id of that page (for instance homepage, smugmug, allthumbs, etc)
- footer_wrapper

BE CAREFUL: in a gallery page set to 'journal' style for instance, there is the body class journal, addressed with `.journal { }`. There is also an element with the id 'journal', addressed with `#journal { }`. This last one is the element that sits between the header_wrapper and the footer_wrapper and contains all the photos.

There are three divs that are used frequently throughout SmugMug: **box**, **boxTop** and **boxBottom**.



These are used on the homepage for most boxes, on the gallery/category pages, and for the comments.

The box contains both the boxTop and the boxBottom. The boxTop usually holds a title, or text indicating what the contents in the boxBottom is.

Clean Code and Comments

The more CSS you have, the more confusing it gets... "where is this?", "why did I do this again?". Before you know it you'll have lots of CSS in your stylesheet, and it will get confusing quickly.

COMMENTS

The first thing you can do to make it easy for yourself is to use comments. Comments will not be read as instructions by your browser, and you can write down pretty much anything you want within a comment. Comments are made between a slash & star and a star & slash. Example:

```
/* this is a comment */
```

Be careful, it has to be in that order.

```
/* this is NOT a comment /*  
*/ this is NOT a comment /*  
// this is also NOT a comment
```

NEVER put comments in comments, as it's asking for trouble. Consider this:

```
/* I want to temporarily comment out this code:  
#header {  
  height: 35px;  
}  
  

```

The idea was to remove the whole code by using comments, however, the star & slash right after 'footer code' closed the comment, and the #footer code will be read as normal. There will also be a star & slash at the end which your browser does not know what to do with. The green will be read as comment, like intended, the red portion will be read as code:

```
/* I want to temporarily comment out this code:  
#header {  
  height: 35px;  
}  
  
/* this is my footer code */  
#footer {  
  color: #ffffff;  
}  
  
*/
```

ORGANIZATION

Another thing you can do, is organization, preferably right from the start.

The best thing to do, code-wise, is to start with the most general items, and work your way to the more specific items. For instance, the background is going to apply to all your pages, but the code that you use to assign a border-color to "button x" may only apply to one button on a single page. I usually also work from top to bottom. First, the header, then the middle section, followed by the footer.

There are many different ways to organize your code, and the code below is merely an example of what is possible. The CSS is just for demonstration purposes. The most important thing in deciding how to organize is that you make sure that you know where to find the code that you are looking for. Here is an example. Pay attention to the way it is organized.

```
/* --- General --- */
body {
  background: #999999;
}

/* --- Header --- */
#header_wrapper {
  width: 750px;
  margin: 0 auto;
}

/* --- Homepage --- */
#bioBox .photo {
  display: none;
}

/* --- Viewstyle specific --- */
/* --- (smugmug/journal/filmstrip/etc) --- */
.journal_entry {
  width: 750px;
  border-bottom: 2px solid #ff0000;
}

#traditional {
  background: #000 url(http://www.smugmug.com/photos/1234567-O.jpg) repeat;
}

/* --- Special pages --- */
/* --- (keyword/popular/date) --- */
.popular .cartbuttons {
  display: none;
}

/* --- Footer --- */
#footer {
  color: black;
}
```

OTHER WAYS TO CLEAN UP

You can also clean the code itself. For instance, telling multiple elements to do the same thing can be simplified:

```
#header {
  display: none;
}

.homepage #breadcrumb {
  display: none;
}

#feeds {
  display: none;
}
```

Can be consolidated into:

```
#header,
.homepage #breadcrumb,
#feeds {
  display: none;
}
```

Note that when multiple selectors are listed, they are separated by commas.

The layout of the code can be adapted to personal taste as well. CSS doesn't care about line breaks, so:

```
.homepage .boxTop {
  height: 28px;
  background: #336699;
}
```

is the same as:

```
.homepage .boxTop { height: 28px; background: #336699; }
```

is the same as:

```
.homepage .boxTop {
  height: 28px; background: #336699;
}
```

is the same as:

```
.homepage .boxTop
{
  height: 28px;
  background: #336699;
}
```

But which one is the easiest to read? Your browser may not care about the line breaks, but people do. Make sure you choose something that makes sense to you, and use it consistently.

REVIEW:

- Comments in CSS are made between `/*` and `*/`
- Organize your code so that it make sense to you.
- You can use comments to help organize your code.
- Format your code so that it's easy for you to read.
- Be consistent.

Problems, What To Do?

aaaaaargggghhh!!!!!!

SMUGMUG SPECIFIC PROBLEMS

My code doesn't do anything, what gives?

When customizing your site, you may need to change some settings found in the 'customize gallery' page*.

- Appearance: You can choose if you want this gallery to have your custom look, or the SmugMug default look. If you want your custom look, "custom" should be selected.
- Hide Owner: When Hide Owner is selected, all traces to you, as account owner, are removed. This includes the breadcrumbs that can lead back to your homepage, and also includes the custom CSS you have been working on. Hide owner should be set to "no" in order to show your custom CSS.
- Community: When a gallery is associated with a community, the choices of layout can be SmugMug or Community. SmugMug will give you the SmugMug default look, Community will give you the look the owner has used for the community. Either way, your custom CSS will not show. There should be no community selected in order for your custom CSS to show.

* If you want to change the settings, first make sure you are logged in. Then, go to the gallery you want to change. Look in the 'gallery tools' menu for the 'customize gallery' option. You can also click on the 'customize' button, below the gallery thumbnail.

CSS RELATED PROBLEMS

So the code does not do what you think it should do - what now? Don't worry, we'll figure it out.

One of the most common causes of code not working, or not working correctly, is related to the fact that CSS can be reset. Consider the following code:

```
#breadcrumb {  
  display: none;  
}  
  
#footer {  
  color: blue;  
}
```

Let's say you forgot about setting the footer color to blue, and you decide to set the footer color to red like this:

```
#footer {  
    color: red;  
}  
  
#breadcrumb {  
    display: none;  
}  
  
#footer {  
    color: blue;  
}
```

even though you set the footer color to red initially, later in the CSS it is set to blue. If you want to set the color to red, it should be:

```
#breadcrumb {  
    display: none;  
}  
  
#footer {  
    color: red;  
}
```

OKAY, I'VE TRIED IT ALL, BUT HAVE NO CLUE WHY THAT PROBLEM IS STILL THERE!

If you are really at a loss, you can relatively simply figure out what is causing the problem. Open the page with the problem in Firefox, and then open the 'edit CSS' panel from the Web developer extension.

Take out the first half of your code, by selecting it, and 'cutting' it. (ctrl+x on windows, cmd+x on mac). If the problems is gone, you now know that the problem is most likely in the code you removed.

Put it back, and take out the first half of the code you just put back. If that removes the problem, you know that it is in that part of the code.

If it does not solve the problem, it is in the other half of the code.

You can keep taking out parts of the code, until you find the code that is causing you problems.

Advanced: Text Formatting & span

Formatting text is nice, and you know how to do that by now, but do you know how to make something like this:

*All photos are **mine!** **Don't use them illegally please.***

The CSS that we have used before, addresses the whole text, so how do we address parts of we text? We can use another element for this. Introducing the element: **span**.

A span is an element, just like div. It is used to group elements and style them inline.

We will 'span' parts of existing code with the span-tag, like so:

```
<div id="my_footer_note">All photos are <span class="red_text">mine</span>!<span class="my_bold_text"> Don't use them <span class="red_text">illegally</span> please.</span></div>
```

Confusing? this is the same code but a different layout:

```
<div id="my_footer_note">All photos are  
    <span class="red_text">mine!</span>  
    <span class="my_bold_text">Don't use them  
        <span class="red_text">illegally please.</span>  
    </span>  
</div>
```

Still confused? maybe this will help:

*all photos are **mine!** **don't use them illegally please.***

id: my_footer_note

id: my_footer_note & class: red_text

id: my_footer_note & class: my_bold_text

id: my_footer_note & class: my_bold_text & class: red_text

Since all text has the id "my_footer_note", we can set the font-family, create an underline and make the text italic for that id.

```
#my_footer_note {  
  font-family: Verdana, Arial;  
  text-decoration: underline;  
  font-style: italic;  
  text-align: center;  
}
```

We only want the text that is classed "red_text" to be red, so we'll add this:

```
.red_text {  
  color: red;  
}
```

Now we only need to bold the text that has the the class "my_bold_text", so add:

```
.my_bold_text {  
  font-weight: bold;  
}
```

That's it, mission accomplished!

Index

Ancestor.....	17	HTML.....	5
background.....	13	id.....	5
body.....	5	internet page.....	5
box.....	25	loggedIn.....	24
boxBottom.....	25	margin.....	13
boxTop.....	25	shorthand.....	14
Cascading Style Sheet.....	8	period.....	8
child.....	19	pound-sign.....	8
Children.....	17	property.....	8
class.....	5	semi-colon.....	8
CSS.....	8	span.....	32
display.....	15	text-decoration.....	22
div.....	5	value.....	8
elements.....	5	Web Developer Extension.....	10
Firefox.....	10	Display Element Information.....	16
font-family.....	21	Edit CSS.....	10
font-size.....	22	webdev.....	10
font-style.....	22	8
font-weight	22	8
head.....	5	8
		8

APPENDIX A - CSS Properties and Values

Property	Description	Values
background	A shorthand property for setting all background properties	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed
background-color	Sets the background color of an element	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent
background-image	Sets an image as the background	url(URL) none
background-position	Sets the starting position of a background image	top left top center top right center left center center center right bottom left bottom center bottom right <i>x% y%</i> <i>xpos ypos</i>
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat
border	A shorthand property for setting all of the properties for the four borders	<i>border-width</i> <i>border-style</i> <i>border-color</i>
border-bottom	A shorthand property for setting all of the properties for the bottom border	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>
border-bottom-color	Sets the color of the bottom border	<i>border-color</i>
border-bottom-style	Sets the style of the bottom border	<i>border-style</i>
border-bottom-width	Sets the width of the bottom border	thin medium thick <i>length</i>
border-collapse	Sets the border model of a table	collapse separate
border-color	Sets the color of the four borders, can have from one to four colors	<i>color</i>

border-left	A shorthand property for setting all of the properties for the left border	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>
border-left-color	Sets the color of the left border	<i>border-color</i>
border-left-style	Sets the style of the left border	<i>border-style</i>
border-left-width	Sets the width of the left border	thin medium thick <i>length</i>
border-right	A shorthand property for setting all of the properties for the right border	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>
border-right-color	Sets the color of the right border	<i>border-color</i>
border-right-style	Sets the style of the right border	<i>border-style</i>
border-right-width	Sets the width of the right border	thin medium thick <i>length</i>
border-spacing	Sets the distance between the borders of adjacent cells (only for the "separated borders" model)	<i>length length</i>
border-style	Sets the style of the four borders, can have from one to four styles	none hidden dotted dashed solid double groove ridge inset outset
border-top	A shorthand property for setting all of the properties for the top border	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>
border-top-color	Sets the color of the top border	<i>border-color</i>
border-top-style	Sets the style of the top border	<i>border-style</i>
border-top-width	Sets the width of the top border	thin medium thick <i>length</i>
border-width	A shorthand property for setting the width of the four borders , can have from one to four values	thin medium thick <i>length</i>
bottom	Sets how far the bottom edge of an element is above/below the bottom edge of the parent element	auto % <i>length</i>
caption-side	Sets the position of the caption according to the table	top bottom left right

clear	Sets the sides of an element where other floating elements are not allowed	left right both none
clip	Sets the shape of an element. The element is clipped into this shape, and displayed	<i>shape</i> auto
color	Sets the color of a text	<i>color</i>
content	Generates content in a document. Used with the :before and :after pseudo-elements	<i>string</i> <i>url</i> <i>counter(name)</i> <i>counter(name, list-style-type)</i> <i>counters(name, string)</i> <i>counters(name, string, list-style-type)</i> <i>attr(X)</i> open-quote close-quote no-open-quote no-close-quote
counter-increment	Sets how much the counter increments on each occurrence of a selector	none <i>identifier number</i>
counter-reset	Sets the value the counter is set to on each occurrence of a selector	none <i>identifier number</i>
cursor	Specifies the type of cursor to be displayed	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help
direction	Sets the text direction	ltr rtl

display	Sets how/if an element is displayed	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption
empty-cells	Sets whether cells with no visible content should have borders or not (only for the "separated borders" model)	show hide
float	Sets where an image or a text will appear in another element	left right none
font	A shorthand property for setting all of the properties for a font	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar
font-family	A prioritized list of font family names and/or generic family names for an element	<i>family-name</i> <i>generic-family</i>
font-size	Sets the size of a font	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> %
font-size-adjust	Specifies an aspect value for an element that will preserve the x-height of the first-choice font	none <i>number</i>

font-stretch	Condenses or expands the current font-family	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded
font-style	Sets the style of the font	normal italic oblique
font-variant	Displays text in a small-caps font or a normal font	normal small-caps
font-weight	Sets the weight of a font	normal bold bolder lighter 100 200 300 400 500 600 700 800 900
height	Sets the height of an element	auto <i>length</i> %
left	Sets how far the left edge of an element is to the right/left of the left edge of the parent element	auto % <i>length</i>
letter-spacing	Increase or decrease the space between characters	normal <i>length</i>
line-height	Sets the distance between lines	normal <i>number</i> <i>length</i> %
list-style	A shorthand property for setting all of the properties for a list	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>
list-style-image	Sets an image as the list-item marker	none <i>url</i>
list-style-position	Sets where the list-item marker is placed in the list	inside outside

list-style-type	Sets the type of the list-item marker	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha
margin	A shorthand property for setting the margin properties	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>
margin-bottom	Sets the bottom margin of an element	auto <i>length</i> %
margin-left	Sets the left margin of an element	auto <i>length</i> %
margin-right	Sets the right margin of an element	auto <i>length</i> %
margin-top	Sets the top margin of an element	auto <i>length</i> %
marker-offset		auto <i>length</i>
max-height	Sets the maximum height of an element	none <i>length</i> %
max-width	Sets the maximum width of an element	none <i>length</i> %
min-height	Sets the minimum height of an element	<i>length</i> %
min-width	Sets the minimum width of an element	<i>length</i> %
outline	A shorthand property for setting all the outline properties	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i>
outline-color	Sets the color of the outline around an element	<i>color</i> invert

outline-style	Sets the style of the outline around an element	none dotted dashed solid double groove ridge inset outset
outline-width	Sets the width of the outline around an element	thin medium thick <i>length</i>
overflow	Sets what happens if the content of an element overflow its area	visible hidden scroll auto
padding	A shorthand property for setting all of the padding properties	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>
padding-bottom	Sets the bottom padding of an element	<i>length</i> %
padding-left	Sets the left padding of an element	<i>length</i> %
padding-right	Sets the right padding of an element	<i>length</i> %
padding-top	Sets the top padding of an element	<i>length</i> %
position	Sets an element to have a static, relative, absolute or fixed position	static relative absolute fixed
quotes	Sets the type of quotation marks	none <i>string string</i>
right	Sets how far the right edge of an element is to the left/right of the right edge of the parent element	auto % <i>length</i>
table-layout	Sets the algorithm used to lay out the table	auto fixed
text-align	Sets the alignment the text in an element	left right center justify
text-decoration	Adds decoration to text	none underline overline line-through blink
text-indent	Indents the first line of text in an element	<i>length</i> %

text-shadow		none <i>color</i> <i>length</i>
text-transform	Controls the letters in an element	none capitalize uppercase lowercase
top	Sets how far the top edge of an element is above/below the top edge of the parent element	auto % <i>length</i>
unicode-bidi		normal embed bidi-override
vertical-align	Sets the vertical alignment of an element	baseline sub super top text-top middle bottom text-bottom <i>length</i> %
visibility	Sets if an element should be visible or invisible	visible hidden collapse
white-space	Sets how white space inside an element is handled	normal pre nowrap
width	Sets the width of an element	auto % <i>length</i>
word-spacing	Increase or decrease the space between words	normal <i>length</i>
z-index	Sets the stack order of an element	auto <i>number</i>

APPENDIX B - Colors

Color values are entered by name, or by RGB value. There are 16 names that can be used to indicate colors. For other colors the RGB value must be used.

RGB values are to be set in the (shorthand) Hexadecimal numerical system, actual RGB values, or RGB percentages.

```
body { color: red; }  
body { color: #f00; }  
body { color: #ff0000; }  
body { color: rgb(255, 0, 0); }  
body { color: rgb(100%, 0%, 0%); }
```

All of the above code does the same.

Accepted Color names

Name	HEX code
aqua	#000080
black	#000000
blue	#0000ff
fuchsia	#ff00ff
gray	#808080
green	#008000
lime	#00ff00
maroon	#800000
navy	#000080
olive	#808000
purple	#800080
red	#ff0000
silver	#c0c0c0
teal	#008080
white	#ffffff
yellow	#ffff00

Some useful links:

- Web Color Chart - <http://html-color-codes.com/>
- The hex hub - <http://www.december.com/html/spec/color.html>
- Color code chart - <http://www.computerhope.com/htmcolor.htm#03>
- Color code combination chooser - http://www.siteprocentral.com/html_color_code.html

APPENDIX C - CSS Pseudo Classes & Elements

Pseudo-class	Adresses
:active	an activated element
:focus	an element that has focus
:hover	an element when you mouse over it
:link	an unvisited link
:visited	a visited link
:first-child	an element that is the first child of some other element
:lang	Allows the author to specify a language to use in a specified element

Pseudo-element	Adresses
:after	Inserts content after an element
:before	Inserts content before an element
:first-letter	the first letter of text
:first-line	the first line of text

APPENDIX D - Information and Links

Reference:

- World Wide Web Consortium - <http://www.w3c.org/>
- Wikipedia - CSS - http://en.wikipedia.org/wiki/Cascading_Style_Sheets

Color:

- Web Color Chart - <http://html-color-codes.com/>
- The hex hub - <http://www.december.com/html/spec/color.html>
- Color code chart - <http://www.computerhope.com/htmcolor.htm#03>
- Color code combination chooser - http://www.siteprocentral.com/html_color_code.html

Training and information:

- w3schools - <http://www.w3schools.com>
- maratz – information on fonts - <http://typetester.maratz.com>

SmugMug

- basic customize FAQ - <http://www.smugmug.com/help/customize-faq.mg>
- advanced customize FAQ - <http://www.dgrin.com/showthread.php?t=52811>
- Dgrin customization forum - <http://www.dgrin.com/forumdisplay.php?f=31>
- SmugMug wiki - <http://smugmug.jot.com/WikiHome/CSS>

APPENDIX E - Browser Specific Problems

Browser specific code

W3C is the organization that creates the CSS recommendations. The problem is that they are recommendations, and not all browsers react exactly the same to the same code. Usually this is no problem, but in certain situations some browsers react differently, and show things differently from what you see in an other browser, and what you intended it to look like.

For this, it is possible to write CSS, that is only understood by certain browsers. This is advanced CSS and usually not needed.

To specifically address:

- IE6 and earlier - `_` before property

```
#header {  
  height: 30px;  
  _height: 35px;  
}
```

- IE7 and earlier - `~` before property

```
#header {  
  height: 30px;  
  ~height: 35px;  
}
```

- all but IE - `html>body` before the element/selector

```
#header {  
  height: 35px;  
}  
  
html>body #header {  
  height: 30px;  
}
```

- SAFARI - `#` after semi-colon of property value.

```
#header {  
  height: 35px;  
}  
  
#header {  
  height: 30px;#  
}
```

Note: Safari will ignore the values marked with the pound-sign, and everything that follows.

This list is by no means all inclusive. There are more hacks to address specific browsers which

are not mentioned here for various reasons.

Internet Explorer 6 and transparency.

Internet Explorer 6, can not handle transparency from PNG files. It is possible to use the AlphaImageloader filter. According to Microsoft, the AlphaImageloader filter displays an image within the boundaries of the object and between the object background and content, with options to clip or resize the image. When loading a Portable Network Graphics (PNG) image, transparency from zero to 100 percent is supported.

Example of how that can be used:

```
.searchglass {  
    background: url(IMAGELOCATION) no-repeat;  
    _background: none;  
    _filter: progid:DXImageTransform.Microsoft.AlphaImageLoader(enabled=true,  
sizingmethod=crop ,src='IMAGELOCATION');  
}
```

For more information, see:

<http://msdn.microsoft.com/workshop/author/filter/reference/filters/alphaimageloader.asp>

APPENDIX F - CSS Cheatsheet

CSS (Customizing SmugMug Simplified) - Your guide to a stylin' SmugMug site - CSS CHEATSHEET

Borders	Containers	Paging	Tables	Color / Background
border border-top border-bottom border-right border-left border-color border-top-color border-right-color border-bottom-color border-left-color border-style border-top-style border-right-style border-bottom-style border-left-style border-width border-top-width border-right-width border-bottom-width border-left-width	margin margin-top margin-right margin-bottom margin-left padding padding-top padding-right padding-bottom padding-left Positioning display position top right bottom left float clear z-index direction unicode-bidi overflow clip visibility	size marks page-break-before page-break-after page-break-inside page orphans widows Dimensions width min-width max-width height min-height max-height line-height vertical-align Miscellaneous content counter-reset counter-increment marker-offset list-style list-style-type list-style-image list-style-position quotes	caption-side table-layout border-collapse border-spacing empty-cells speak-header Aural volume speak pause pause-before pause-after cue cue-before cue-after play-during azimuth elevation speech-rate voice-family pitch pitch-range stress richness speak-punctuation speak-numeral	background background-color background-image background-repeat background-attachment background-position color Fonts font font-family font-style font-variant font-weight font-stretch font-size font-size-adjust Text text-indent text-align text-decoration text-shadow letter-spacing word-spacing text-transform white-space
Interface cursor outline outline-width outline-style outline-color				

Units	Color notations	Container layout	CSS Syntax
px Pixels % Percentage em 1em equal to font size of parent (same as 100%) ex Height of lower case "x" in Inches cm Centimeters mm Millimeters pt 1pt = 1/72in pc 1pc = 12pt 0 0 requires no unit	#123abc RGB Hex Notation #12a Equates to #1122aa rgb(0,25,90) Value (0 to 255) of each of red, green, and blue. Can also be percentages	<p>The diagram illustrates the layout of a rectangular container. It shows an outer rectangle representing the total container size, with a double-headed arrow labeled 'margin' indicating the space between the container and its parent. Inside this is a gray-shaded rectangle representing the container's content area, with a double-headed arrow labeled 'padding' indicating the space between the content and the border. Within the content area is a white rectangle representing the 'visible area', with a double-headed arrow labeled 'border' indicating the thickness of the border. The width and height of the visible area are also indicated with double-headed arrows labeled 'width' and 'height' respectively.</p>	<pre>/* Comments */ selector { property: values; }</pre>