Introductory Web Design Series

Build a Website from Start to Finish

by

LearnWebDesignOnline.com

Version 1.2 dated June 11, 2007

In the lessons, you will find links to source code and live pages which will require an internet connection to access.

Table of Contents

Part I: Introduction to Fireworks

Part II: HTML / Dreamweaver

Part III: Introduction to CSS

Introductory Fireworks Lessons

Fireworks is a general graphics program created by Macromedia (before they were bought by Adobe) used by web designers to create design layout for websites among other things.

In these lessons we will be creating the design comp for our website project. You will learn the basic tools and interface of Fireworks as well as performing a basic photo enhancement.

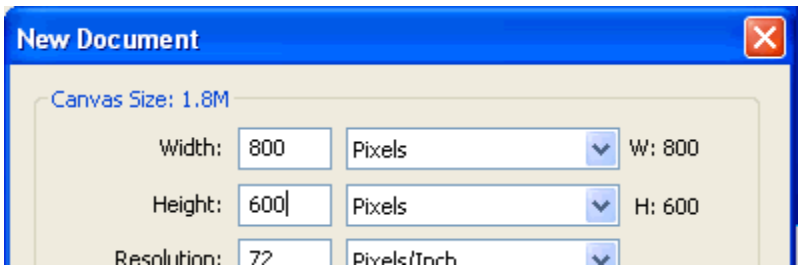
Here is the design comp that we will be creating in these lessons.

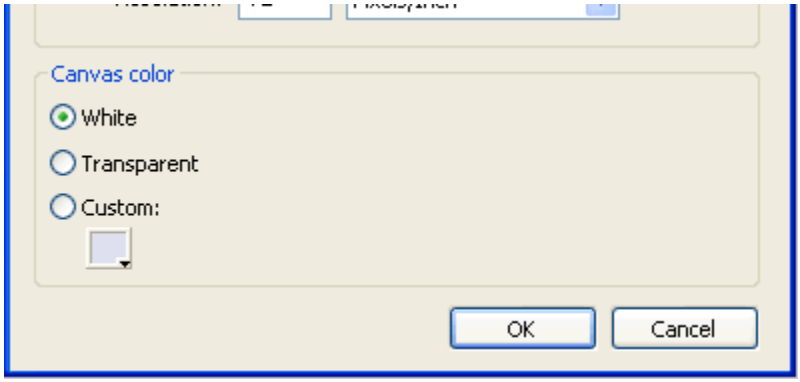
Getting Started with Fireworks

We want to create our CreativeWorks website design layout (also will be referred to as "design comp") for a website that is nicely viewable in a 800x600px screen -- that is 800 pixels wide and 600 pixel high. Designing a website with a width of 800px is fairly standard. There are some sites that are starting to make their widths larger under the assumption that most people have their monitor resolution set to 1024x768 or higher and that the site will be viewable in those resolutions without horizontal scrolling. However, if you go to MSN, yahoo, or google, they all make their width fit within a width of 800px. And we will continue to use this standard.

The first thing that you see when you open up Fireworks is the Welcome screen.

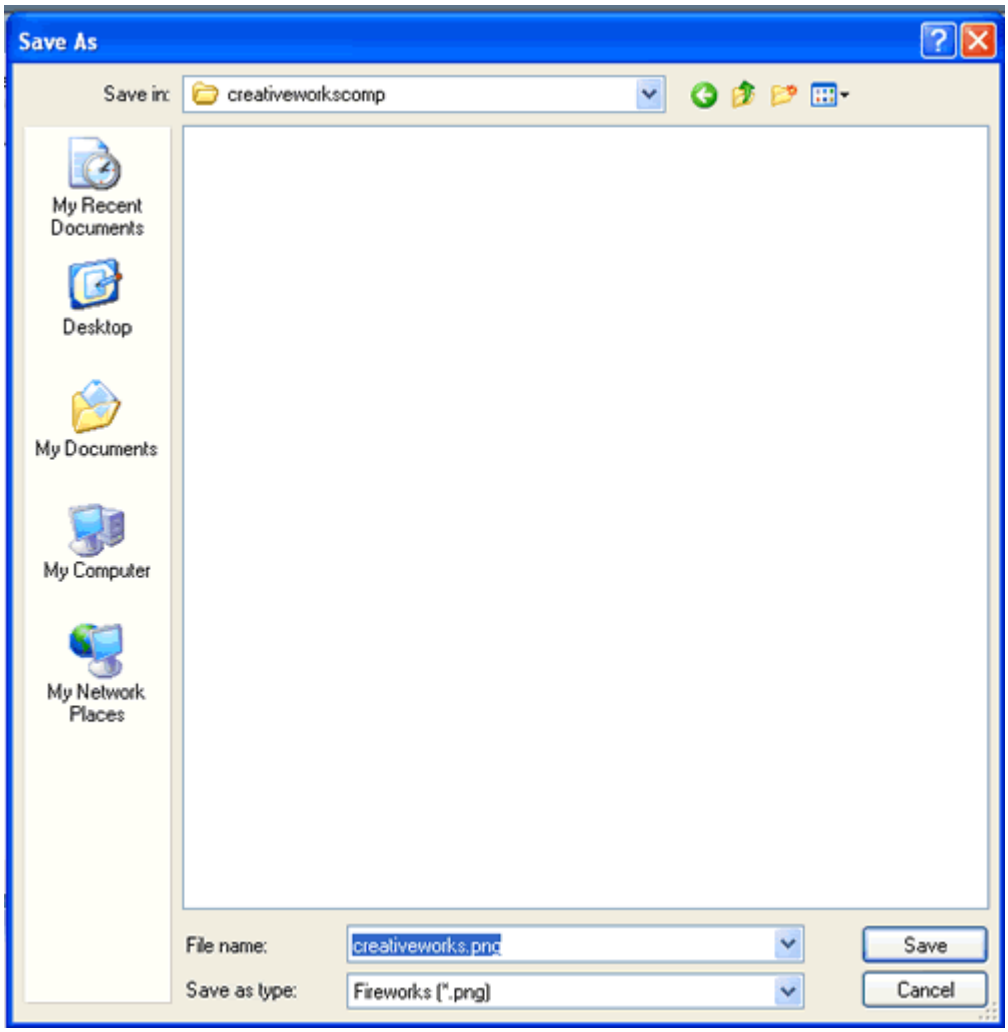
Next, we will create a new file by selecting menu **File -> New**.





Enter the stated dimensions as shown. A resolution of 72 pixels/inch is standard for images that are planned to be viewed on screen. For images that are planned to be printed on brochurs, you will need to set this higher (say 300 pixels/inch). The measure pixels/inch is equivalent to dpi (or dots per inch). For web work, we keep the pixels/inch at 72. Selecting a white canvas is fine for now. Click **Okay**.

Next, we save our document by **File -> Save As**. Save the file with the name **creativeworks.png** in a folder called **creativeworkscomp** as shown. I happen to put mine in **c:\creativeworkscomp** for example.



The next time you want to work on this Fireworks file, simply load Fireworks and open this PNG file. Alternatively, you can find the **creativeworks.png** file in Windows Explorer and double-click on it. Windows will typically launch Macromedia Fireworks for you and load the PNG file.

File Formats

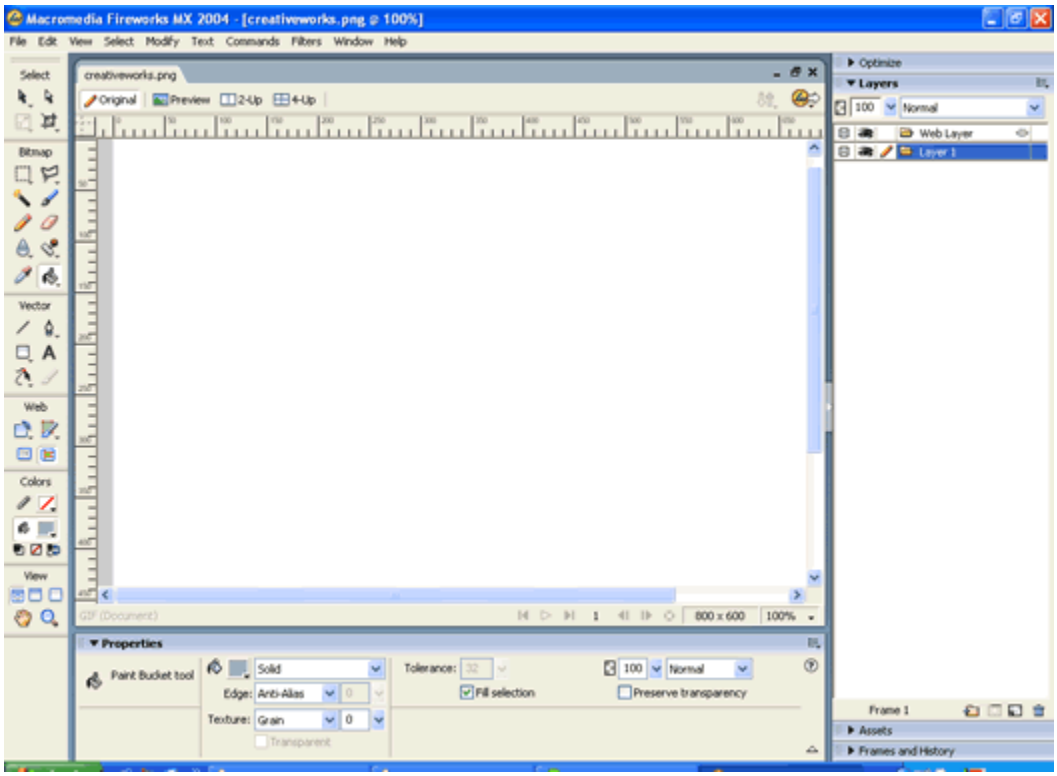
Notice that Firework's native format the the PNG (Portable Network Graphic) format. Traditionally, the two file formats that are used on the web are GIF and JPG (also known as JPEG).

The GIF format good for images containing a lot of solid colors. For example, icons, cartoons, backgrounds, etc. GIF also supports transparency and image animation. JPG is good for photographic images, or images with a lot of gradients and fine details. JPG does not support transparency nor animation. All browsers will be able to display these two types of formats without any problems. And most designers will use their graphics program to export to the GIF and JPG format before placing the image on the web server.

Most modern browsers is now able to display PNG file format without any problem. And you can put images that are in this format directly on the web server without exporting. However, throughout this course, I will still perform the export the traditional way, because I can perform optimizations that can get me smaller file sizes in the GIF and JPG format than in the PNG.

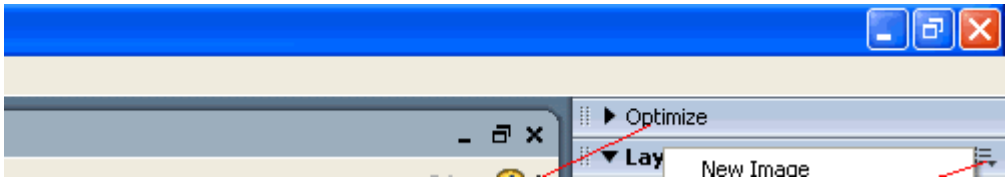
Fireworks Environment

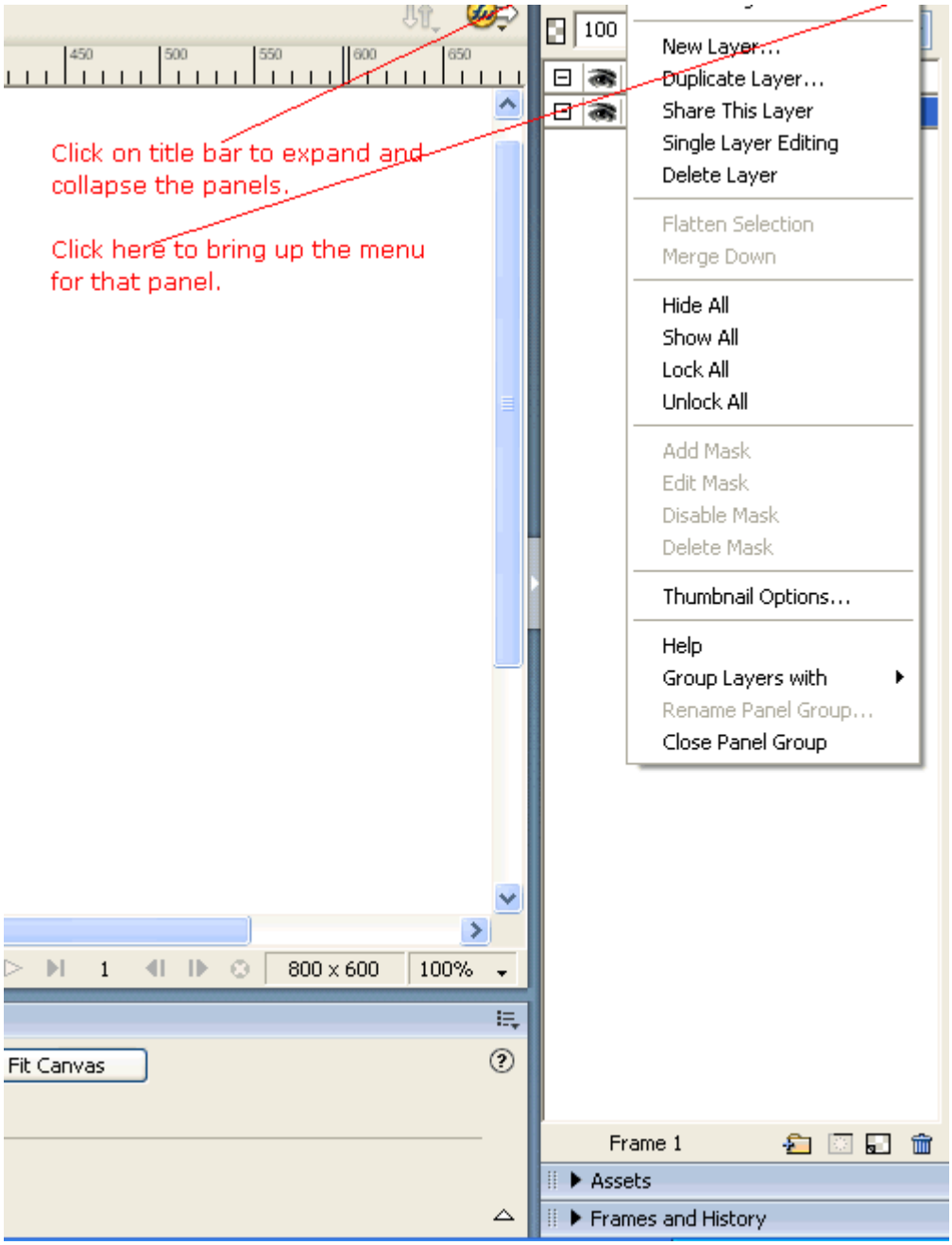
What you see in the middle of Fireworks environment is your canvas.



To the left is the toolbox. If you don't see the toolbox, select **Windows->Tools** from the menu.

To the right are some panels as shown. If you click on the titles of each panel, it will open and collapse the panel. For example, open up the **Layers** panel as shown below. If you do not see this panel, bring it up using menu **Windows-> Layers**. Panels have context menus when you click as shown.

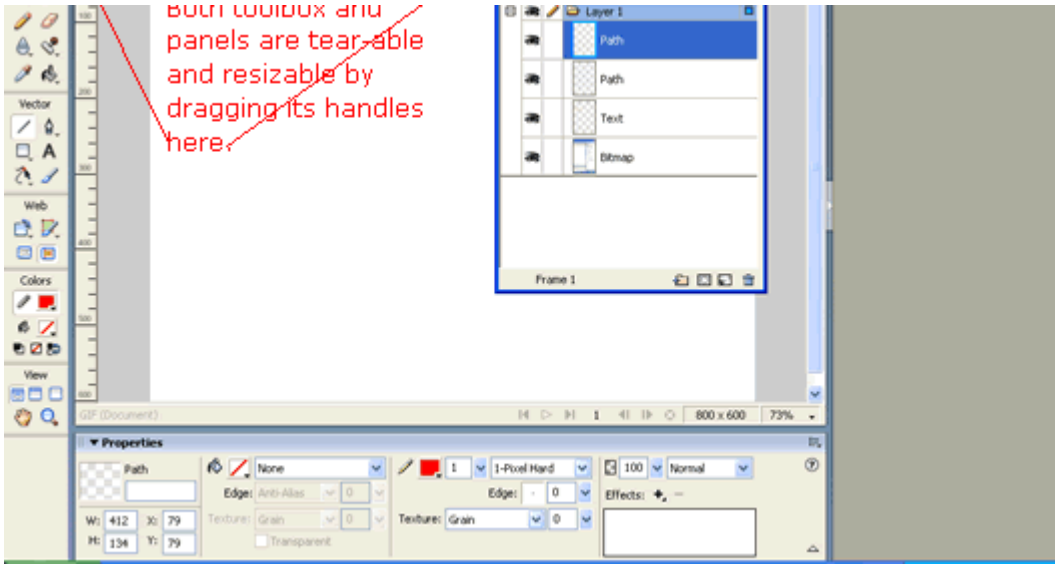




You can close the panel to make it disappear from view by selecting **Close Panel Group** from the context menu. Bring it back with the menu command as described above.

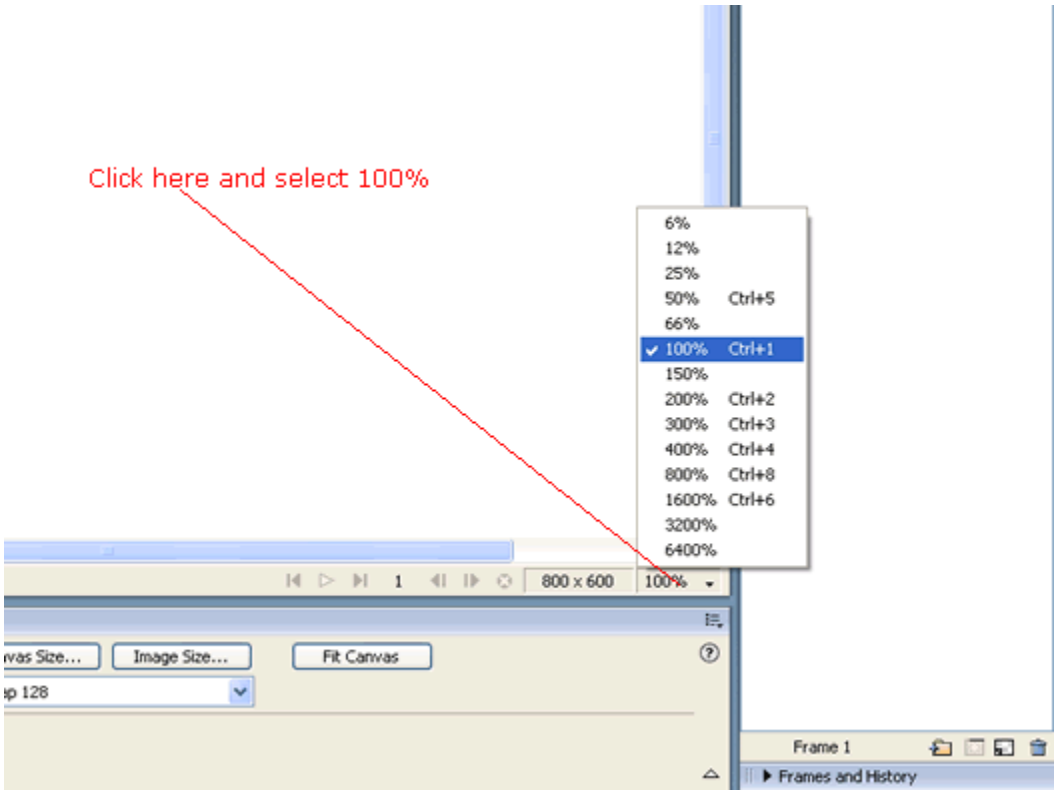
One of the more important panels is the **Properties** panel which you can bring up with **Windows->Properties** or **Ctrl-F3**. This panel will look different depending on which object you have selected. We will talk more about this panel later.

Both the toolbox and panels can be torn from its dock by dragging its handles. Once detached, they can be resized as well. They can be docked back in place, but you might have to give it several tries.

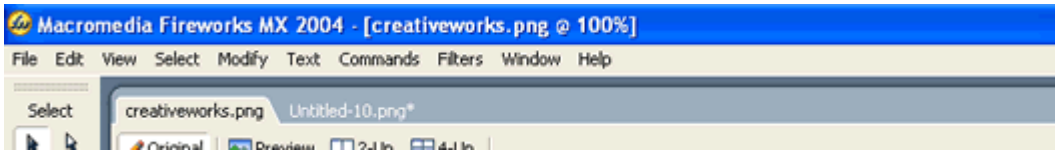


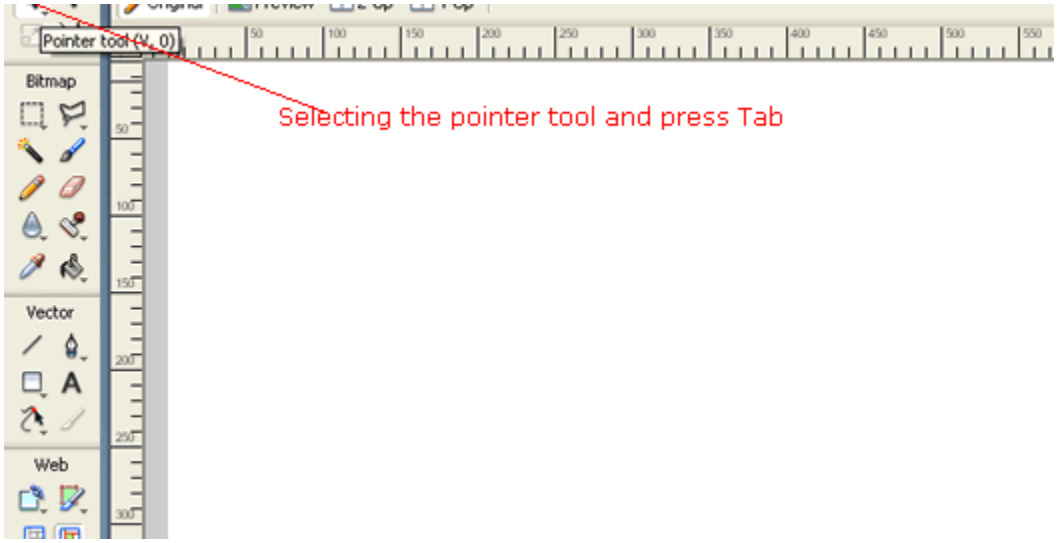
100% Scale

When designing, we often need to see how our art looks at 100% scale, or in actual size. Make sure our document is in 100% Scale by **Ctrl-1** or selecting as shown.



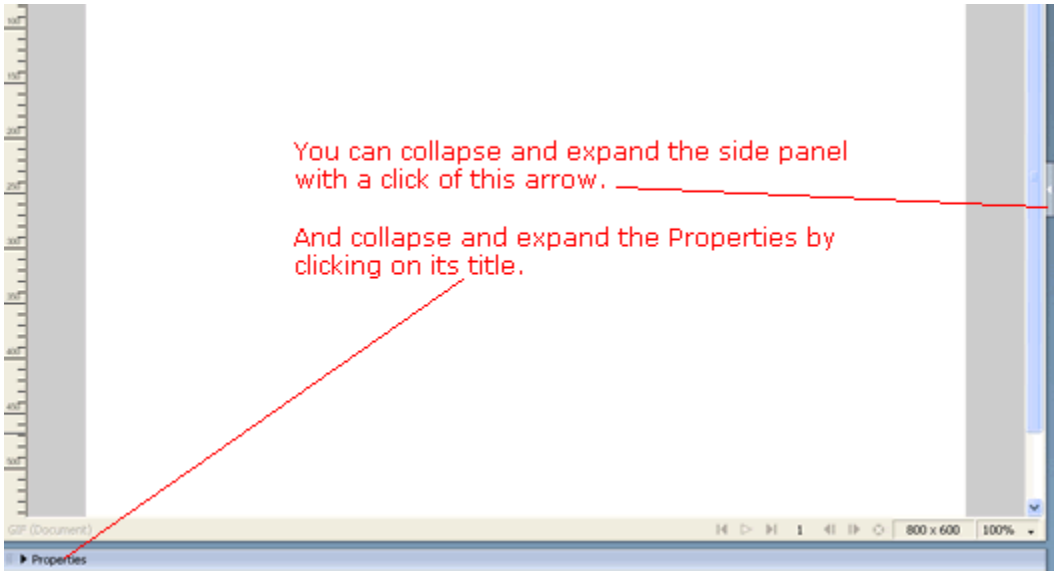
When we do this, you might find you can not see the entire canvas because there are too many panels in the way. Plus those ugly panels detract from your art anyways. Get rid of all those panels and toolboxes by selecting the **pointer tool** and then press **Tab**.





Press **Tab** again and your panels and toolbox comes back.

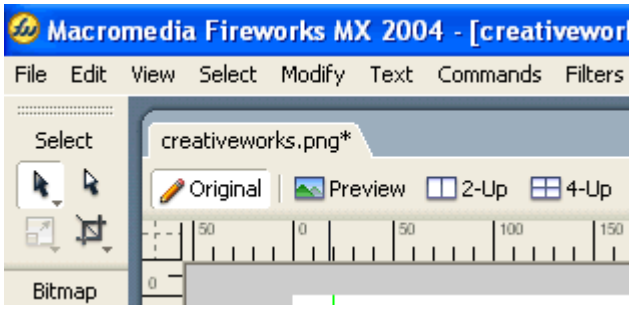
Alternatively, you can get more canvas space by collapsing them...

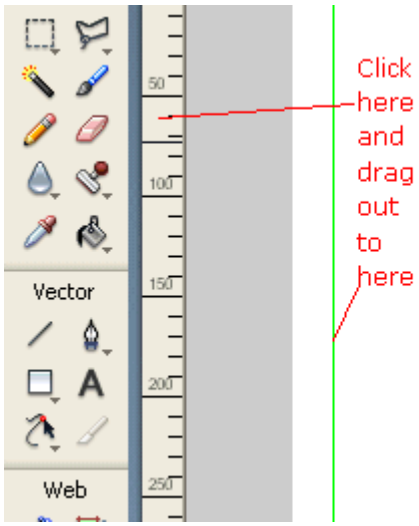


Setting Guides in Fireworks

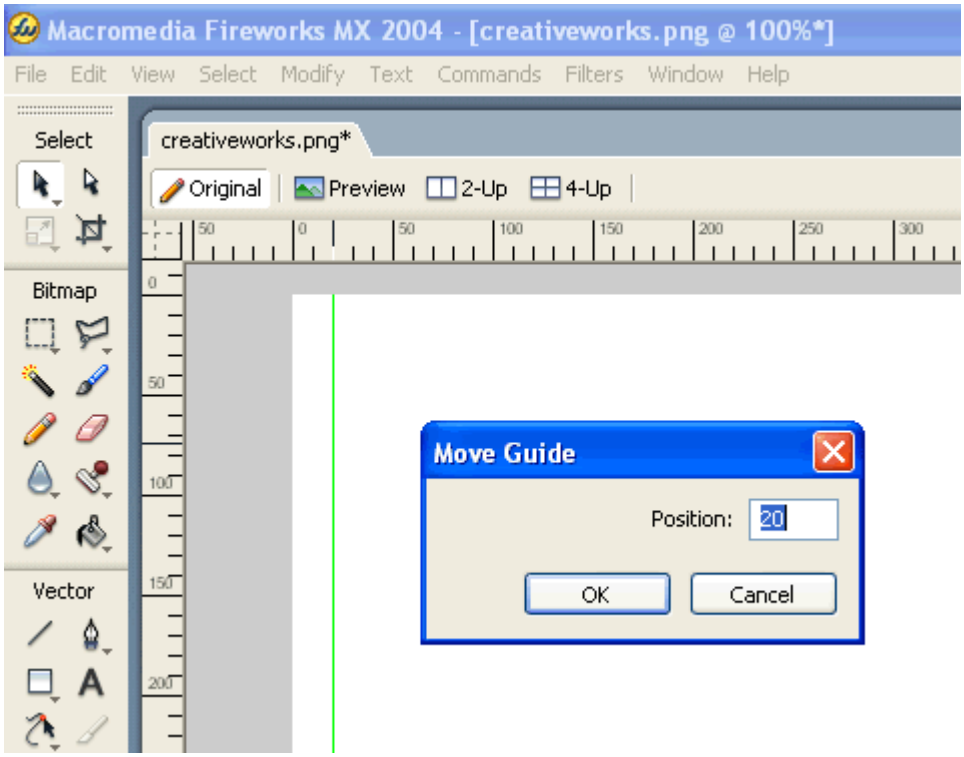
Although our canvas is 800px wide, our web page is going to 760px wide so we have 20px on each side for some air and things like browser frames. Let's have 10 pixel top margin also (just for some air).

We need precision, so drag out a vertical guide to about 20px according to the ruler. If you don't see the ruler, do **View -> Rulers**.



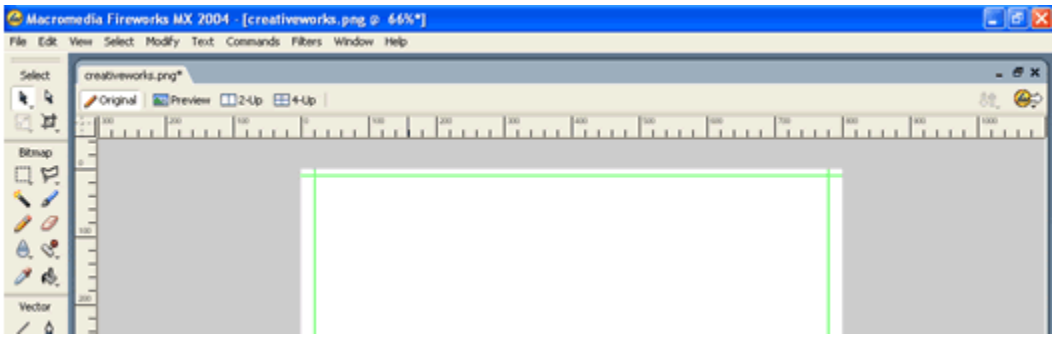


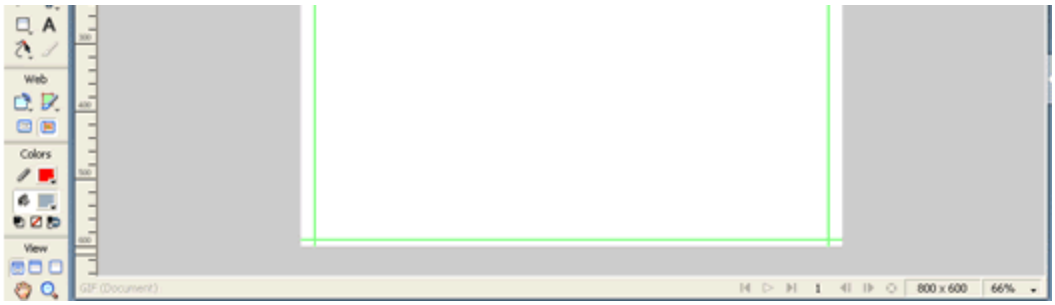
No need to drag it precisely, because with the pointer tool selected, we double-click on the green guide.



And set its position to 20.

In a similar manner. Set the other vertical guide to 780. And set two horizontal guides -- one at 10 and one at 590.



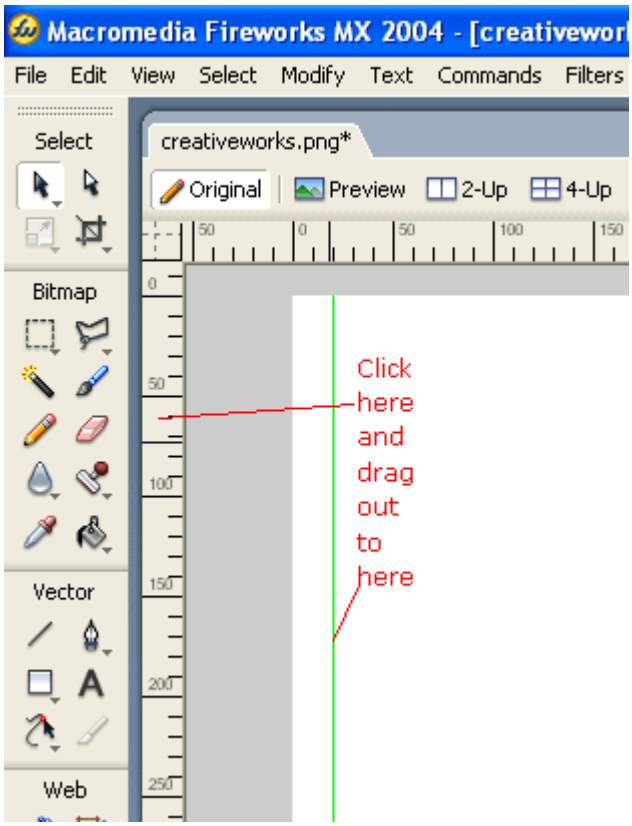


I have set the scaling to 66% so you can see the whole canvas.

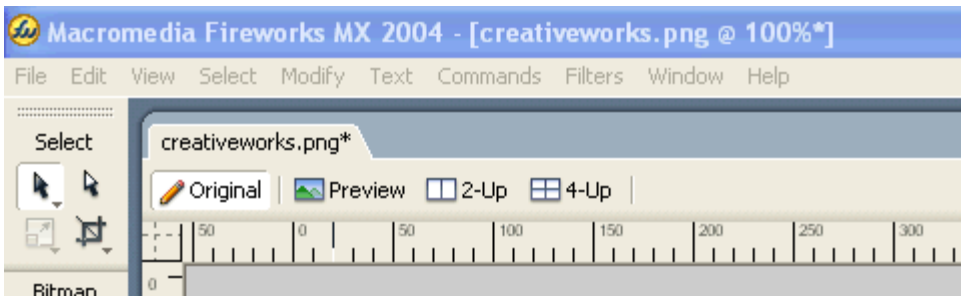
Setting Guides in Fireworks

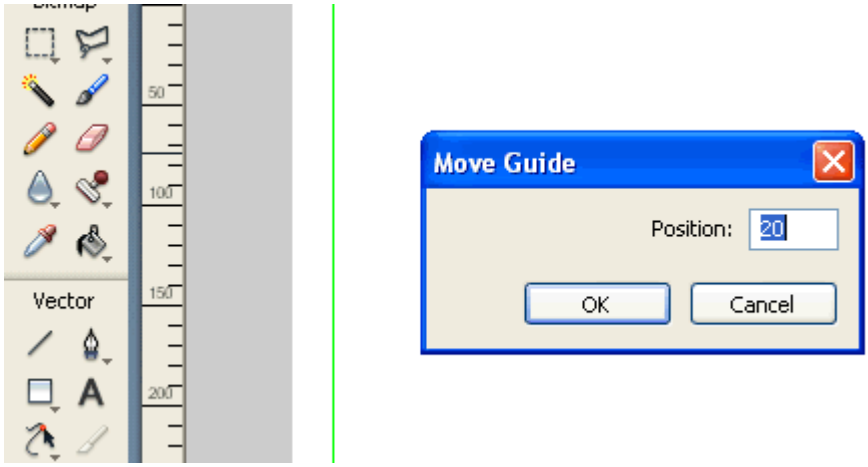
Although our canvas is 800px wide, our web page is going to 760px wide so we have 20px on each side for some air and things like browser frames. Let's have 10 pixel top margin also (just for some air).

We need precision, so drag out a vertical guide to about 20px according to the ruler. If you don't see the ruler, do **View -> Rulers**.



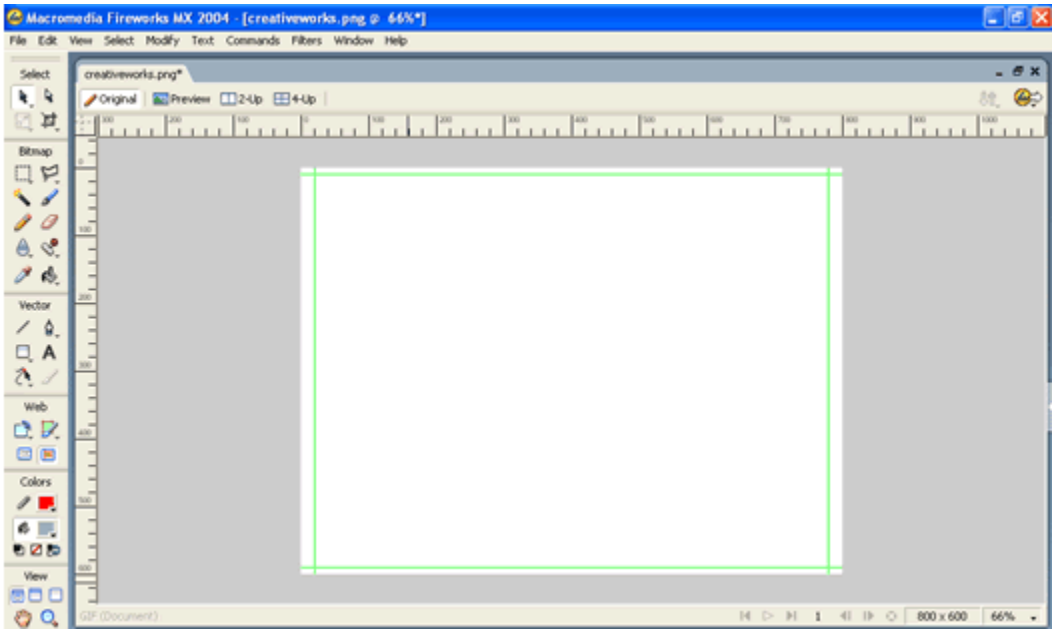
No need to drag it precisely, because with the pointer tool selected, we double-click on the green guide.





And set its position to 20.

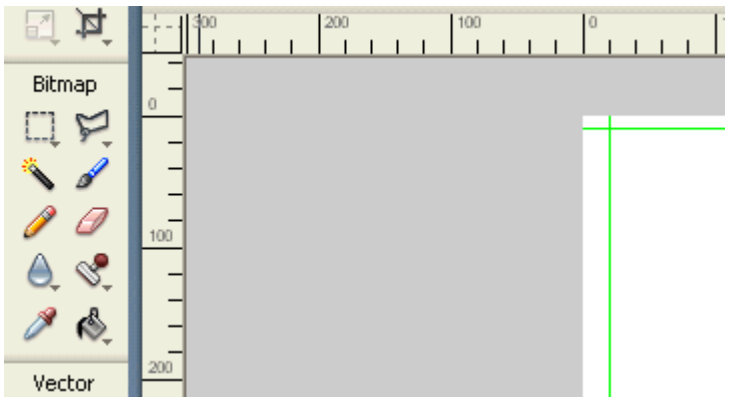
In a similar manner. Set the other vertical guide to 780. And set two horizontal guides -- one at 10 and one at 590.

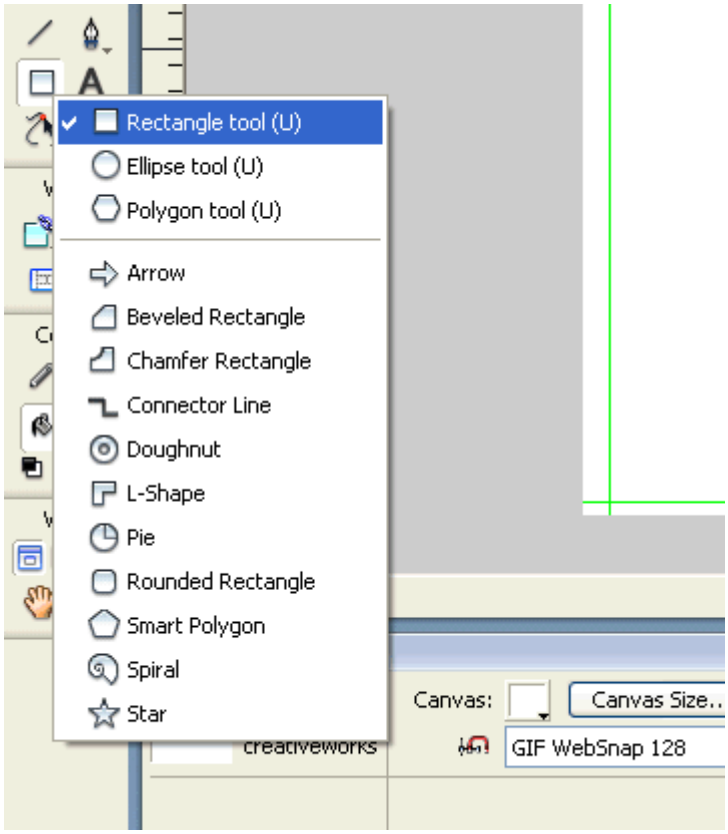


I have set the scaling to 66% so you can see the whole canvas.

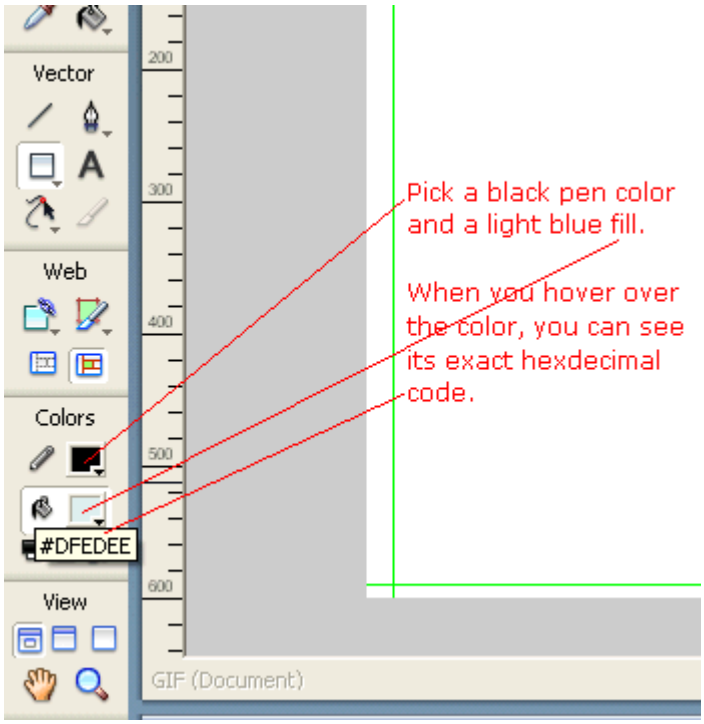
Picking Out Colors

Select the rectangle tool

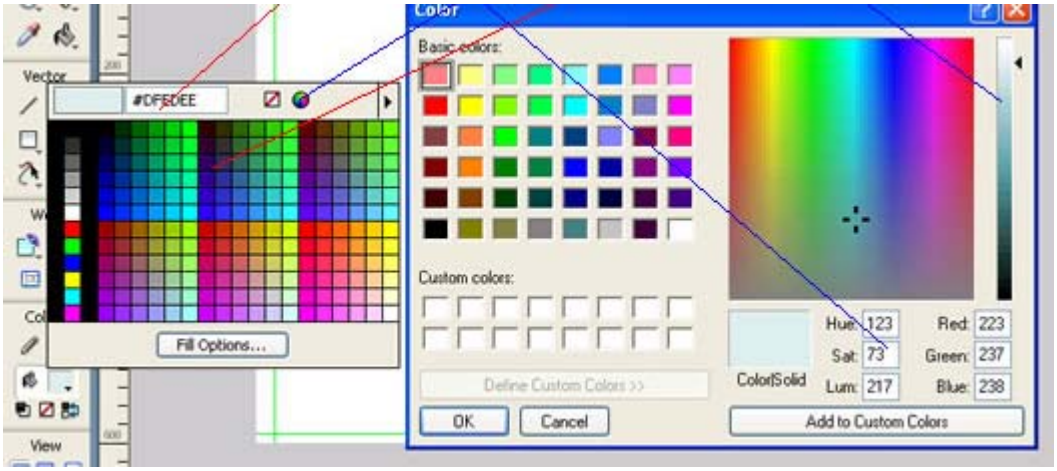




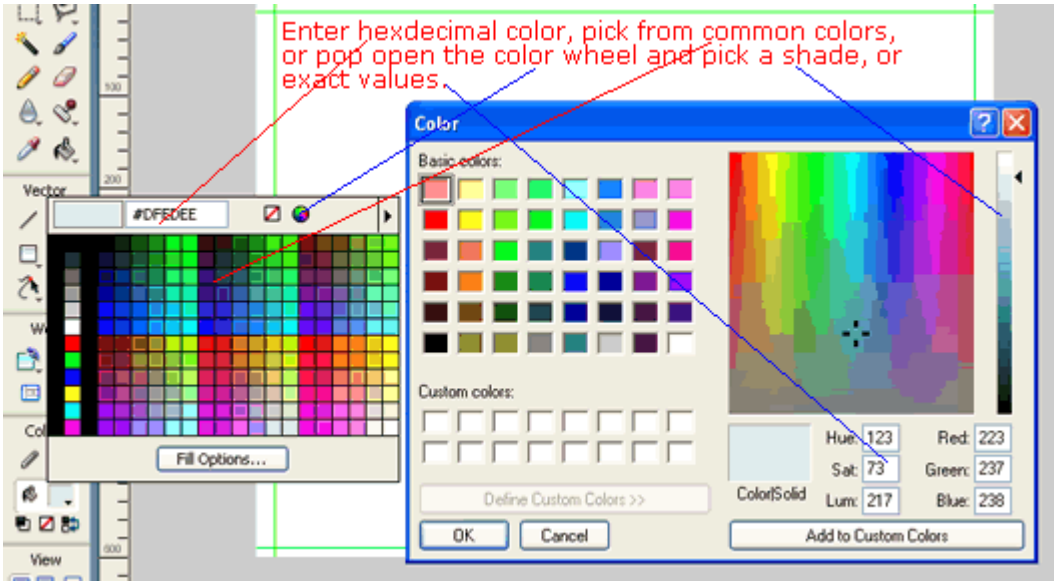
Pick a black pen and a light blue fill.



By clicking on the color, you can have your choice of colors:



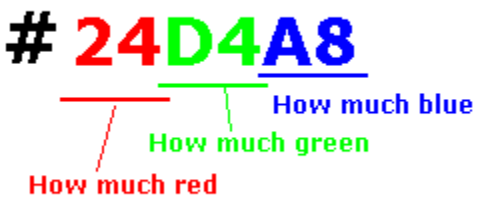
As I was putting this screen shot together, I came across a great example of the difference between GIF and JPG. The above picture was saved in JPG format (Looks good right?). The below picture was saved in GIF format (not so good).



Most of the time, I get my screen shot in GIF format because it is sufficient and gives me small file sizes. But the reason why this image does not look good in GIF is because GIF can have at most 256 colors, whereas JPG can have about a million.

Color Codes

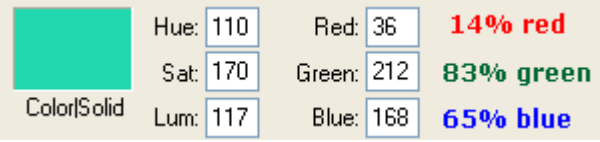
Colors are encoded in 6-digit hexidecimal code such as DFEDEE where each digit can be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. With 6 digits of such values, there are more than a million combinations (enough to encode a million different shades of colors). Hexidecimal code are usually denoted with an # symbol in front. For example,



Colors are a mixture of three primary colors of light: red light, green light, and blue light. The first two digits tells you how much red light is in the color. **00** means no red light at all. **99** means a significant amount of red light. **AA** means even more red light. **CC** means even more. And **FF** means the maximum amount of red light. The next two digits tells you how much green light. And the rightmost two digits tells you how much blue light. You can think of the digits A, B, C, D, E, F as the numbers following 7, 8, and 9. After 9, comes A, then comes B, etc.

Hence #000000 is no red, no green, and no blue light. In other words, black. And #FFFFFF is the color white. This implies that white light is a mixture of all lights. This makes intuitive sense as we recall that if you pass white light through a prism, it spits apart to all the colors of the rainbow.

The above hexidecimal value of #24D4A8 represents the following color.



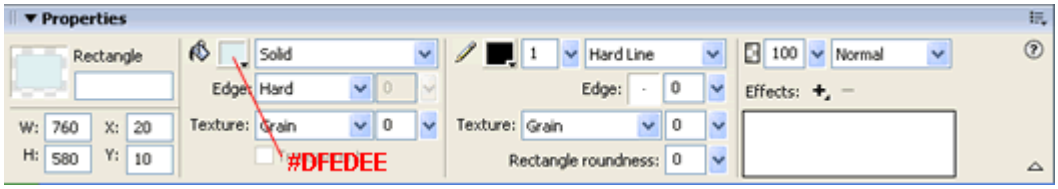
Another way to represent color is the RGB value. The color #24D4A8 can be equivalently represented in RGB form (or red-green-blue form) as **rgb(36,212,168)** where those number come directly from the Fireworks color dialog seen just above.

The way to equate the hex value of #24D4A8 to the rgb value of rgb(36,21,168) is to look at it this way. At what division is the hex value #24 at in the range from #00 to #FF? Just count from #00 to #FF. There are 255 counts (or divisions) as you count from 00 to FF. And about 14% of the way (or at count 36) through your counting, you will hit value #24.

Don't worry too much about the color representation. We just pick whatever color looks good.

Properties of the Rectangle Tool

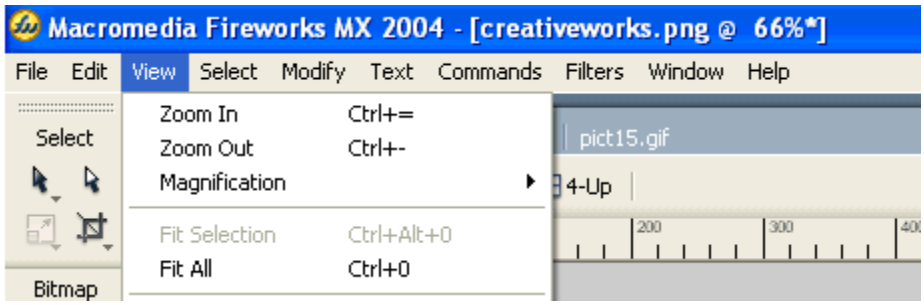
So we have seen how to pick colors from our Toolbox. Another way to pick the pen and fill color for the rectangle tool is to select the rectangle tool and then click the colors in the Properties panel.

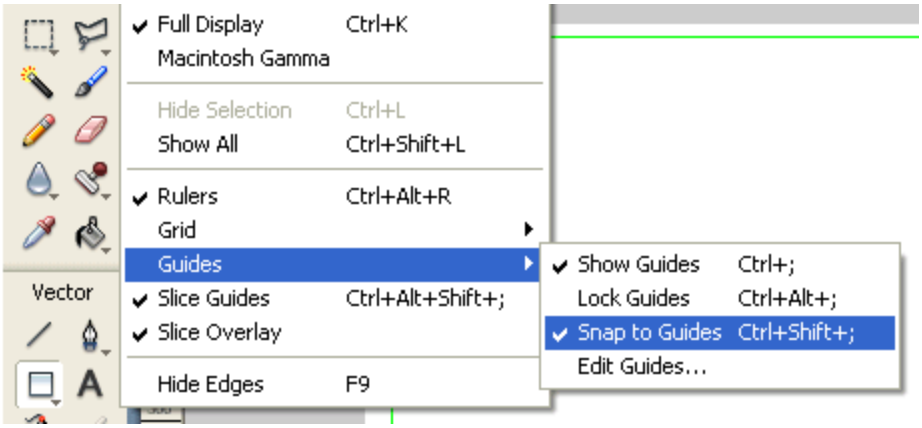


With the rectangle tool selected, there are additional properties in the **Properties** window that you can set. Set them as shown in the above. For the Fill, set its color to **#DFEDEE**. Set the **Edge** to **Hard** and **Texture** to **0**. For the Pen, set the thickness to **1** and **Hard Line**. Set **Edge**, **Texture**, and **roundness** set to zero. The opacity is set to **100 Normal**.

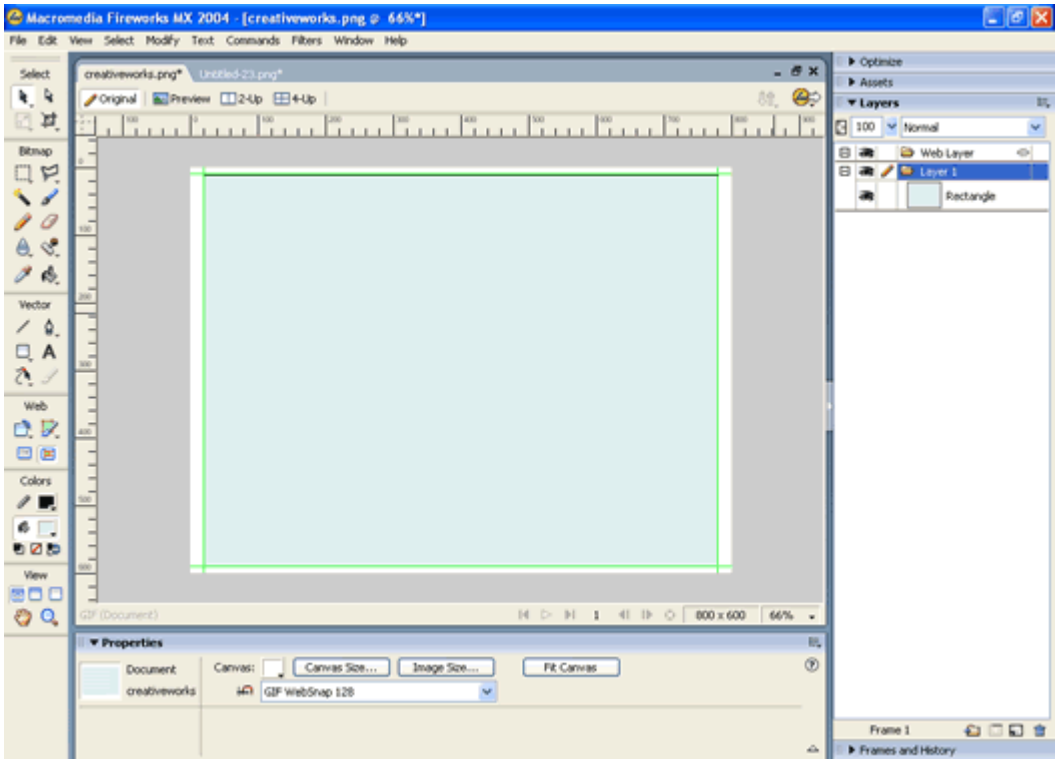
Drawing Our Layout

Next, make sure that **Snap to Guides** is checkmarked...





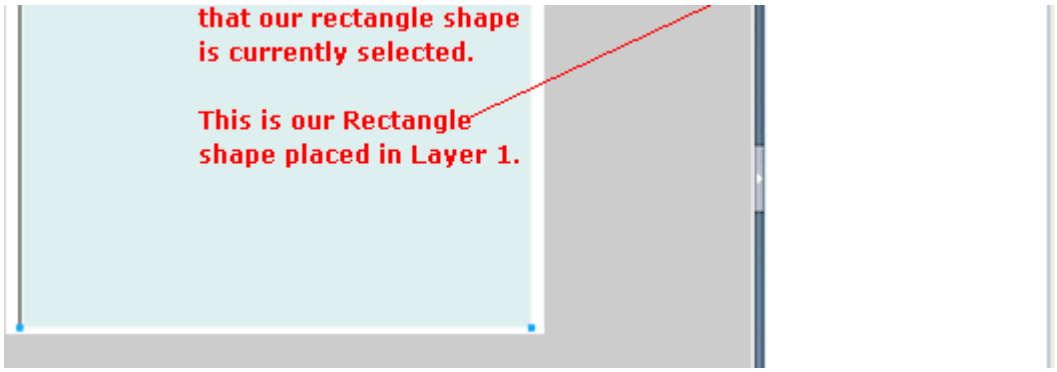
And draw out our rectangle as shown.



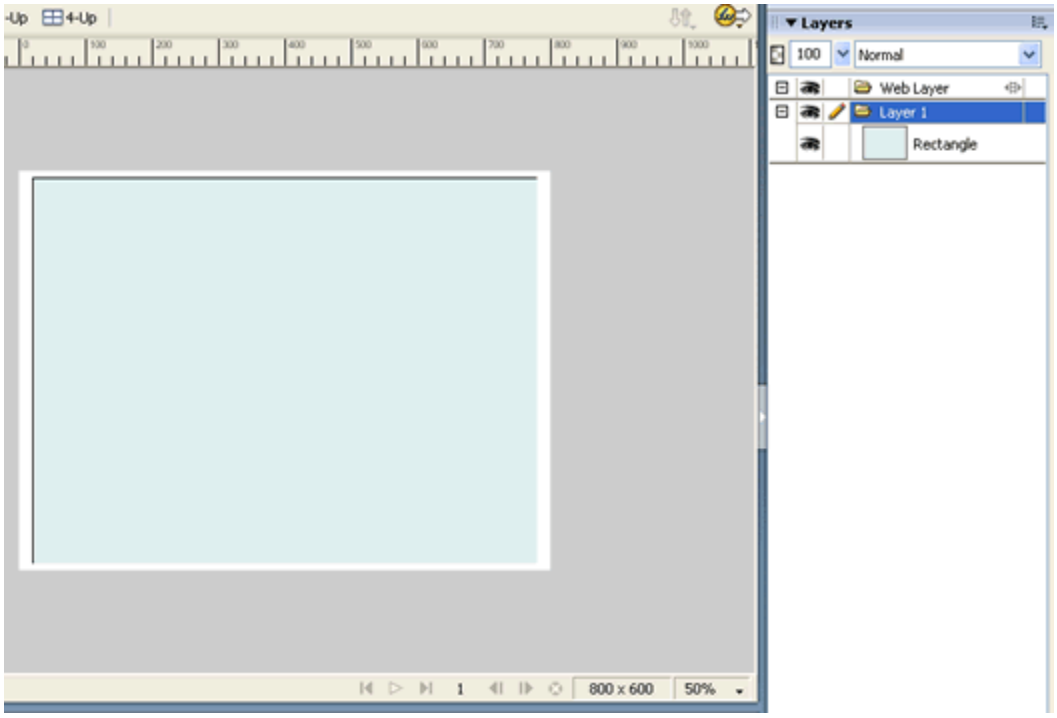
We are going to have a bluish webpage with a black border. But we can not see the black border quite clearly since it was set to be only 1px thick and is being covered up by the guides.

Do **Ctrl-;** (that is pressing the semi-colon key while the Ctrl key is depressed) and the guides will disappear. This is equivalent to the menu command **View -> Guides -> Show Guides**.

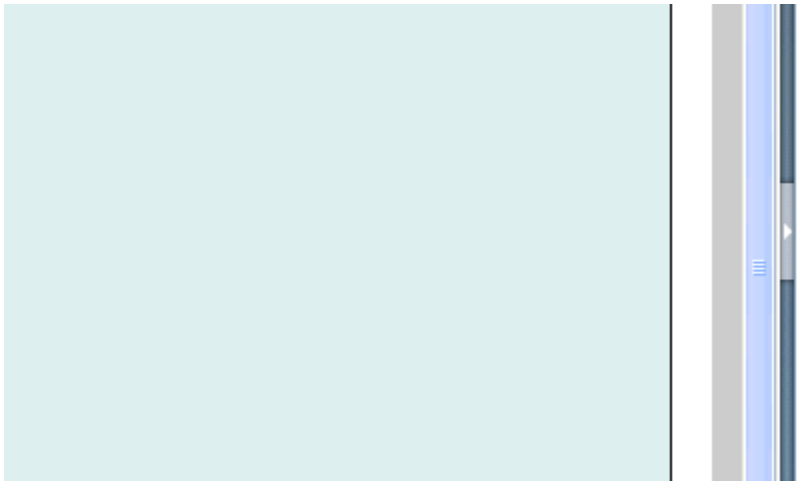
Still can't see our border clearly. Because now, Fireworks is putting handles on our shape since it is the currently selected shape.

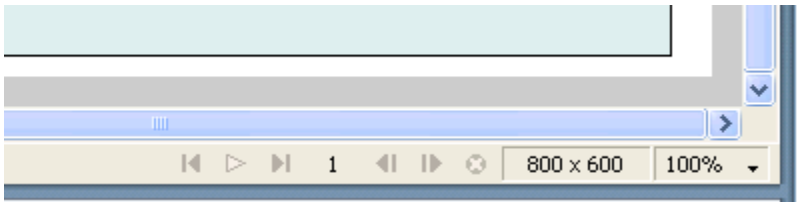


De-select our shape by clicking on an empty spot in the canvas.

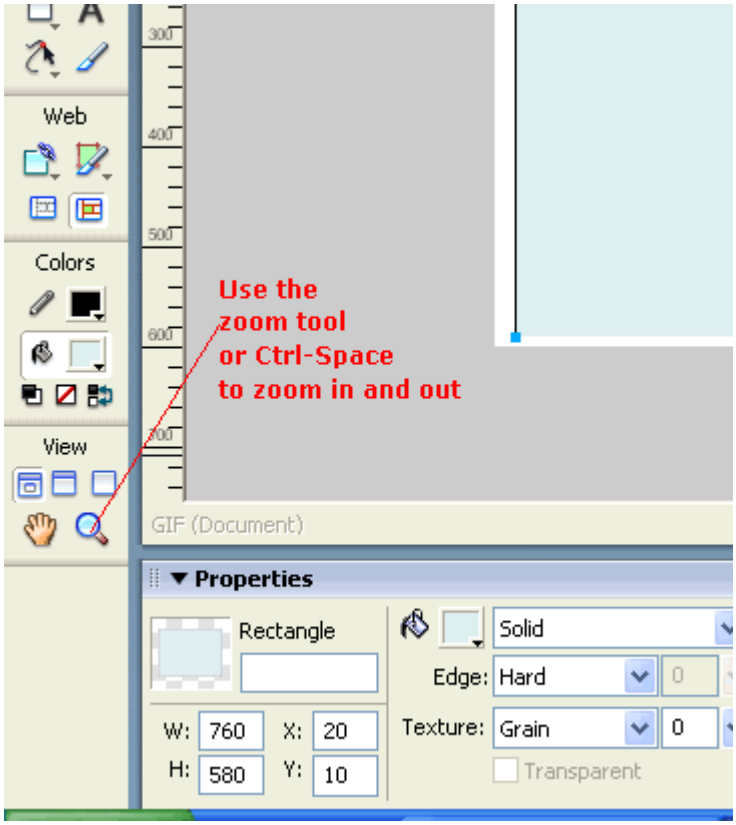


Our handles have disappeared and our Rectangle shape in our **Layers** panel is no longer selected. But it looks like we are missing the right and bottom border? Actually no, we do have all four border, but since you are viewing at 50% scale, the monitor screen does not have enough pixels to show you all the details. Set you view back to 100% scale, scroll around, and you will see your borders. Okay, now you see why I made such a big deal about 100% scale (or **Ctrl-1**).

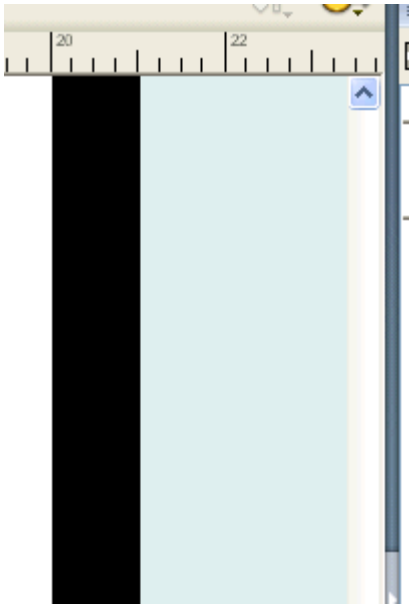


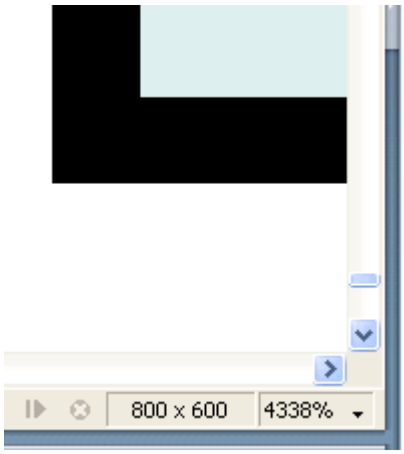


Zoom Tool



Another way to look in closer is to use the zoom tool. I like to use **Ctrl-Space** and drag to zoom in. If the **Alt** key is depressed while using the zoom tool or using **Ctrl-Space**, it performs a zoom-out instead of zoom-in.

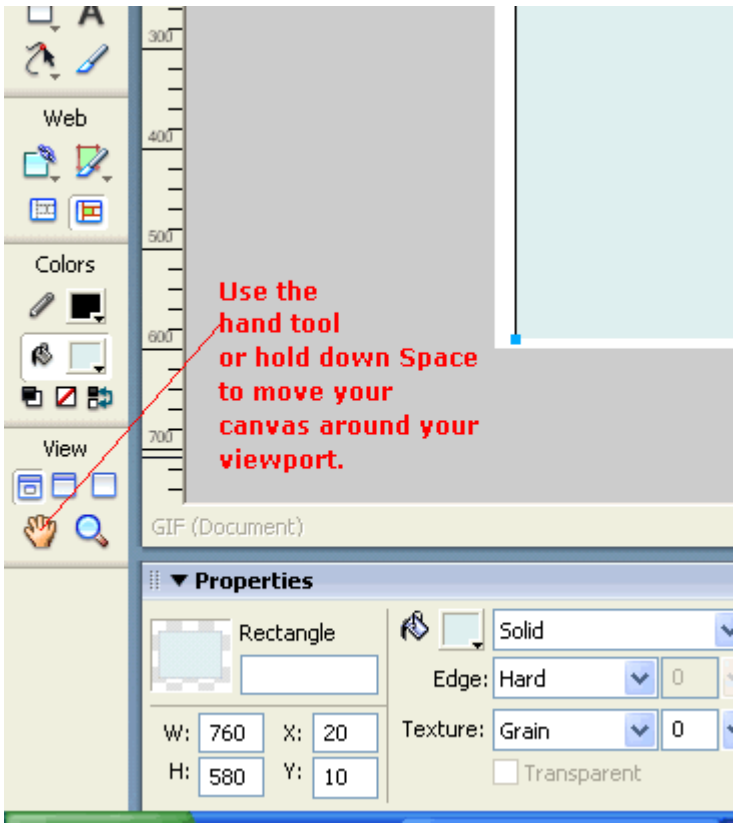




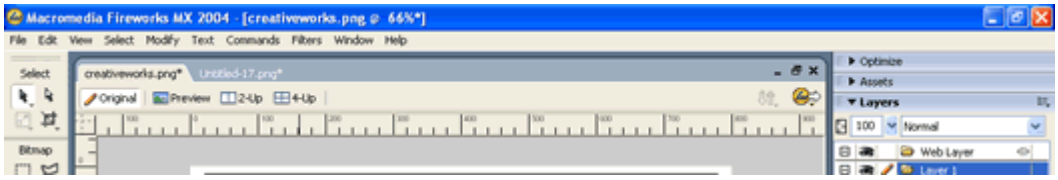
Here I have zoomed in to about 43 times its normal size (4338%). See that the ruler measurement always mark off the correct pixel dimension. Although on the screen, the line looks as if it is 43 pixels wide; according to the ruler, it is really only one pixel wide.

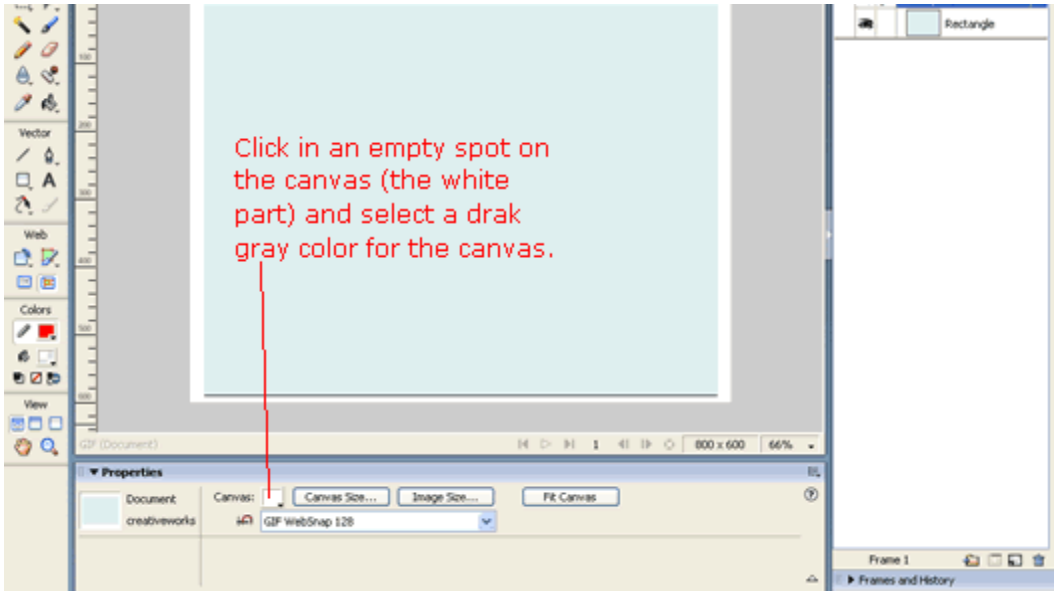
Hand Tool

Another useful tool is the hand tool invoked via the space bar or from the toolbox as shown. Try it. It enables you to move your canvas around.

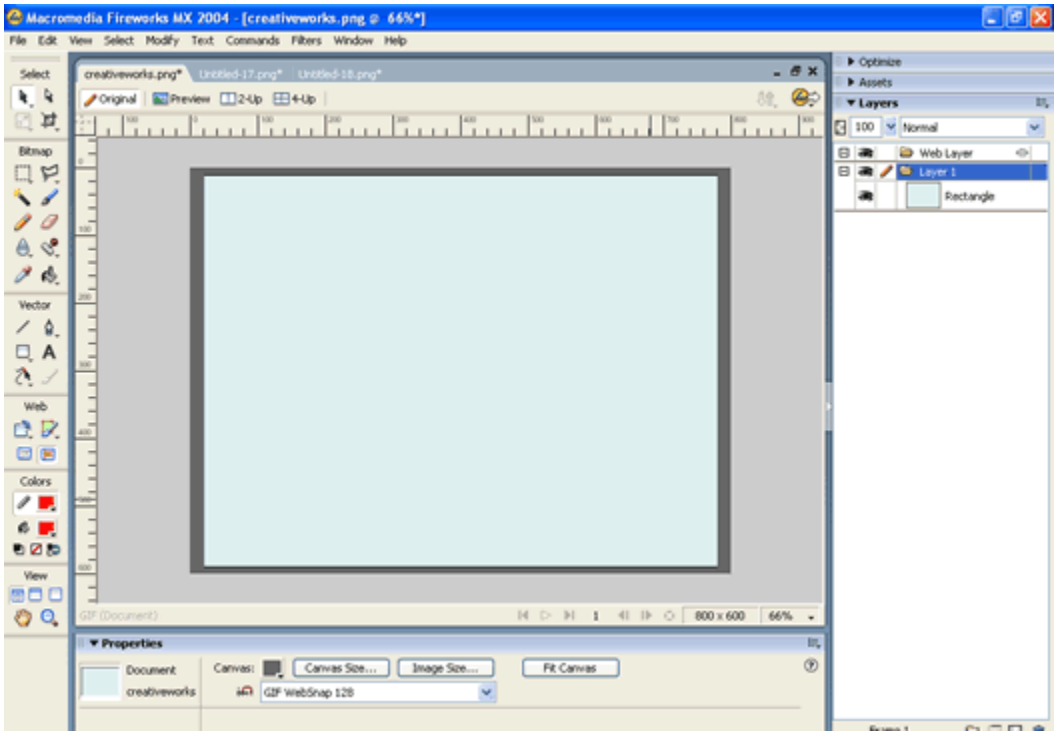


Changing Canvas Color





We want a dark gray background to constrast nicely with our light page. So change the canvas background color by the above instructions and we get



Saving the File

The asterisk next to the filename in the file tab indicates that our file has been modified and requires saving. Good time to save the file and also a good time for a break.

Here is the file we have so far:

creativeworks.png

Click on the link to bring it up in the browser. Then right-click on the image and do a "**Save Picture As**".

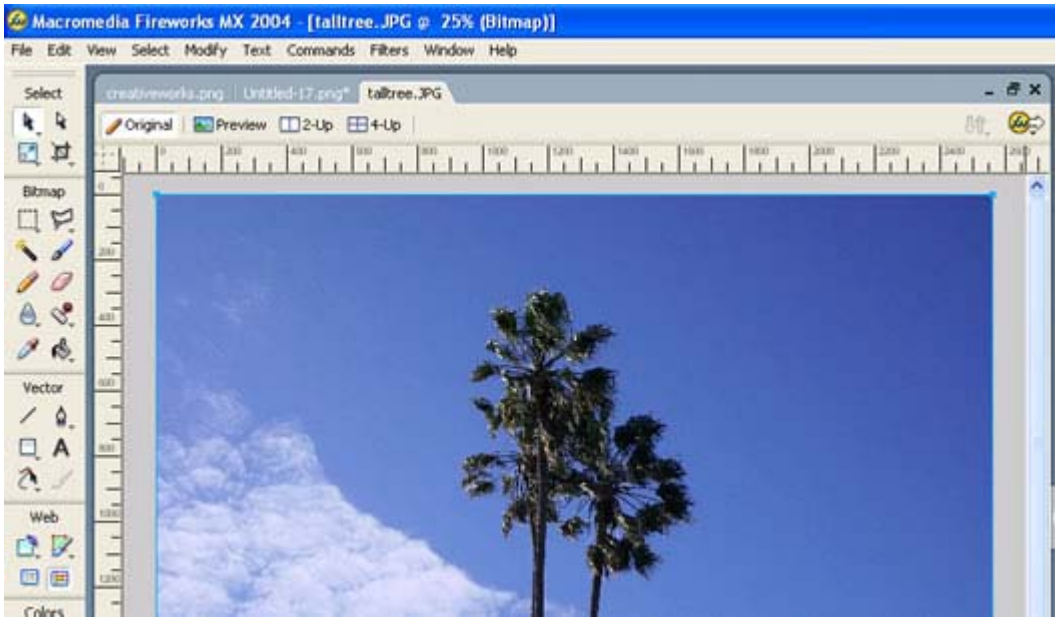
Enhancing Photos in Fireworks

We want to import an nice graphic of size 160x473px on the left side of our page. I picked an image that is tall and slender in proportion to those dimensions. Also the blue sky matches the blue color scheme of our page.

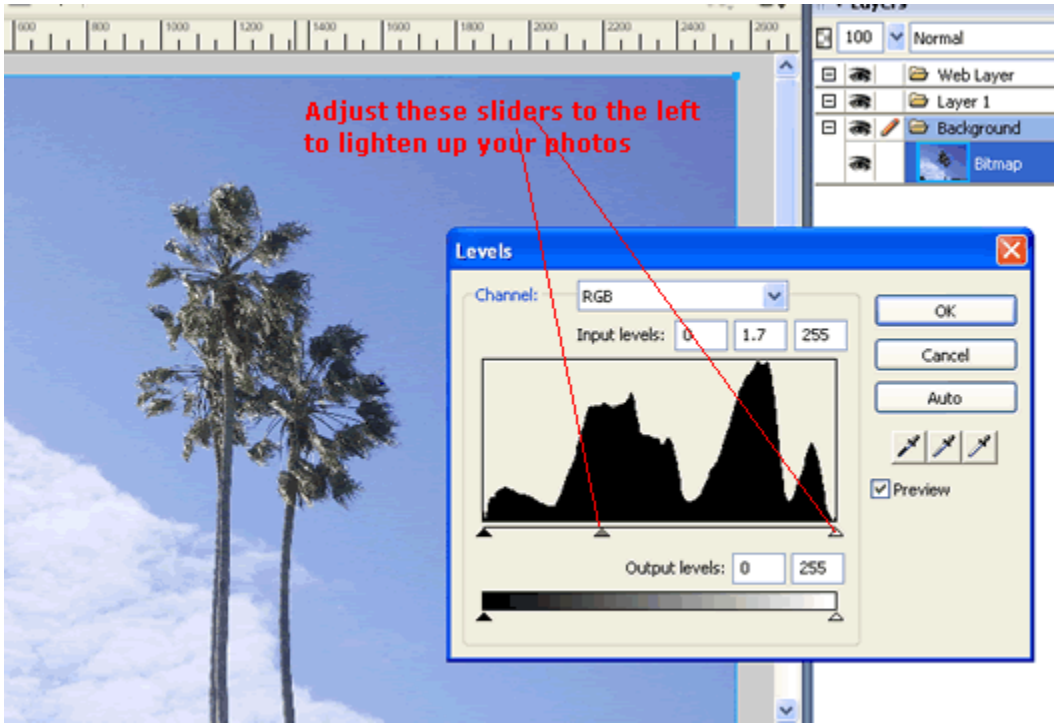
If you like to use this image, you can download from **here**. Click on the link to open it in a browser. Then right-click on the image and do a "**Save Picture As**". Or you can use your own photograph. Often as designers, we need access to stock photography. There are quite a few stock photography sites out there, some of which are listed **here**. **iStockPhoto.com** and **dreamstime.com** are popular sites that provide stock images at low cost. **Stock Exchange** has some photos at no cost. But please observe image copyrights and do not just perform a Google image search and grab images off the web, as these images are not consider public domain. Save the downloaded image in your project folder **creativeworkscomp** so that you have it on hand.

Often the image will be too large. We eventaully want an image that is 160x473 in size. To do this, you will need to use the crop tool as well as change the image size. Always make the alterations on a copy of the images so that the original remains unaltered.

In Fireworks, open the image file and change the view scale as necessary to view the whole file. I find that more often than not, my images from the digital camera comes out too dark due to poor lighting, bad settings, etc.

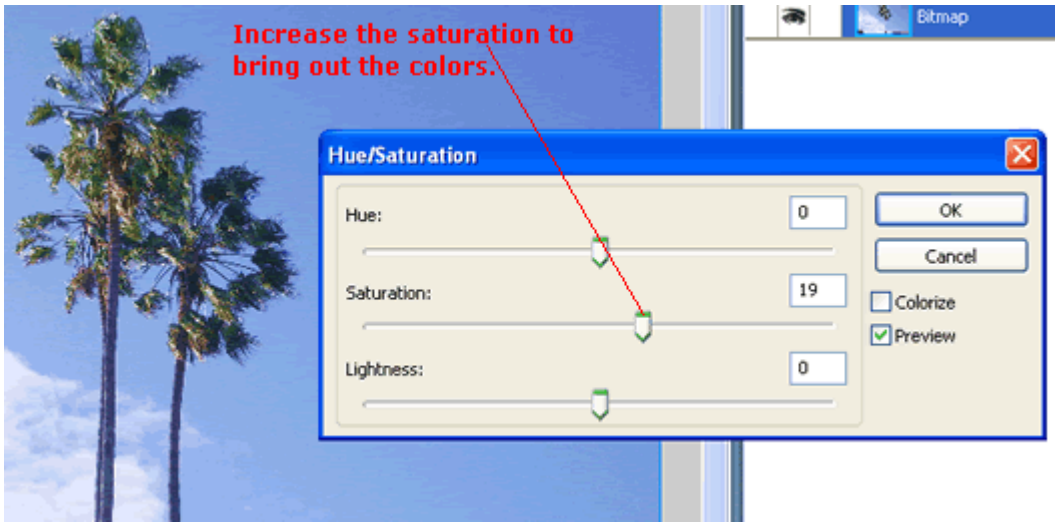


If that is the cae, you can lighten it by **Filters -> Adjust Colors -> Levels** and play with the sliders...



Or you can have Fireworks do it for you by **Filters -> Adjust Colors -> Auto Levels**.

Next I increased the saturation to bring out the colors a bit. But don't overdo it.



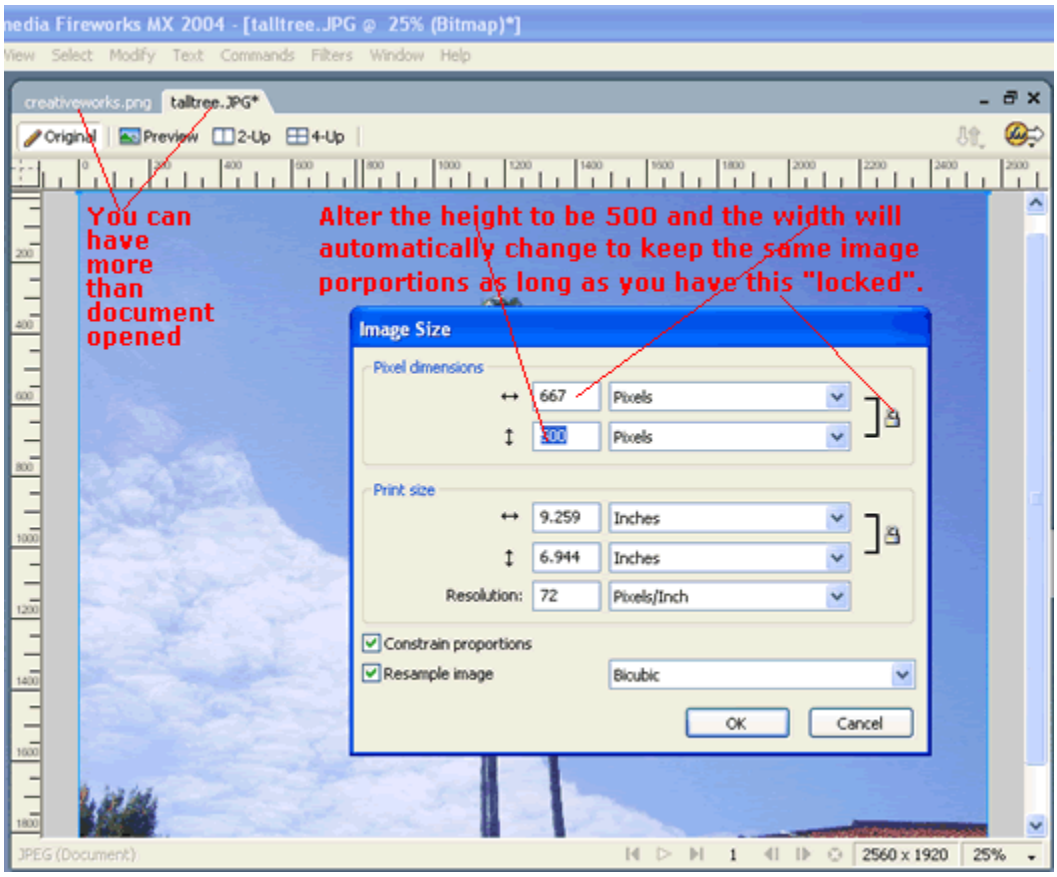
Feel free to make other enhancements such as **Brightness/Constrast** and **Filter -> Sharpen -> Sharpen** if needed.

Fireworks is decent in performing the common basic enhancements. But Photoshop can often do more and be better at photo touchups.

Resizing Images

This image that came out of my camera has a dimension of 2560x1920px and was over 3 MB in size in JPEP format. Obviously, we do not want such large images on our webpage. I resized my image to a height of 500px by **Modify -> Canvas -> Image Size**. By clicking on the lock icon in the **Image Size** dialog, I can lock or unlock aspect ratio. Keep it

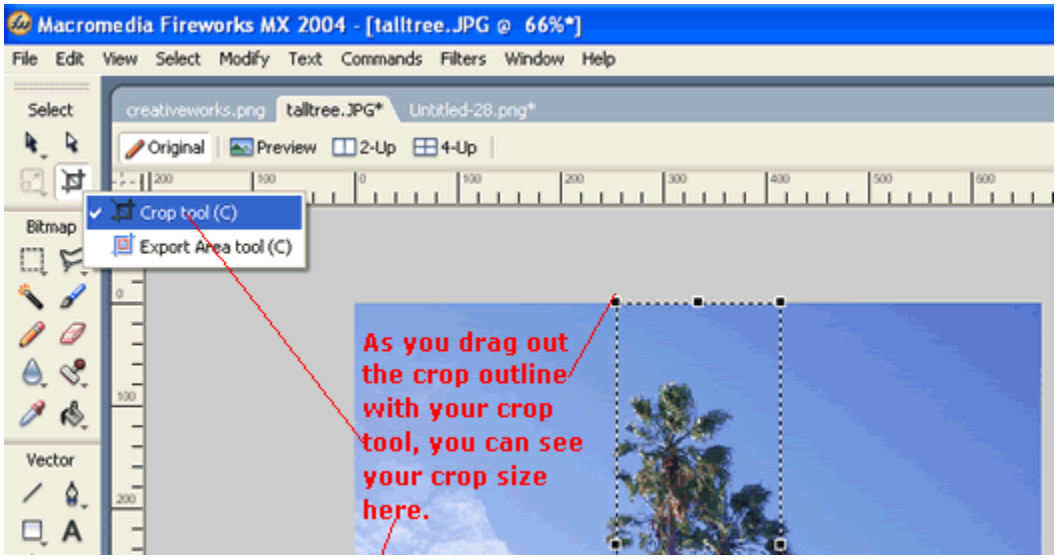
locked and change height to 500px. The width will automatically change to keep the image in the same proportions.

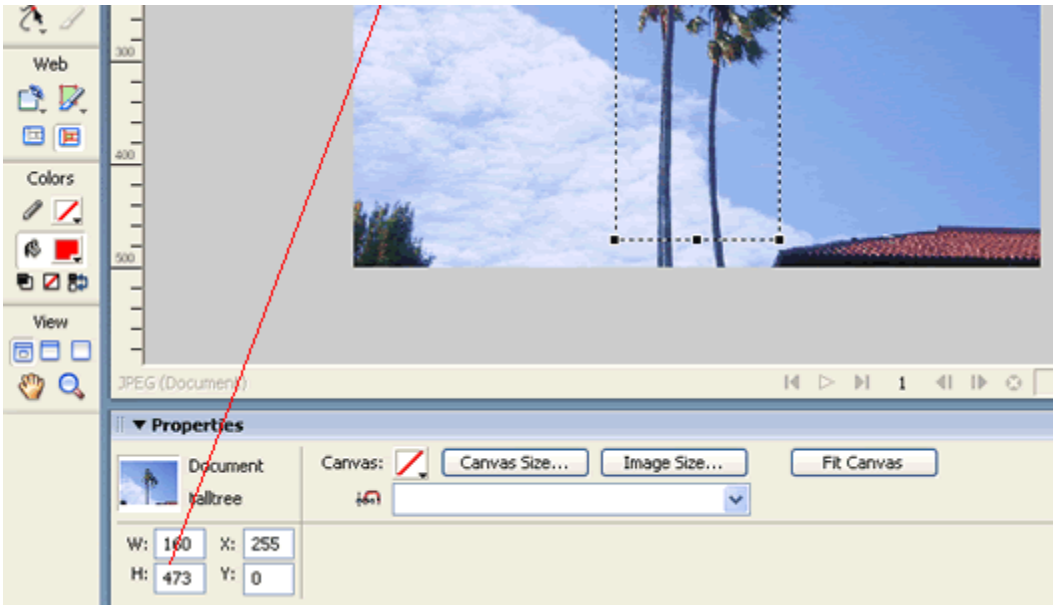


The reason why I resized the image to 500px even though I knew I wanted an image that is 473px high is because I may want to crop out some of the image at the bottom (such as that distracting roof). I don't want to have an image that is too small and then have to resize it larger. You loose quality whenever you resize an image larger since you are asking the computer to fill in pixels where there had been none before.

Cropping

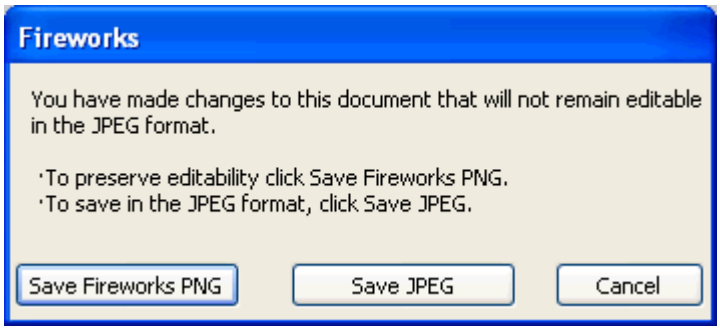
Now that our image is smaller, it is time to use the crop tool to cut the image out to the exact dimensions of 160x473px. Select the crop tool and drag.





Don't worry about letting go of the mouse. Even after you let go of the mouse, you can further adjust the crop by its resizing handles as well as moving the entire crop outline as a whole. In fact, to get get the fine control needed for the precise dimensions, you most likely will need to release the mouse, zoom in, and adjust the crop outlines. Once you have the outlines as you like them, press **Enter** and the image is now 160x473.

When you save this image, Fireworks may ask



I'm just going to save it as JPEG format with the filename **talltree.jpg** in the folder **creativeworkscomp**.

Here is my final image. If you rather use my image for the rest of the lesson, click on the below image to bring it up in the browser and right-click on it and do a "**Save Picture As**".

Importing Image in Fireworks

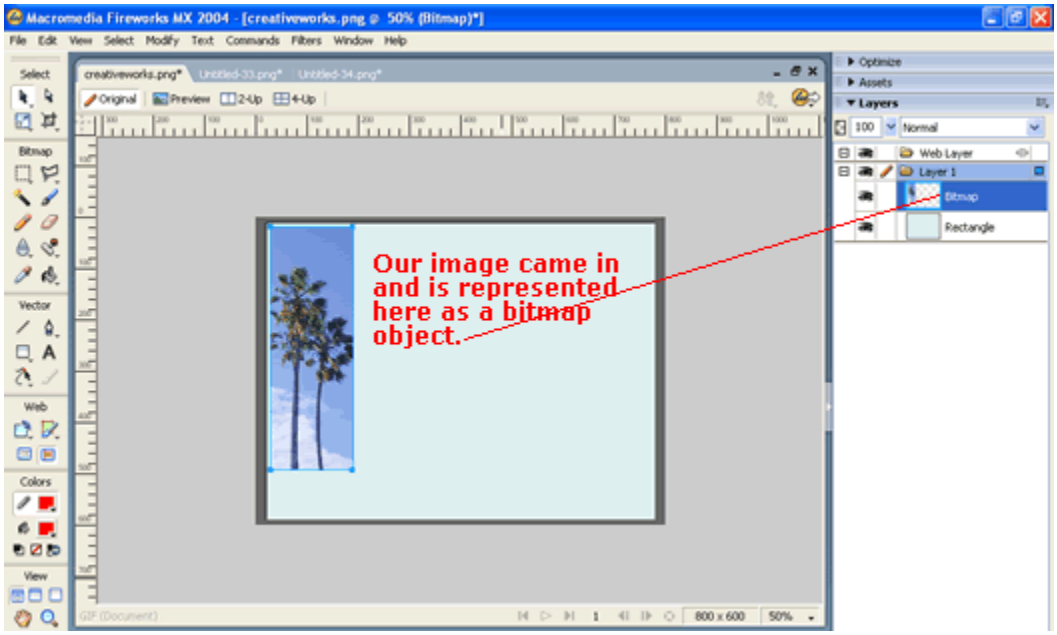
We want to import a photo into a design comp.

In a previous lesson , we had started a design comp called **creativeworks.png** (which you can get from the link). It is a rectangle with a black border.

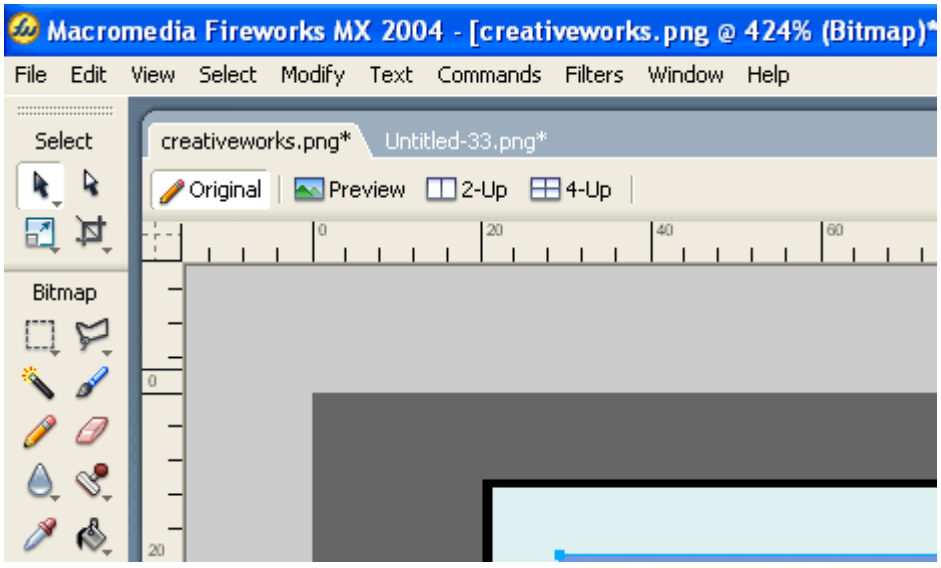
In the last lesson, we had prepared the photo **talltree.jpg** in the folder **creativeworkscomp**.

We want to import the **talltree.jpg** file into our **creativeworks.png** layout file.

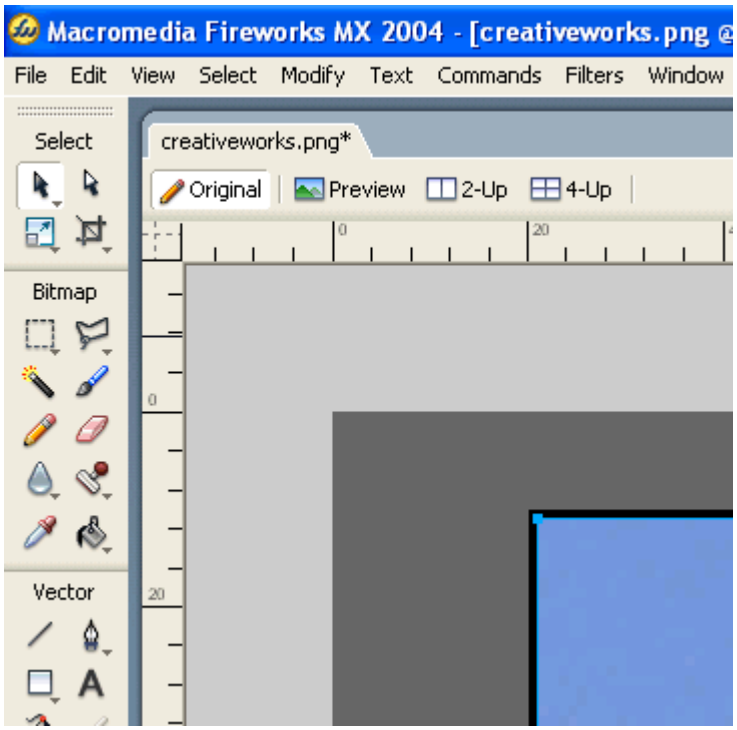
With **creativeworks.png** file opened, do a **File -> Import**, select **talltree.jpg**. Your cursor becomes an L-shaped cursor indicating that you are about to drop an imported image to the canvas. Click near the upper-left corner of our canvas to drop our image there. You don't need to be precise since we will position it exactly later. Here is what it looks like after we drop the image...



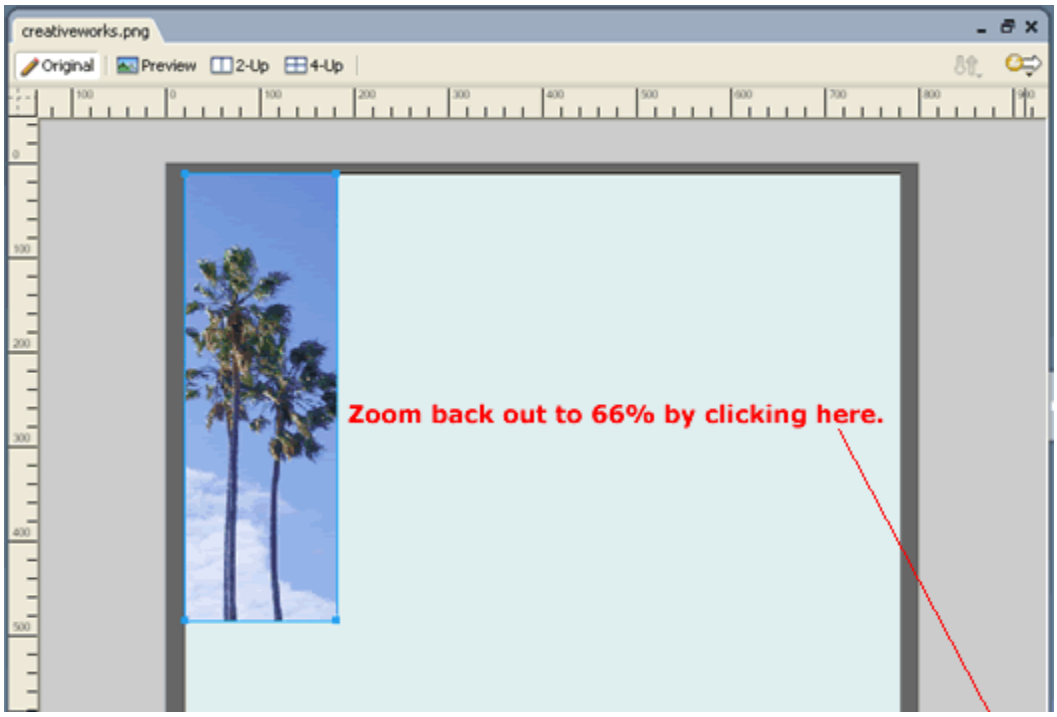
After dropping our image in the upper left corner, we can more precisely position it by zooming in ...



With the image selected, move it in place with the arrow keys so that it touches right at the borders as shown below...



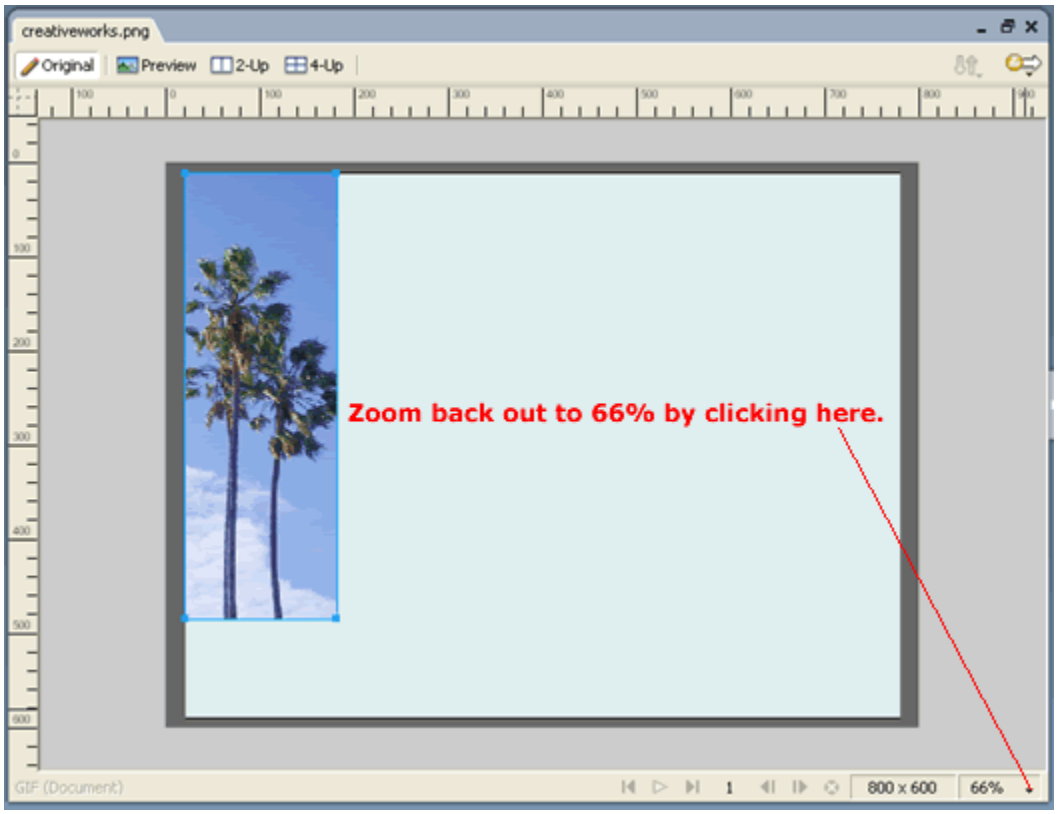
After zooming back out to 66% in order to see the entire canvas, this is what we have...



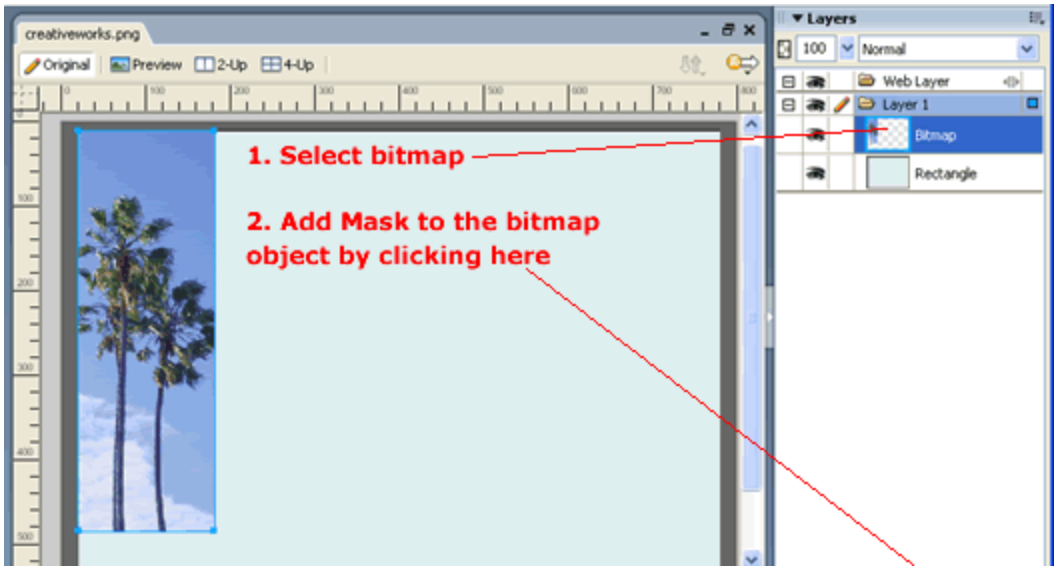
And you can download the resulting **creativeworks.png** here.

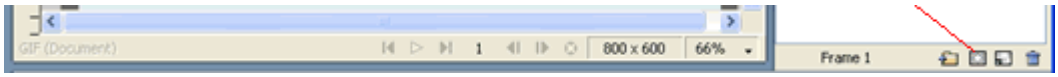
Gradient Transition

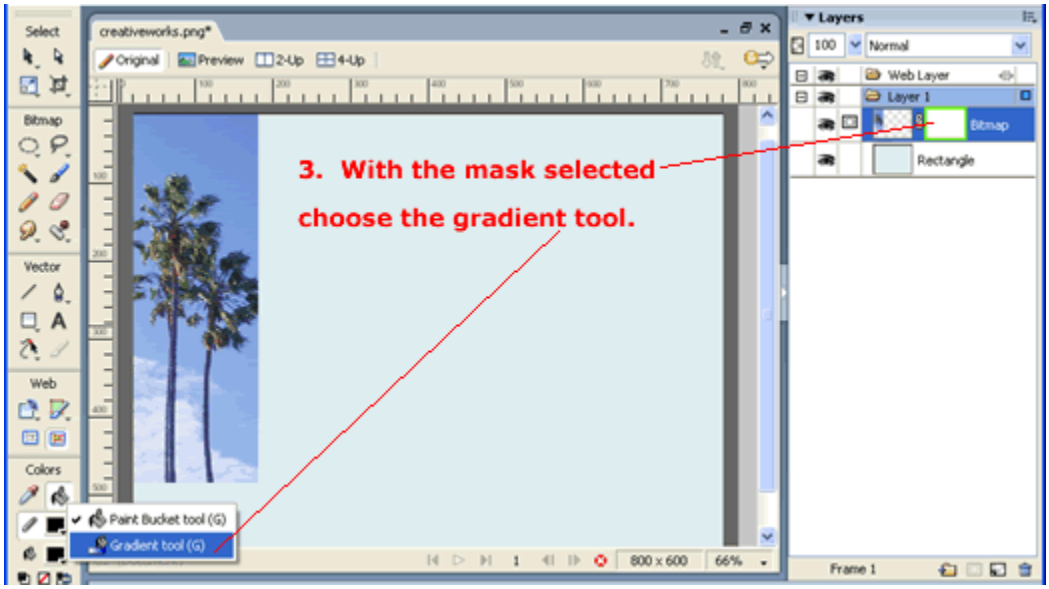
After importing a photo into our design comp, you might find it necessary to zoom back out to about 66% so that we can see the entire canvas.



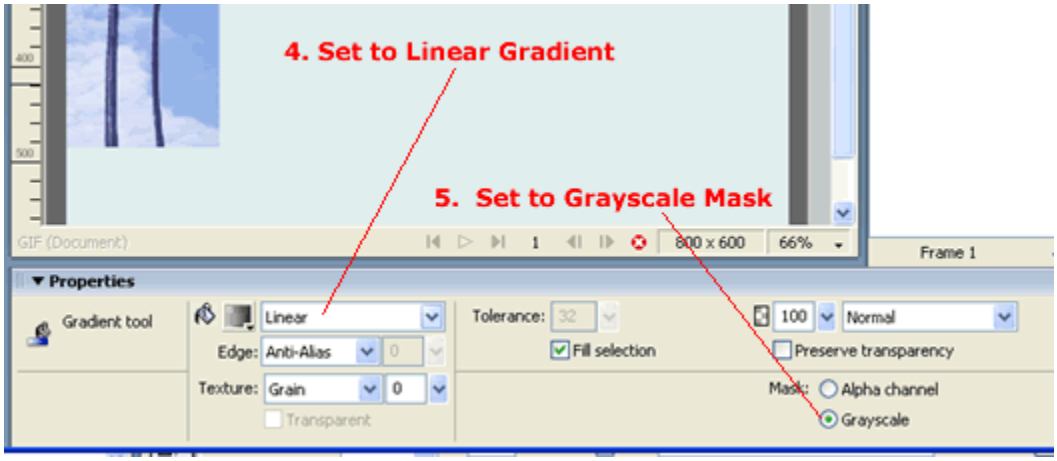
Instead of the abrupt edge where the bottom of the image meets the page, we can have a transitional gradient by following these steps...

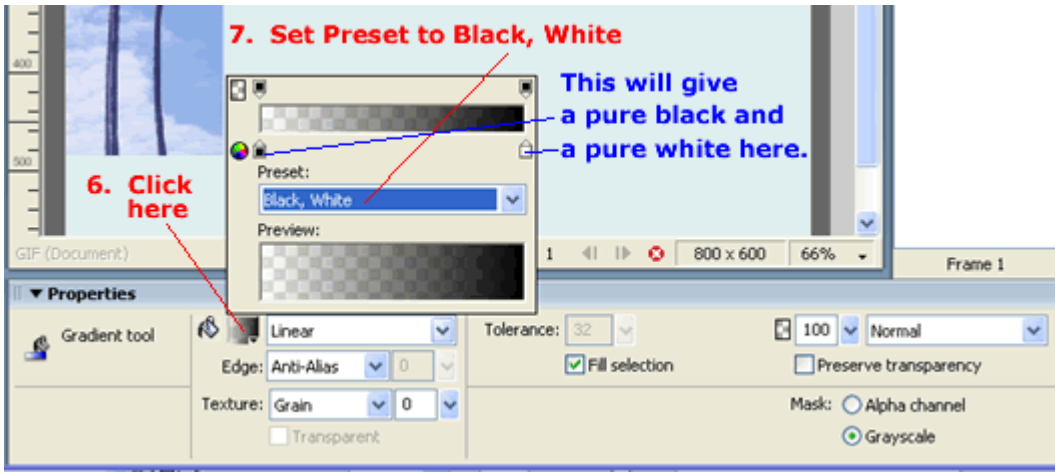


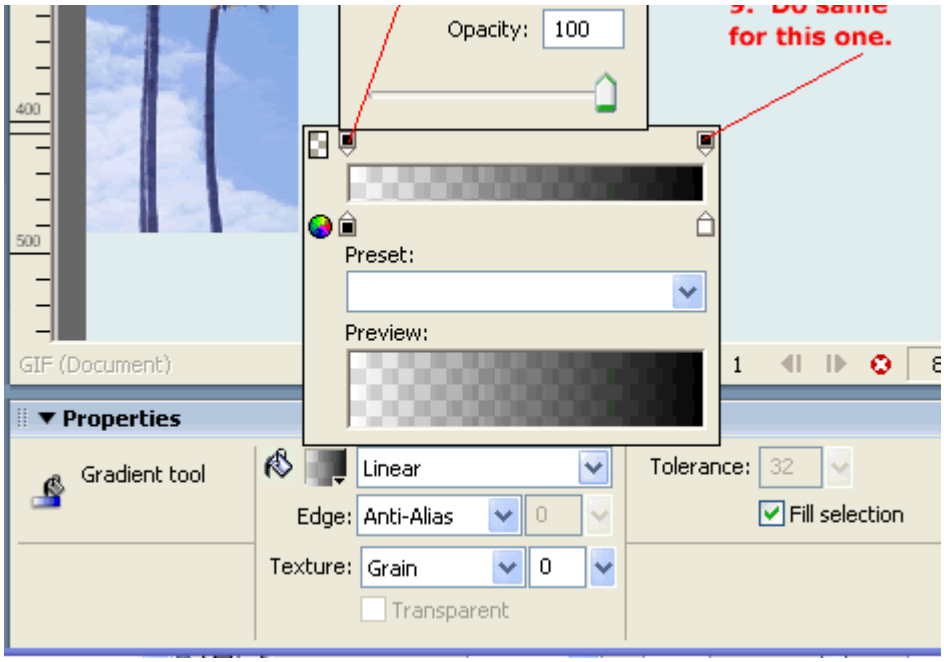




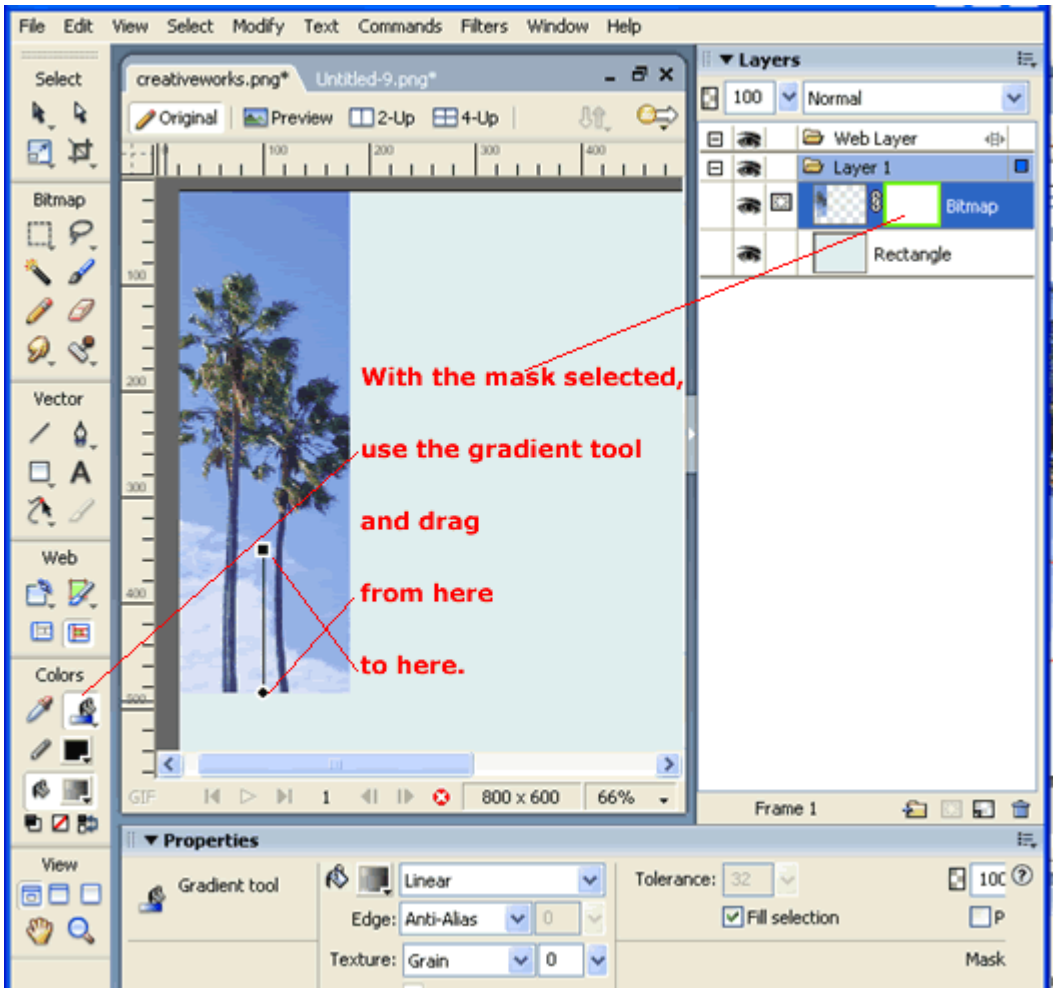
Set the gradient tool to have a linear gradient and a Grayscale mask and the rest of the setting as shown.



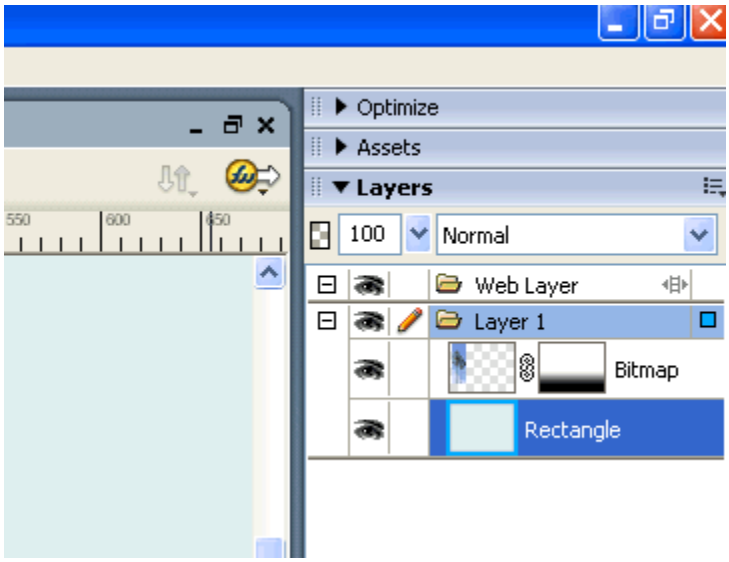




With the mask still selected, apply the gradient tool with a drag from bottom to top as shown. If you hold down the shift key while dragging, the line will be constrained to be perfectly vertical.

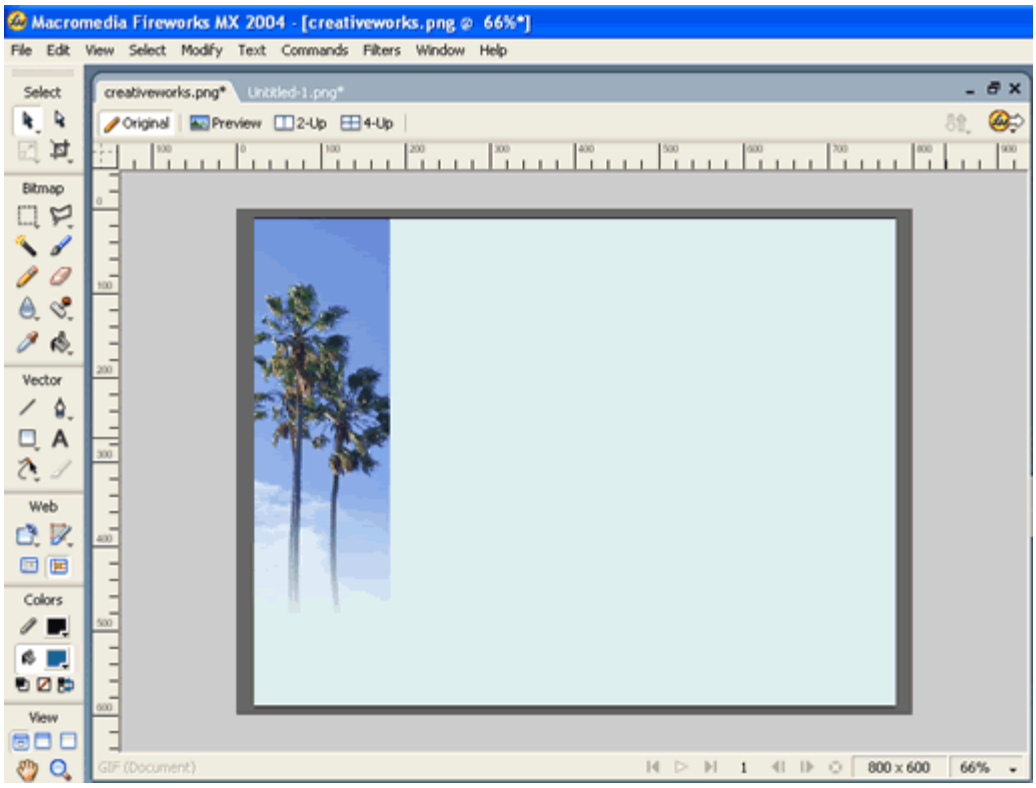


Look at our mask...



We had applied black on our mask at the bottom of our image. Pure black on our image will mean that that portion is completely ***masked out*** (or invisible). Applying pure white means completely visible. Our gradient goes from complete black at the bottom to complete white higher up. Meaning that our image will be totally invisible at the very bottom and slowly transition to completely visible about a quarter way up.

Here is what we have...

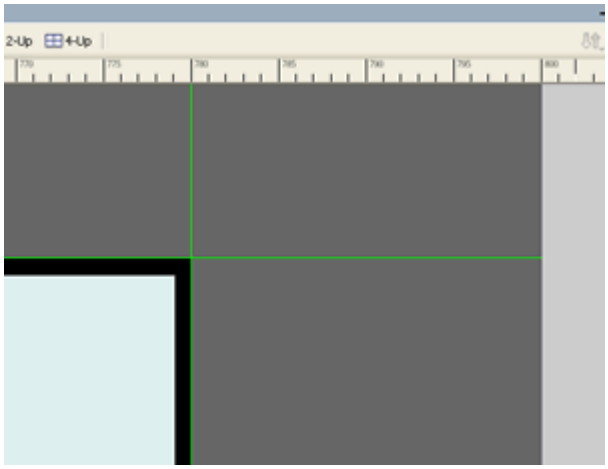


You can download my resulting **creativeworks.png** file from the link.

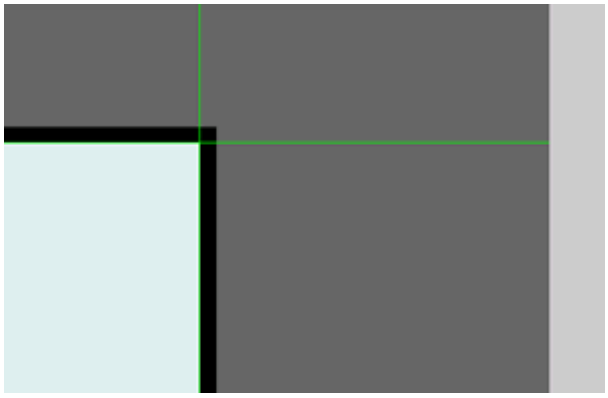
If you want to learn how to fade more than one edge of an image, see this **related tutorial**. Or if you need to fit an image into a rounded rectangle instead of a square corner, take a look at **this tutorial**

Making the Header

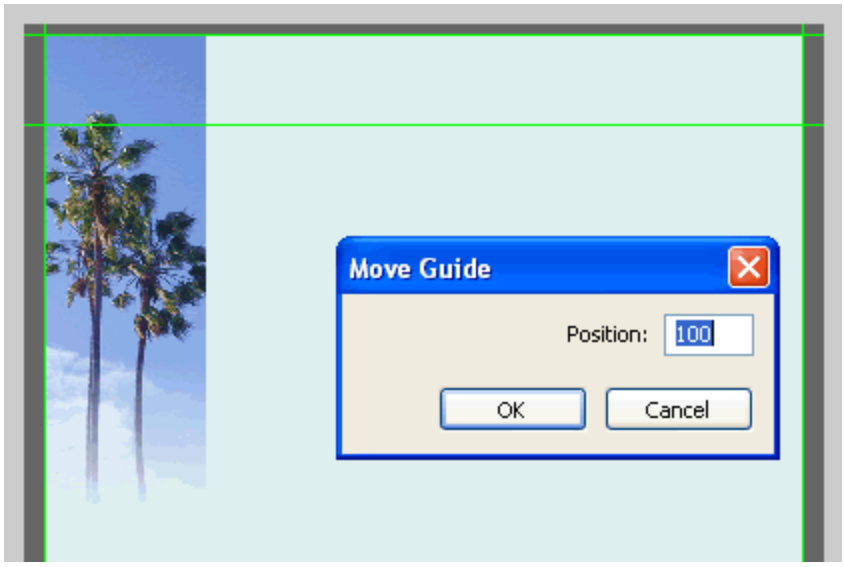
We are now going to put in a header or masthead with the title of our website. Let's make it the same color blue as the sky. We need to draw a rectangle that is precisely lined up with the black borders. Since we had already created some guides previously, turn them on with **Ctrl-;** or menu **View -> Guides -> Show Guides**. If you zoom in on the upper right corner, you might see that our guides are outside the border.



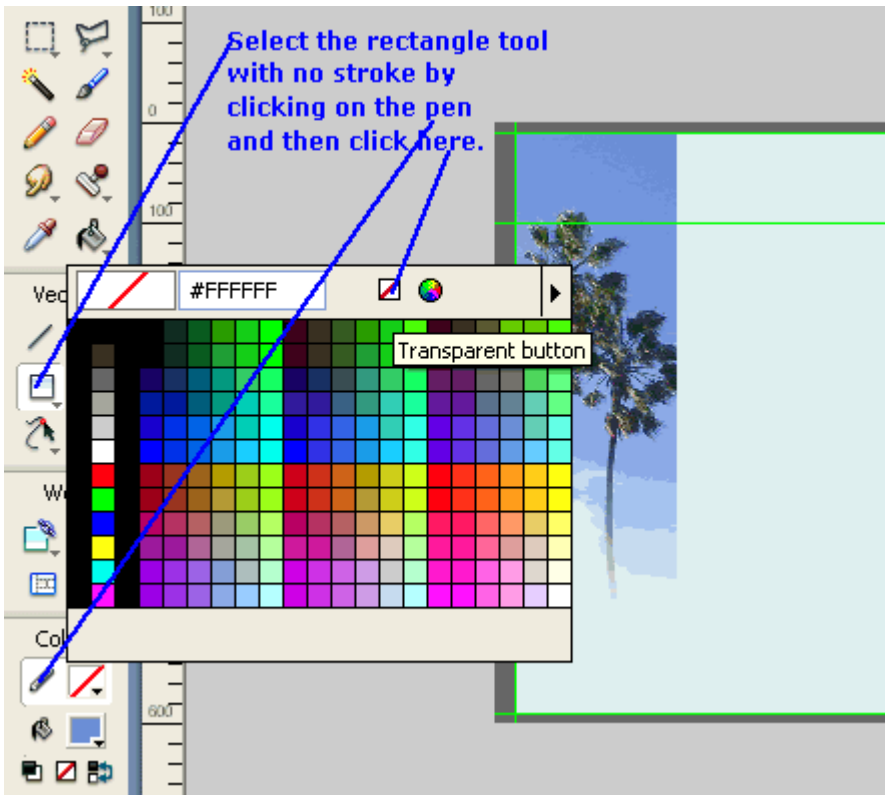
If that is the case, move the guides to be inside the borders by dragging them...



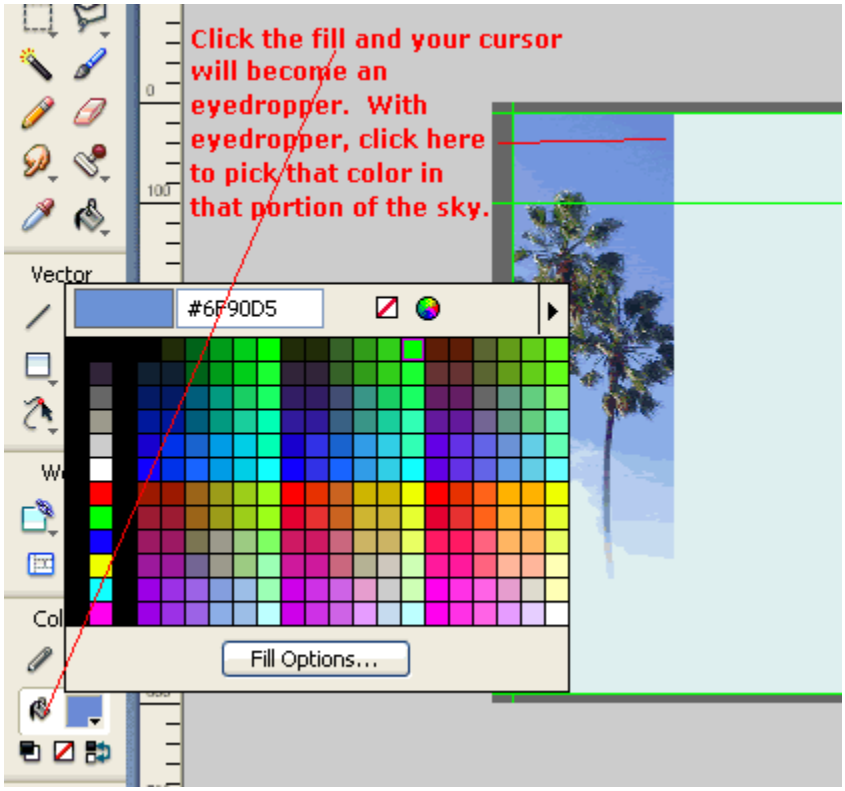
Put another horizontal guide at position 100...



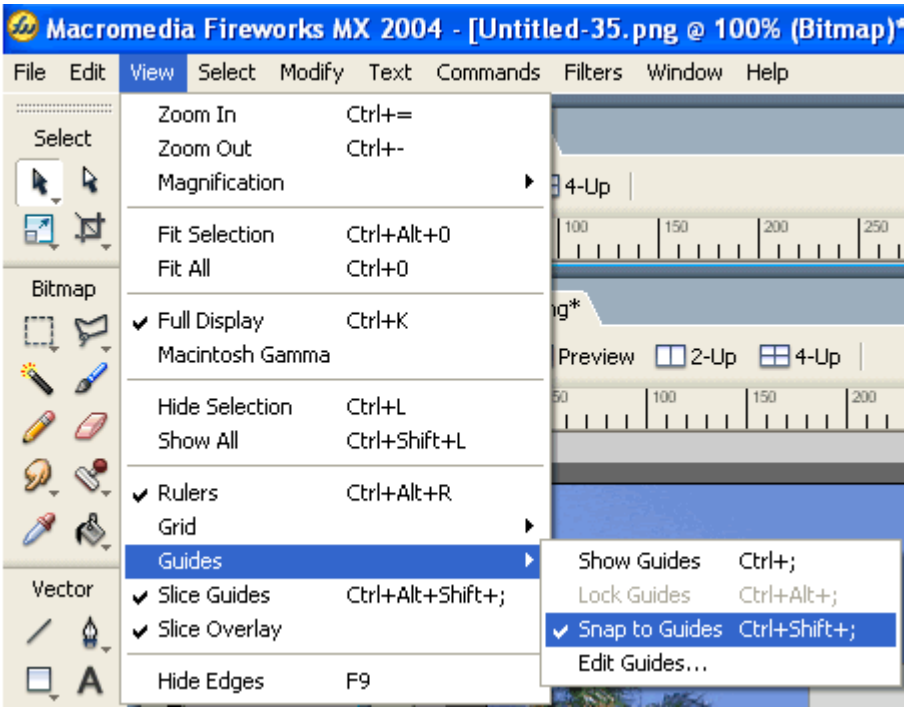
Select the rectangle tool with no stroke...



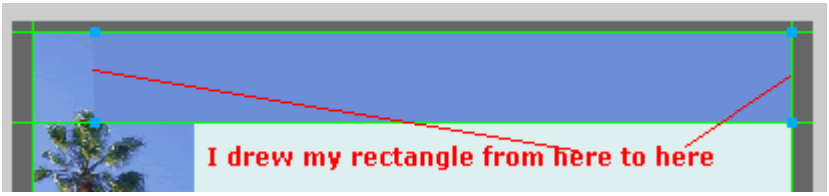
And pick a fill color that matches the sky ...



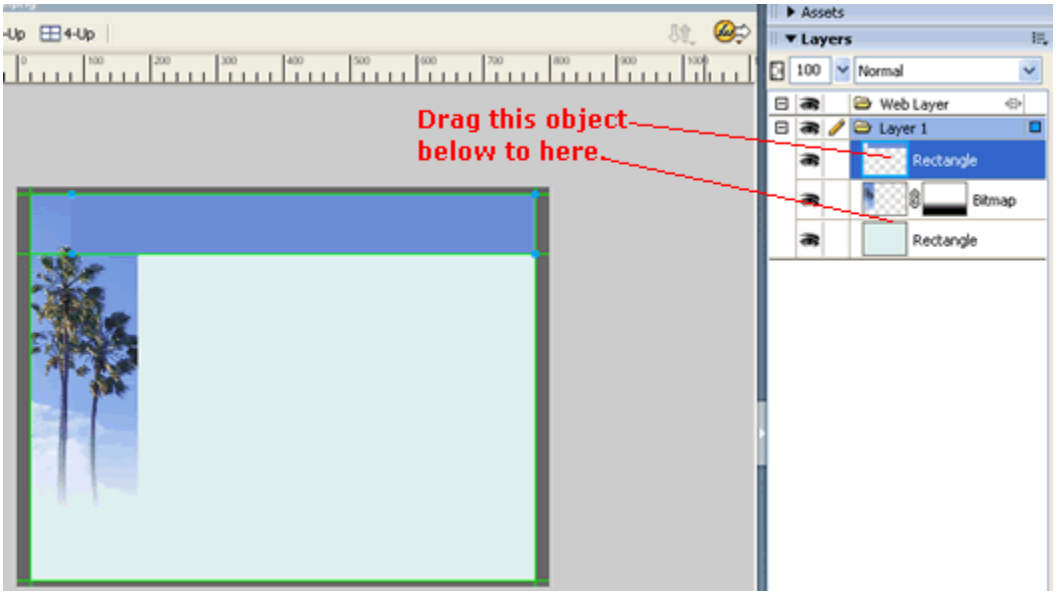
Next make sure that **Snap to Guide** is checkmarked...



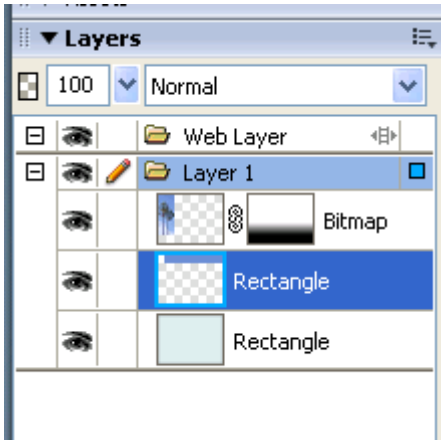
Now draw out our rectangle as shown. Note that we are drawing the header much further into our side image.



This is of no consequence, because we can drag the rectangle object to be below our image.



Now our rectangle is below our image

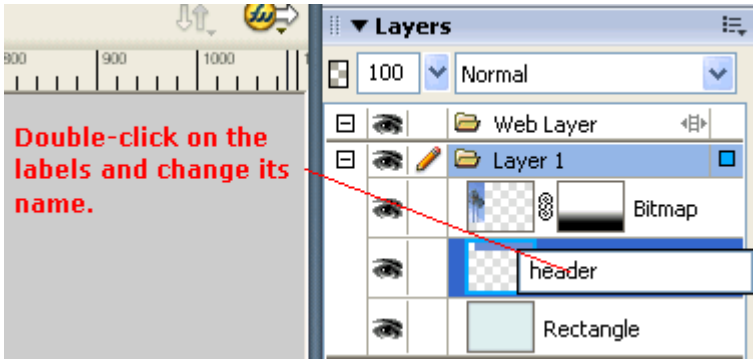


Another reason why I started my rectangle mid-way from the side image is that we have the flexibility to blend the two if needed (as you will see later).

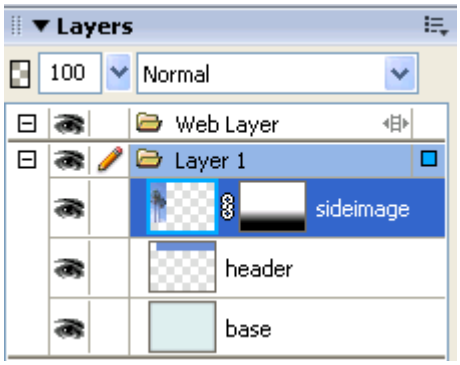
Naming our Layers

Technically speaking, ***layers*** are those folders icons that you see in the Layers panel. See picture above. So **Layer 1** is a layer. And you can have many such layers. Those items inside our folders are ***objects*** in our layer. See that we have a **Bitmap** object on top of two rectangle objects.

Give these objects better names to keep organized. For example...



I have given them the following names...



This is the file that I have so far: **creativeworks.png**

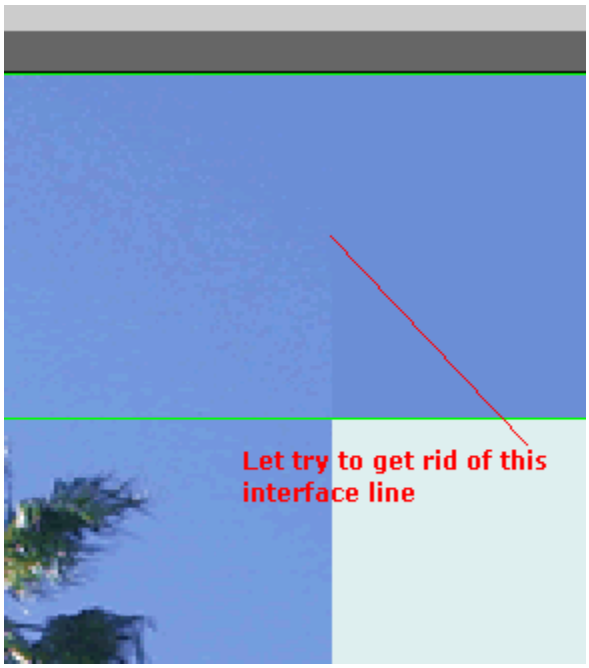
Backing Up Files

Save your file now. In addition to saving the file, you should occasionally make backup copies of past versions so that if we mess up in the future steps, we can always come back to this good version.

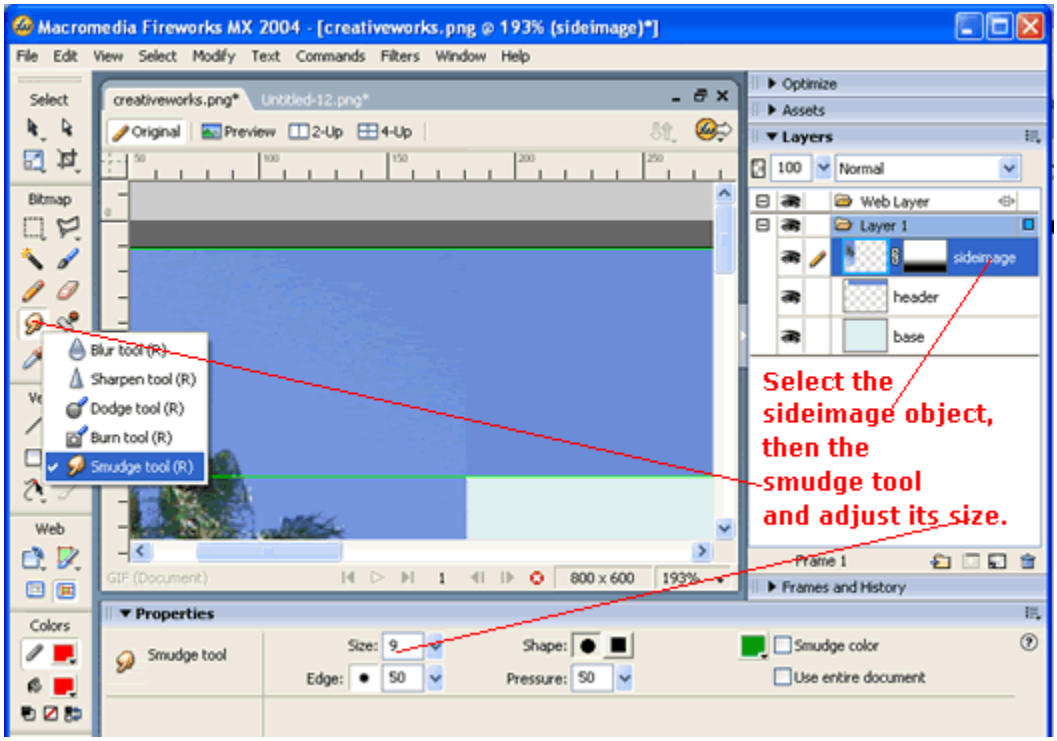
We make backup copies by doing **File -> Save A Copy** and give it a name of **creativeworks_1.png** and put in same folder as your current folder of **creativeworkcomp**. The extra **_1** in the filename is to indicate to me that this is my version 1. Notice that you still have your working copy of **creativeworks.png** opened in Fireworks and this is still the document that we are continuing to edit.

Touch-Up with the Smudge Tool

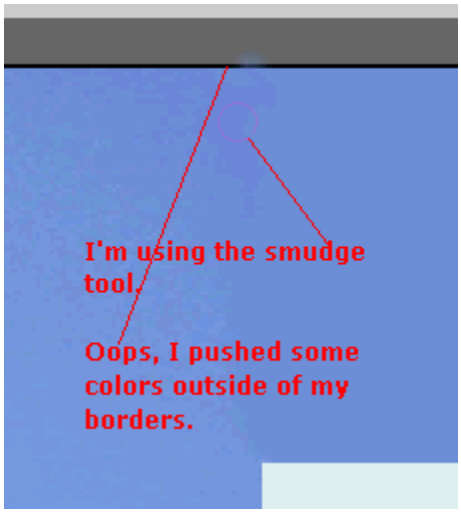
If you zoom in close in the **creativeworks.png** file that we created, you will see that is an vertical interface line between the sky and our blue header.



Try to get rid of it by using the **Smudge** tool with it properties set as shown.

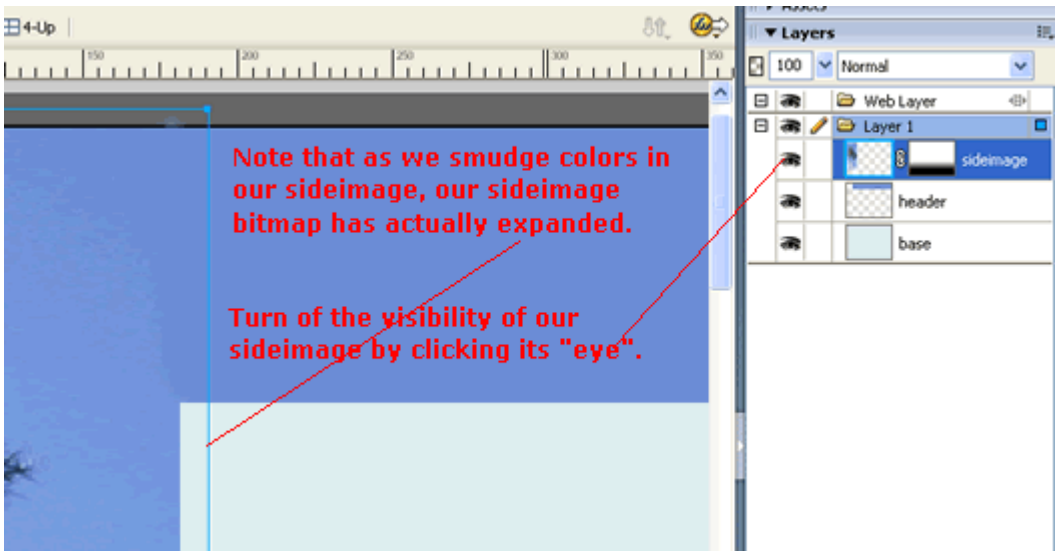


And just smudge around the edge there.

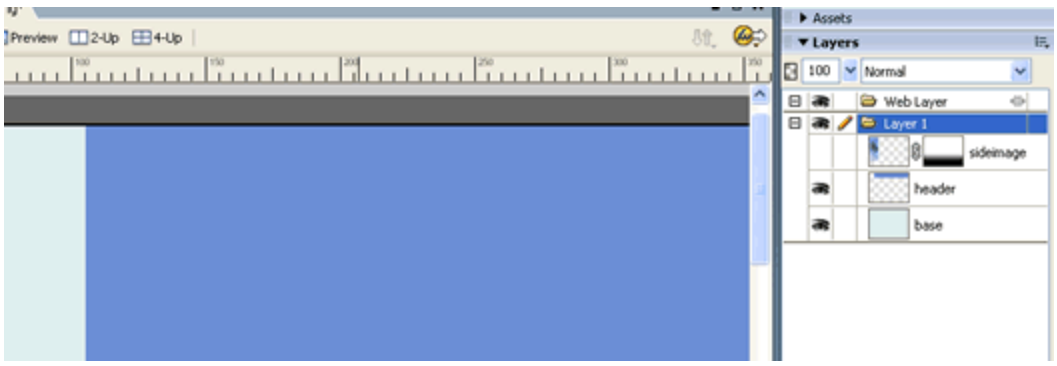


If you mess up, do **Ctrl-Z** to undo your strokes. If you pushed some colors outside the borders, don't worry about it, we take care of that later.

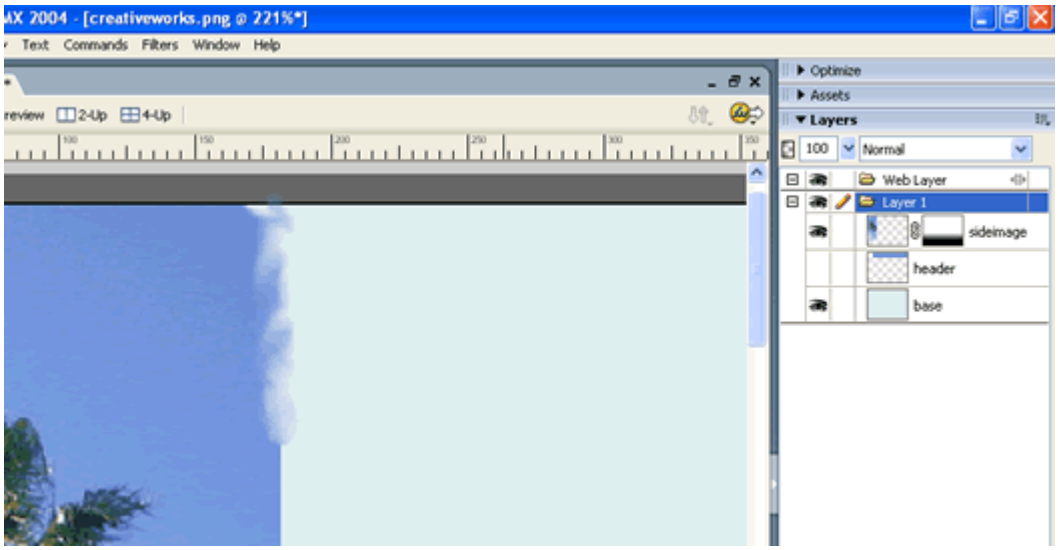
Because we had our sideimage object selected while we were doing the smudging, this will only affect our sideimage and not our header at all. If you turn off the visibility of our side image...



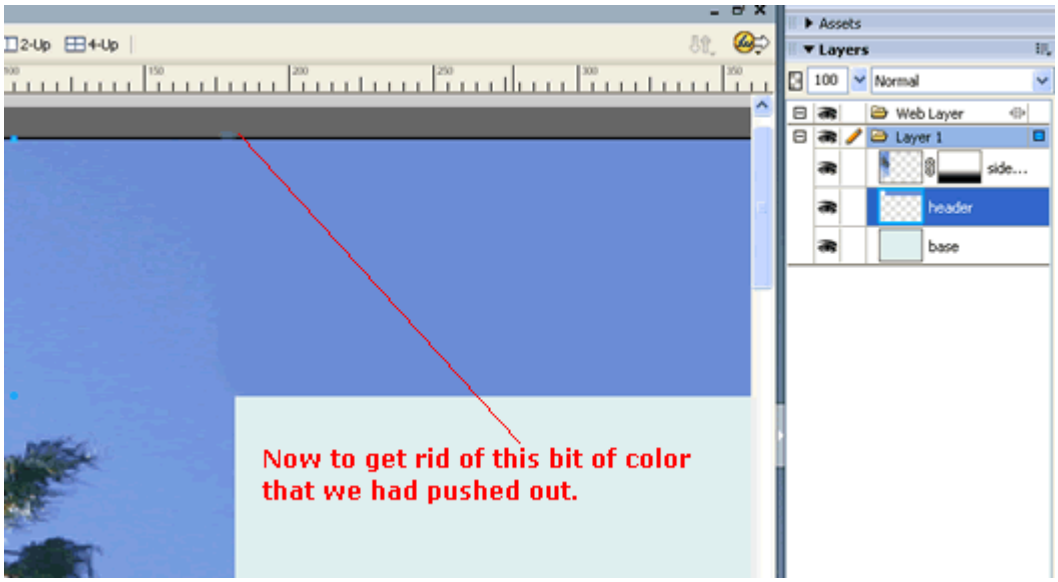
you can see how our header is unaffected...



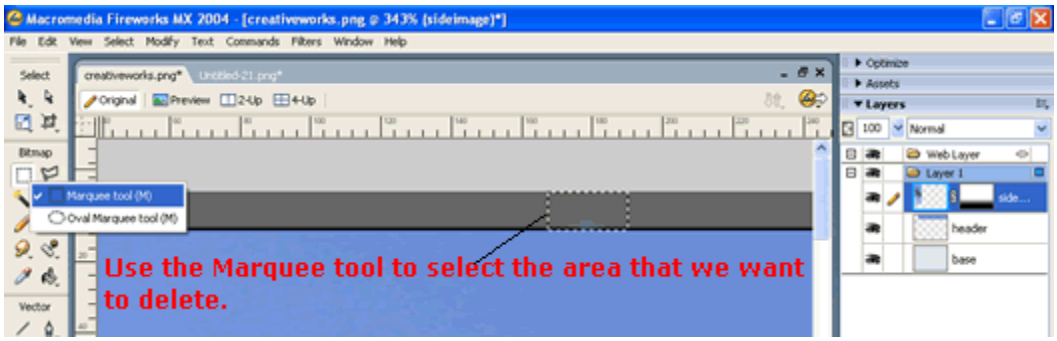
Now if you turn on our sideimage and turn off our header, you can see our smudging handiwork...



We had actually erased some of the colors from our image. But since we have our header underneath that extends all the way to the middle of our side image, no one will know the difference (when we turn on visibility on all layers).



To get rid of this bit of color, recall that this bit of color is on the sideimage object. So select that in our layers panel and marquee out a selection as shown.

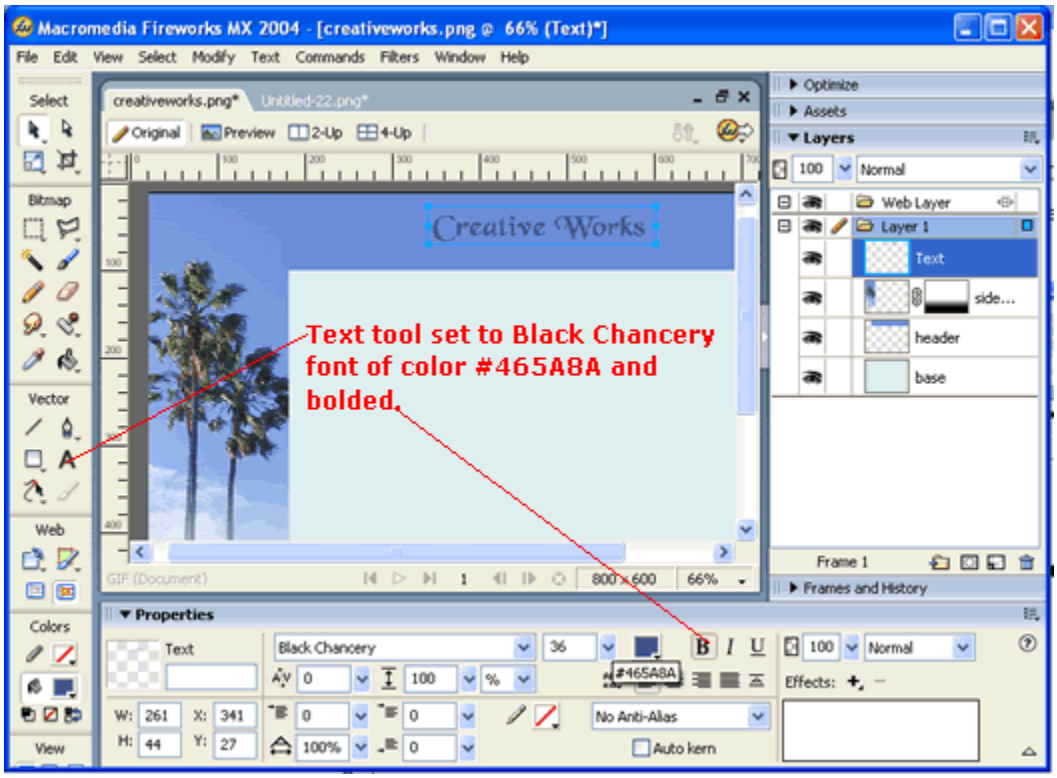


Press the **delete** key and the bit of color is gone. To deselect our marquee so that we don't keep seeing these ***marching ants*** (that is what they call that selection outline) do **Ctrl-d**.

Good time to **Save**.

Writing Out Our Title

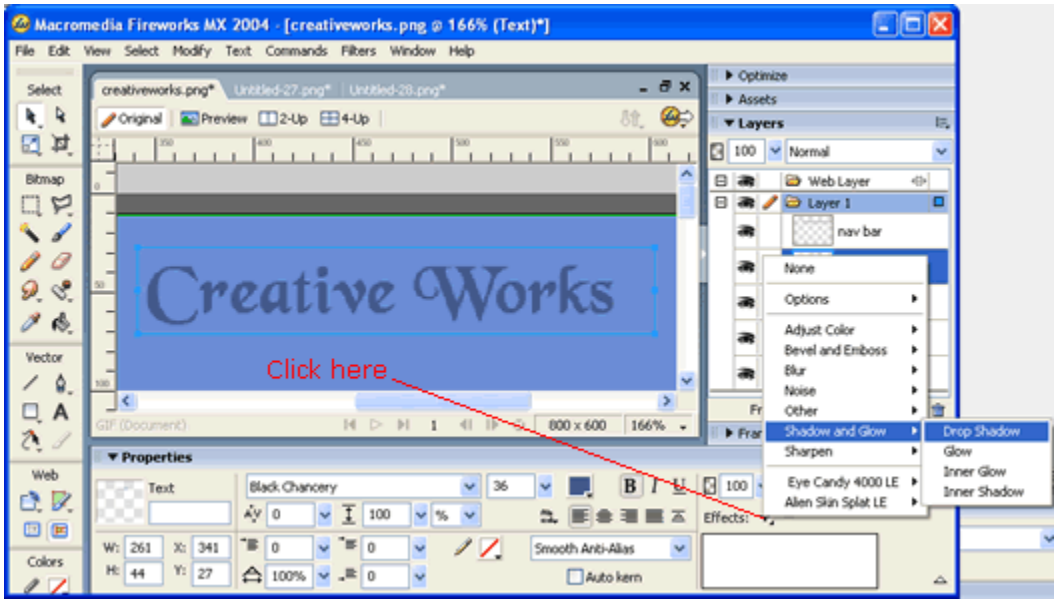
Use the **Text** tool to type out our header with the property setting shown. I used a more distinctive font called **Black Chancery**. You can use any other font if you like.



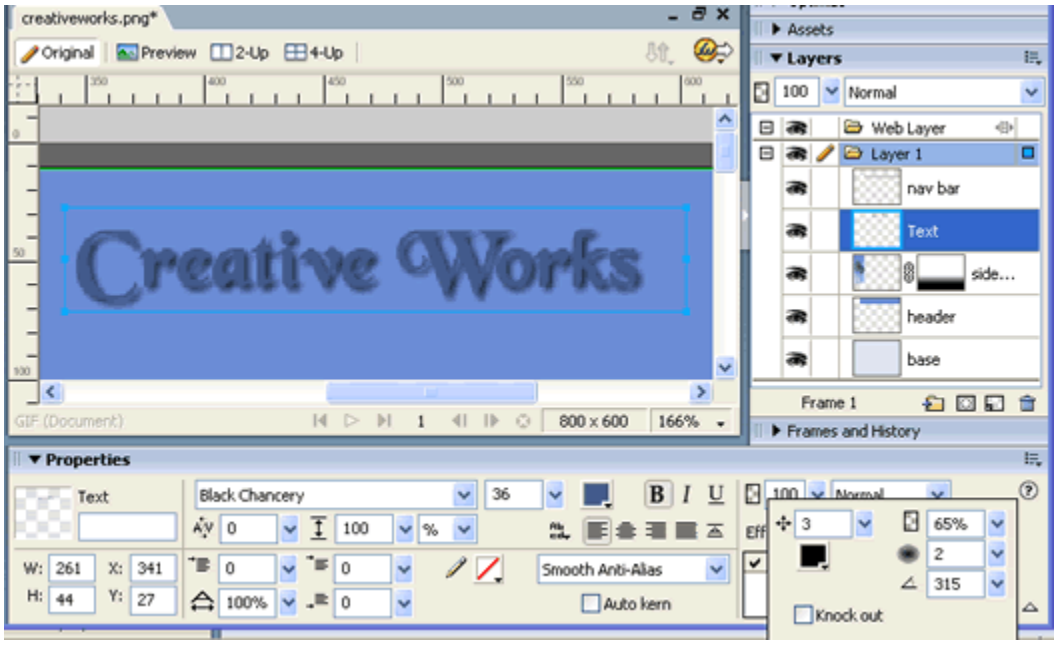
Note that in our **Properties** window, we have **No Anti-Alias** set (see above picture). And if we zoom in closer, we see...

If we switch to **Smooth Anti-Alias**, the text will blend better with the background.

Our text is very flat. Spruce up our title with a drop shadow by clicking the plus button under **Effects** for the property of our text and selecting **Shadow and Glow -> Drop Shadow**.

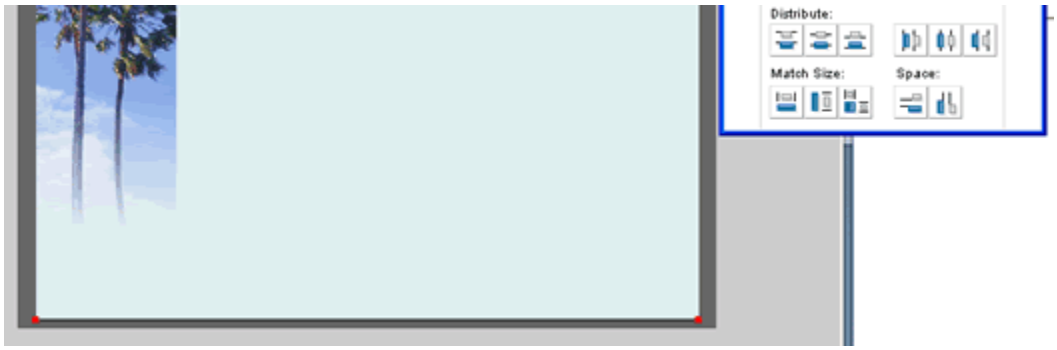


Play around with the setting to find one that you like. I set mine to be ...



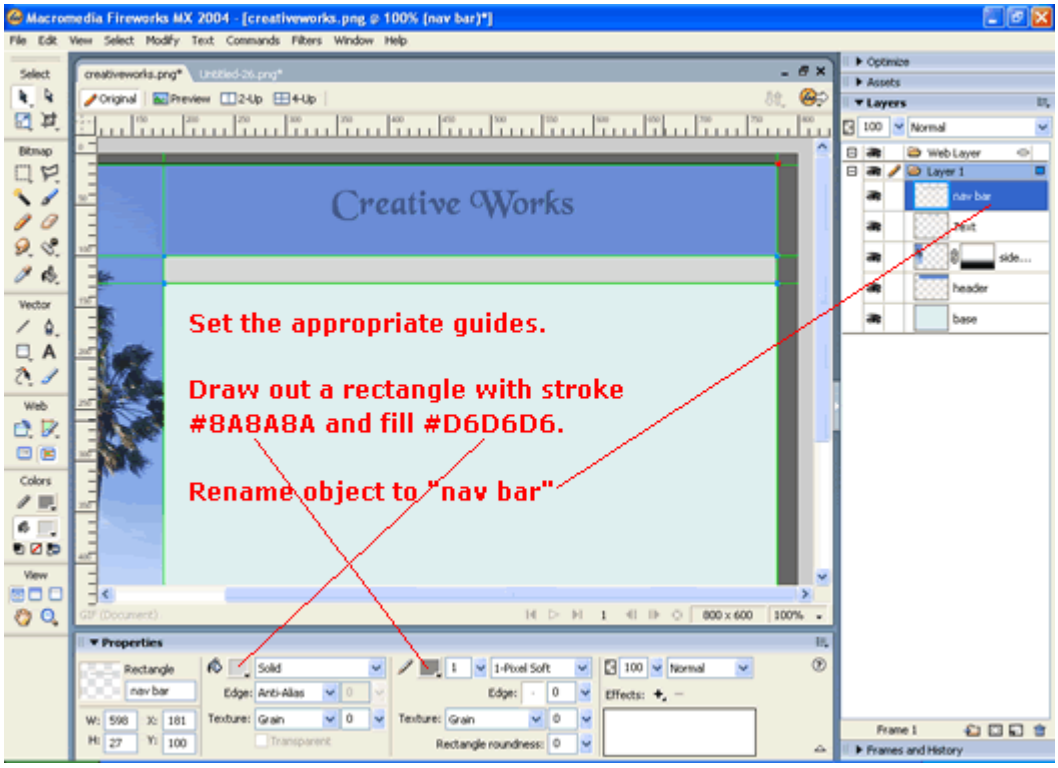
It should be subtle, just enough to give a bit of dimension to the text.

To center our title across our page, select the title and center align it to the canvas by menu **Windows -> Align**.



Creating the Navigation Bar

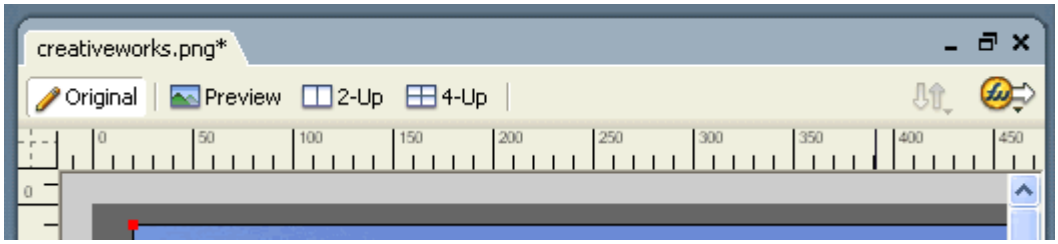
It is time to draw out our navigation bar using the two gray color as shown...

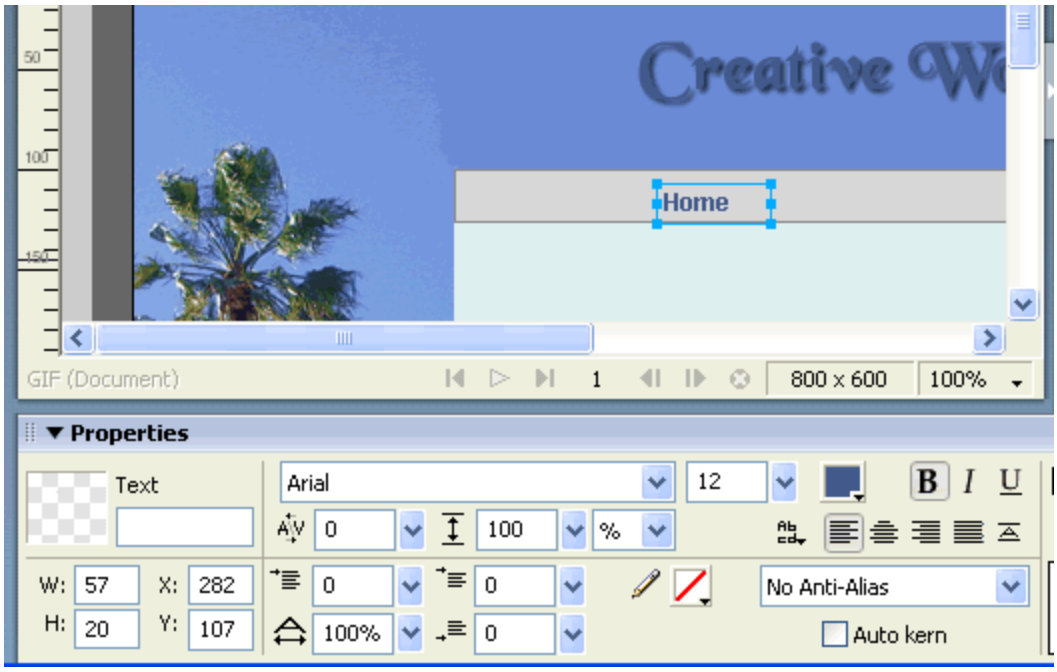


Why did I pick these colors? No reason, they were suggested by the **ColorMatch5K** web color tool after inputting an initial color of **#6D8DD6** (which is our blue header color, which in turn was determined by our sky in the image). And of the two grays, I used the darker for the border and the lighter for the fill. It is typical to darker borders in order to provide definition and boundary.

Adding Text to Nav Bar

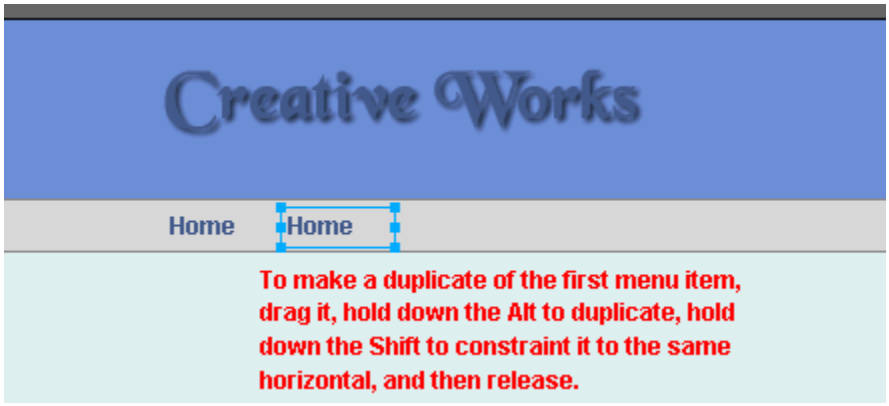
Drop in some text menu items for our nav bar. When we code this website in the HTML/Dreamweaver lessons, the text will be acutal text and not graphical images. Hence we picked **Arial** which is a font that is common on most users' system.



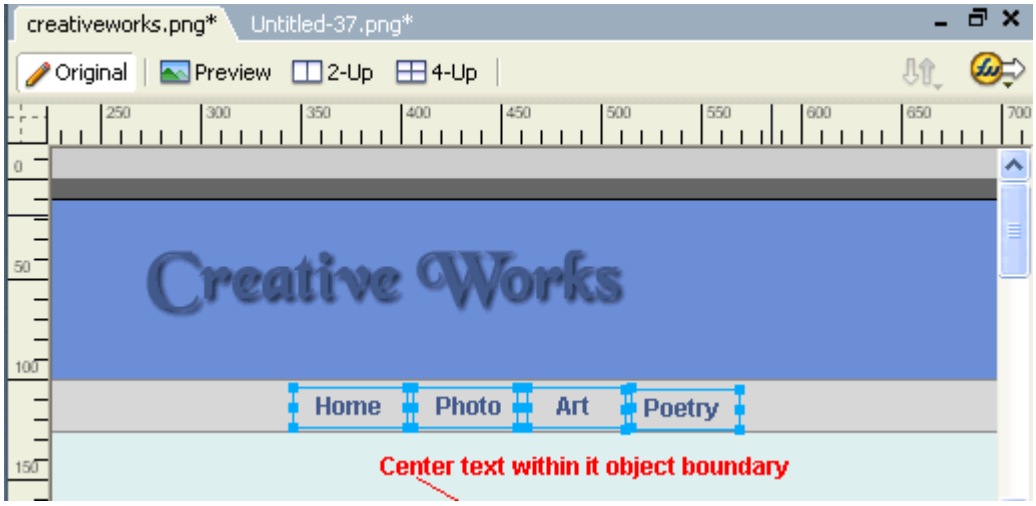


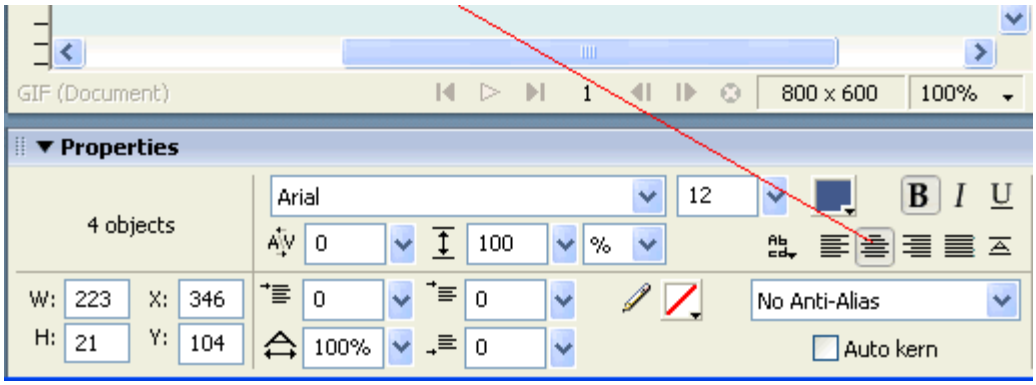
If you find that the text is snapping to certain location and you can not position it to where you want, make sure your **Snap to Guide** is turned off.

Duplicate our first menu...



until we have four. Then with the text tool selected, you can double-click on the text to change them to the following...





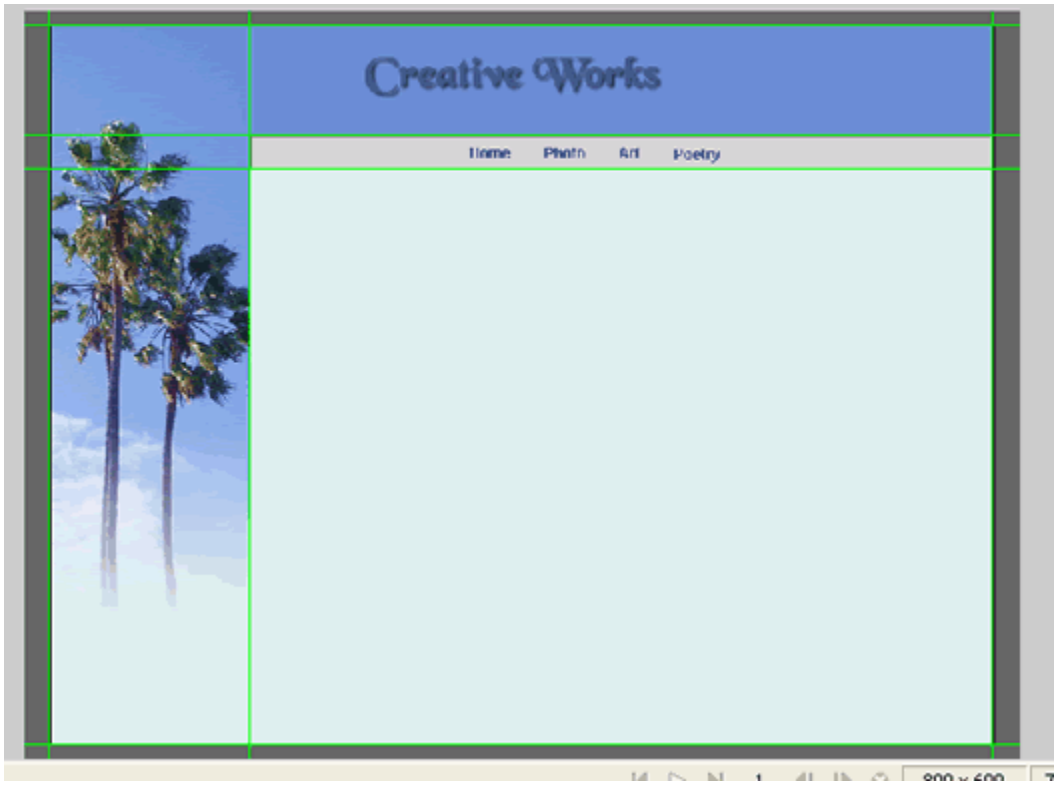
For each of the text, you can have them be centered-aligned with the center button in the property panel as shown.

With all four text selected, you can **Modify -> Align -> Distribute Widths** to make them look neat.

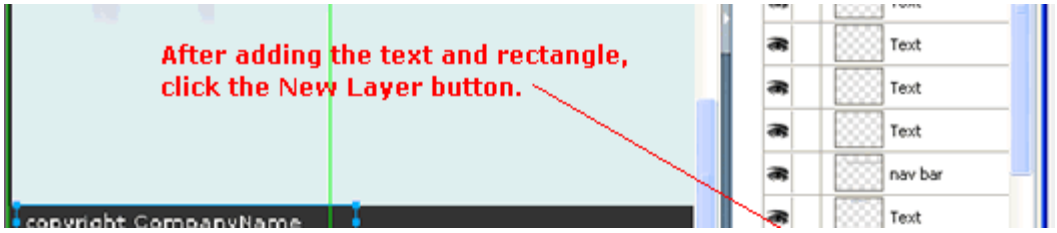
This is what we have so far: **creativeworks.png**

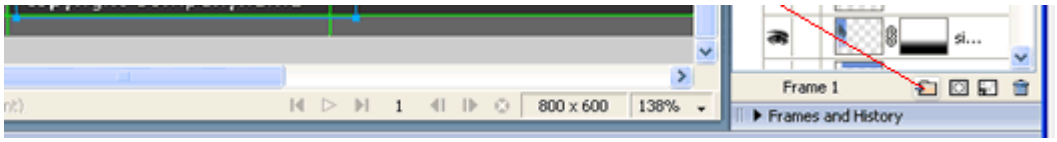
Creating the Footer

Here is what we have so far...

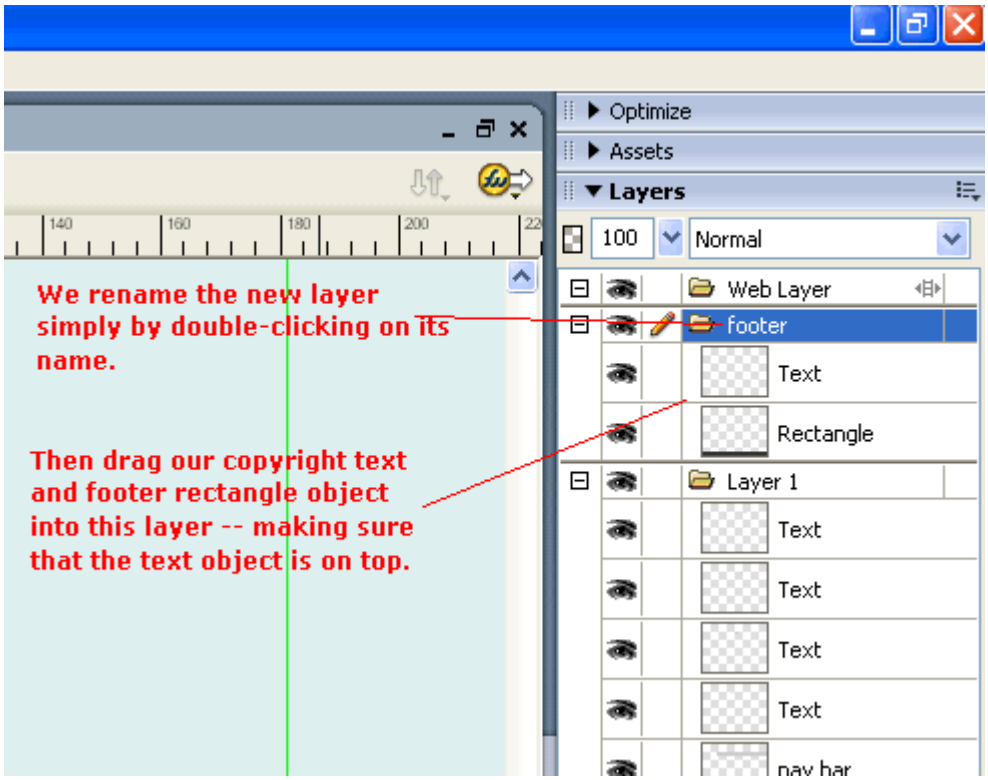


Typically, a website will have a footer element. Using guides, draw a rectangle of color **#333333** and put on top of it white text in **Verdana** font that says "**copyright CompanyName**" as shown...





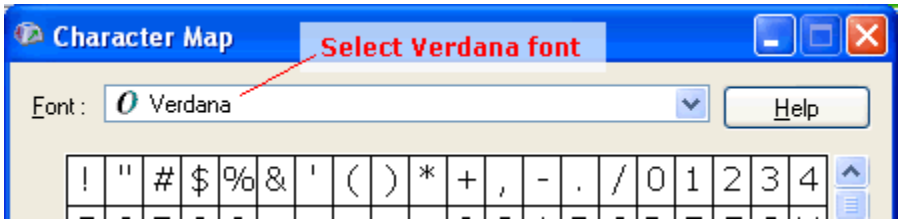
Then create the **New Layer** button to create a new **layer folder** and name it **footer**. We are trying to be more organized now.

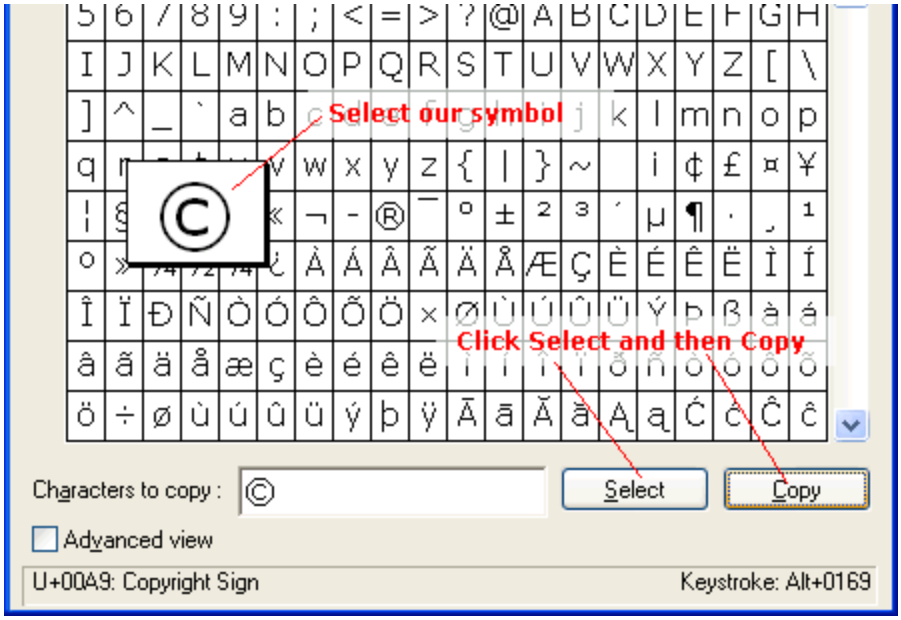


Special Symbols

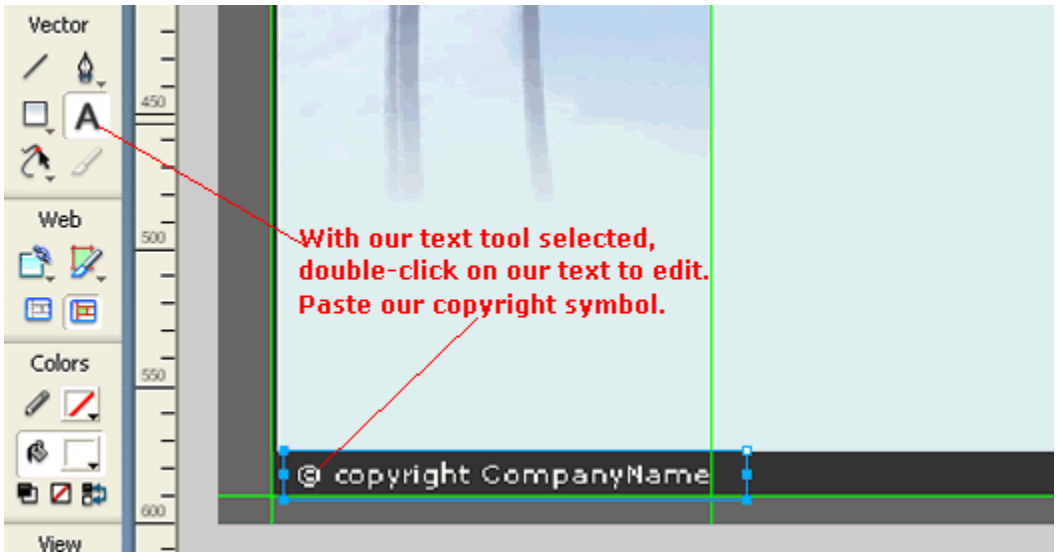
What if we want our footer to have the special copyright symbol such as ...

Windows has a program called **Character Map** that can give us these symbols. Go to Windows **Start Menu -> All Programs -> Accessories -> System Tools -> Character Map**. Since our text was in Verdana font, we select that font in Character Map, click on the copyright symbol that we desire, click **Select**, and then click **Copy**.



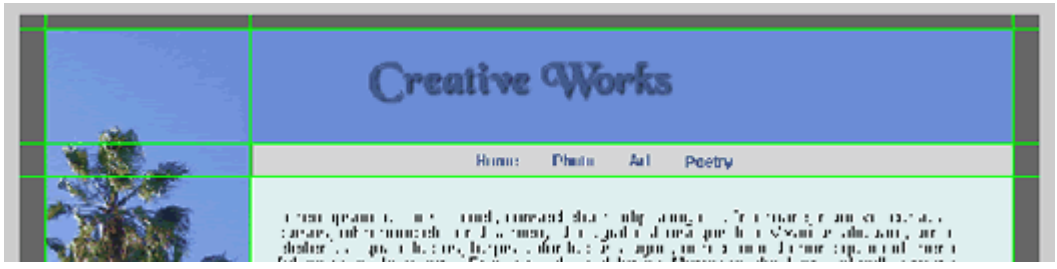


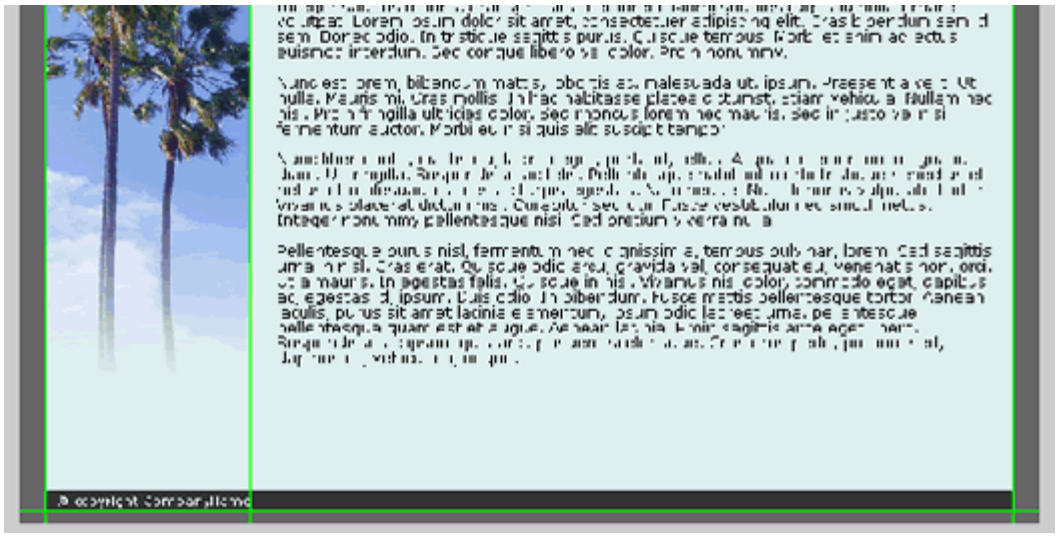
Now this special character is in Window's **clipboard**. So now we can edit and paste this symbol in our copyright text as shown.



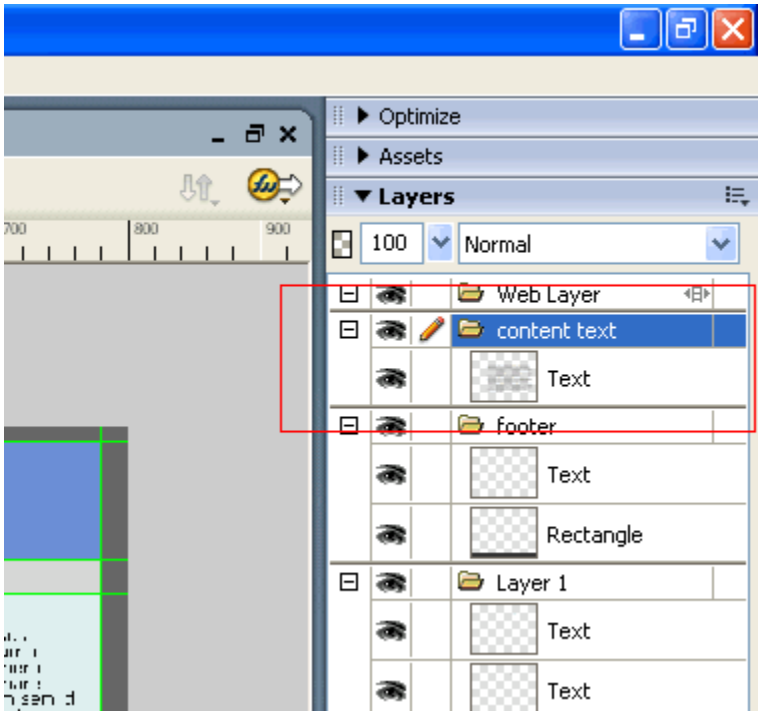
Adding Lorem Ipsum Text

Now we just need to fill our empty page with some fake content to make it look like a real page. Often designers want to get a sense of what a page will look like with some content text in it. But when content text is not ready and they don't want to make some up, they will use **Lorem Ipsum** text. These are just dummy filler text, but the characters are statistically choosen to make it look like an actual language. Using the **Lorem Ipsum generator**, we generated four paragraphs and pasted it to our page. And this is what our page looks like ...

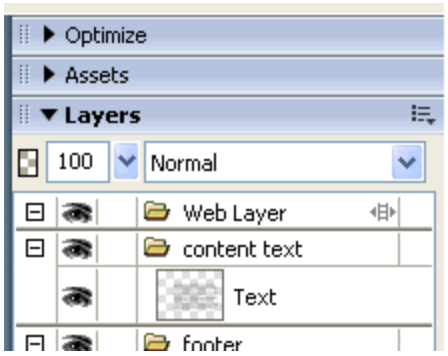


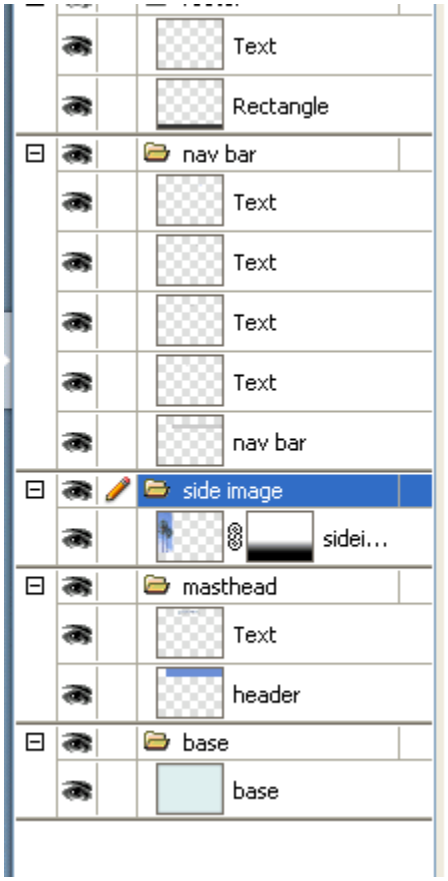


Since we had been working in our **footer** layer last when we added our text, that text object went into our **footer** layer. But since that is not really a part of our footer, we better put it in its own layer called **content text** as shown.



Similarly, organize the rest of our objects as shown below. You can see which objects goes with items in the design view by clicking on and off the **eye** icon.



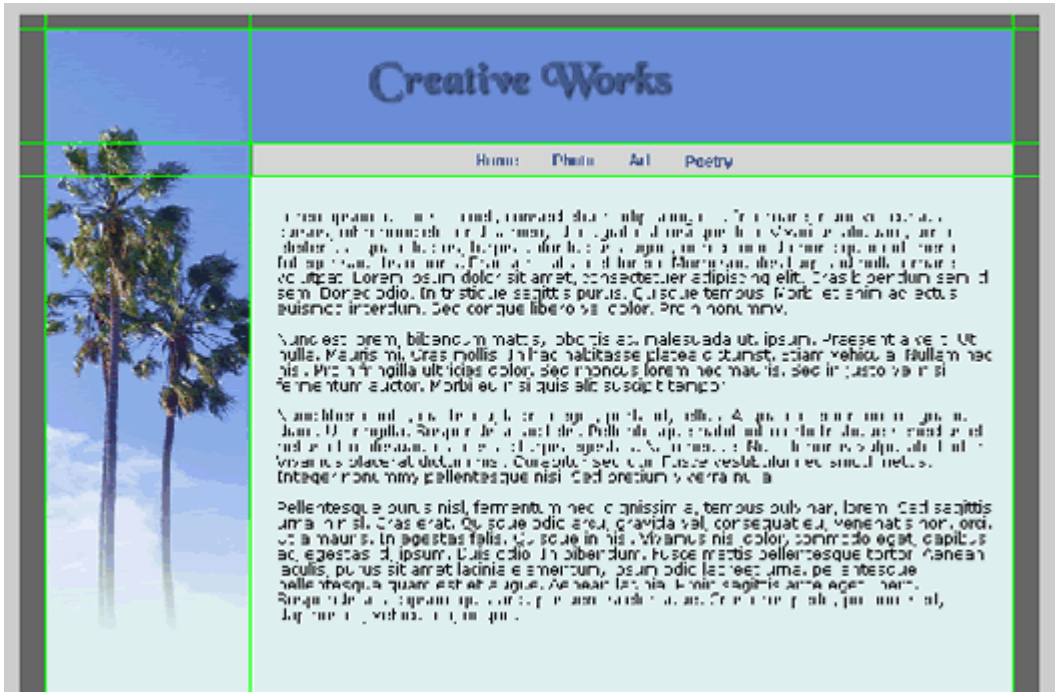


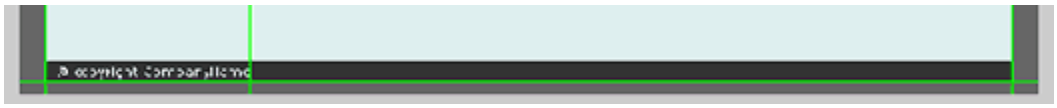
Make sure that the **side image** layer is above the **masthead** layer.

We are now done with the design, and you can get the file here: **creativework.png**.

Slicing the Side Image

Here is what we have so far...



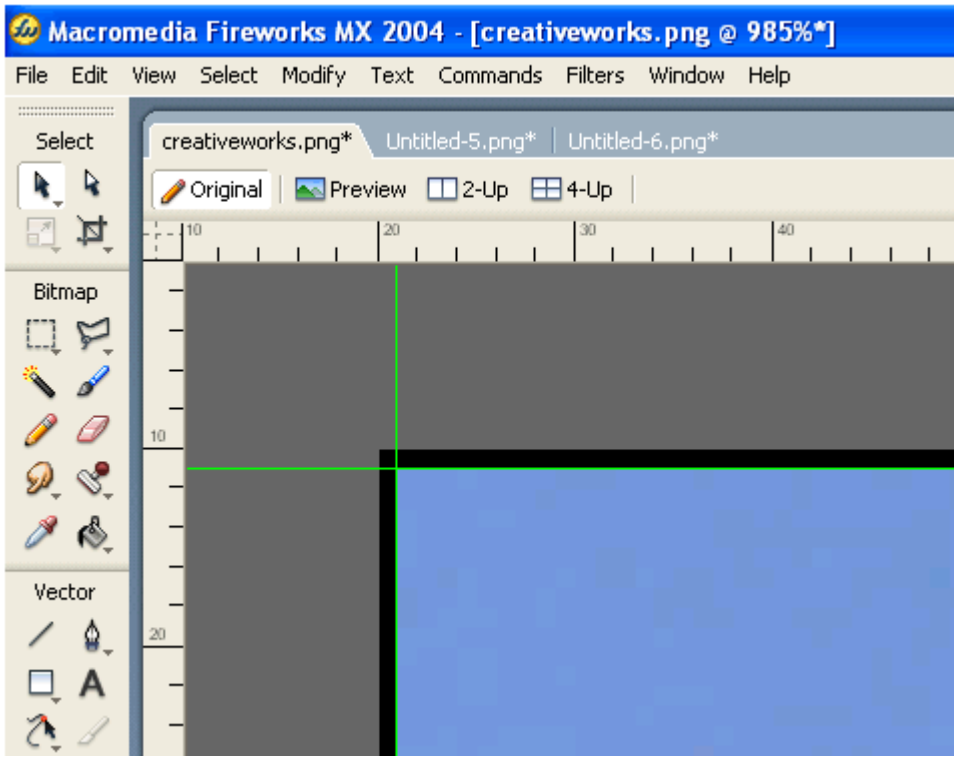


When creating a website, we have to determine which part of the comp will be implemented as a graphics and which part of the comp will be implemented in HTML. The side image for example, will be implemented as an image. Because we can not do drop shadows in HTML and the text font is in an distinctive font of Black Chancery that most user would not have on their system, the title will also be implemented as a graphics.

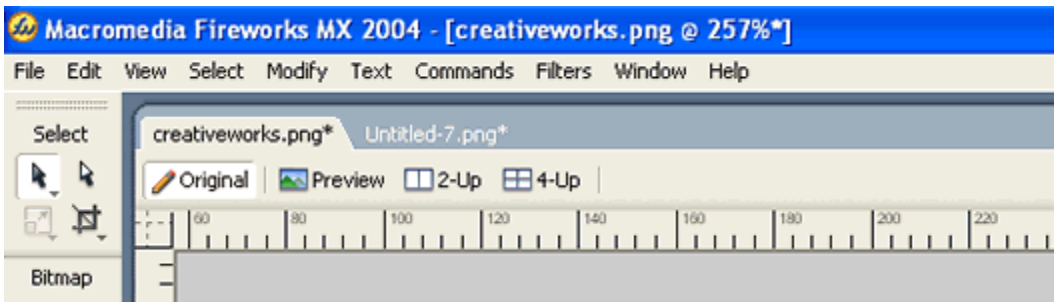
The navigation menu bar is a solid color and the menu items are just text using a common Arial font. This can be easily done in HTML. And hence the nav menu will not be implemented as a graphic. Plus it contains link and interactivity that needs to be implemented in HTML. The content text is just text and should also be implemented in HTML. Same for the footer.

So the only graphics that we will need for this website is the side image and the title.

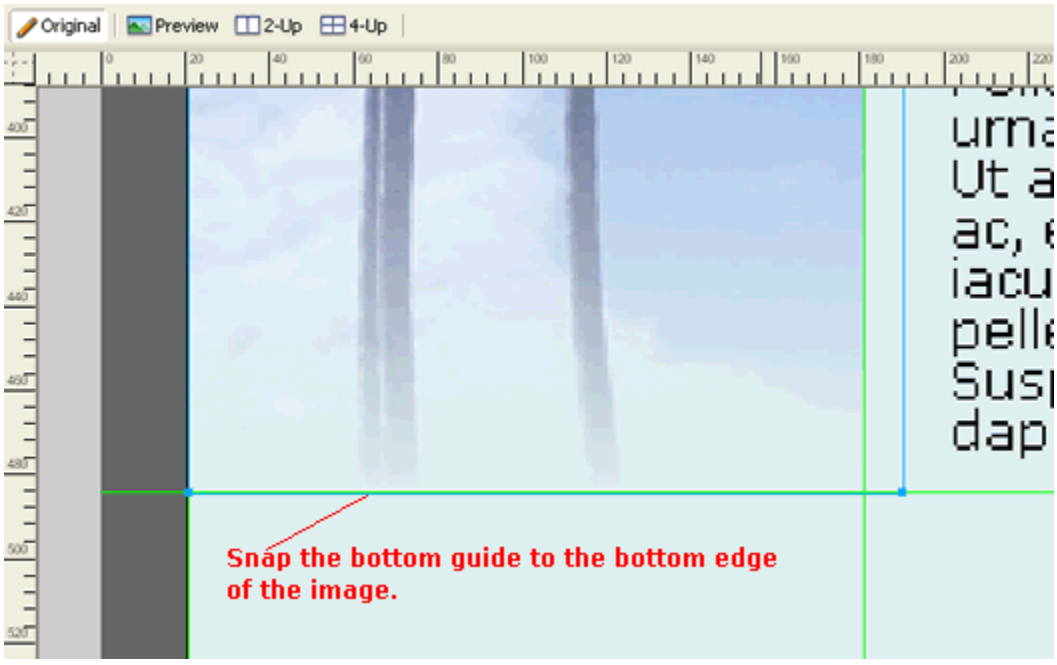
We need to ***slice*** out these images. In the industry, we call this stage ***slicing*** the website. We will slice out the side image first. We need to slice to pixel accuracy; no rough eye-balling. Hence we need to use guide. Zoom in close and adjust the guides onto the left edge and top edge of the image as shown below. We don't want the black border to be part of the image that we are slicing out.



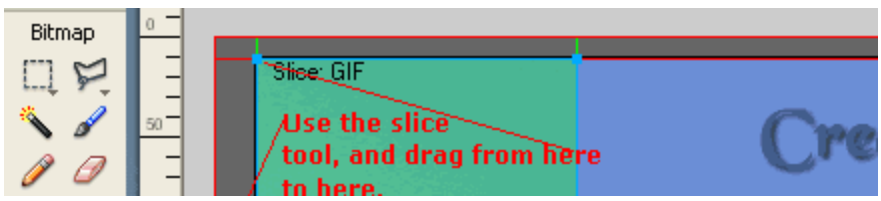
Similarly, adjust the right guide as shown...

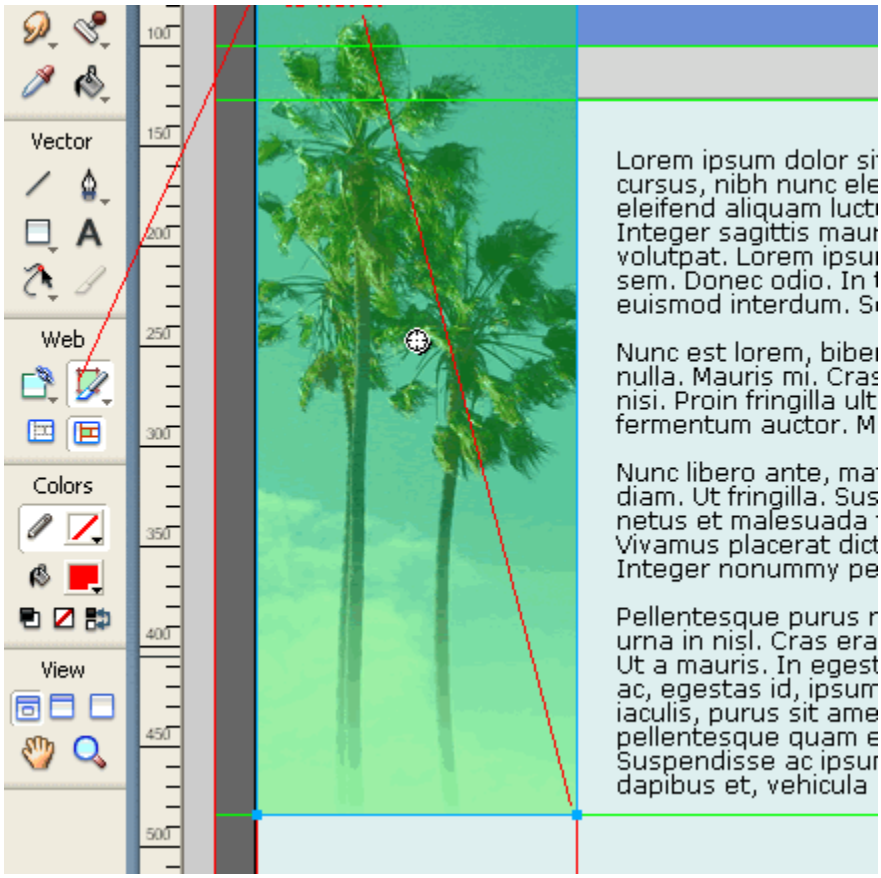


With the side image selected as shown below, you can snap the bottom guide to the bottom edge of the image

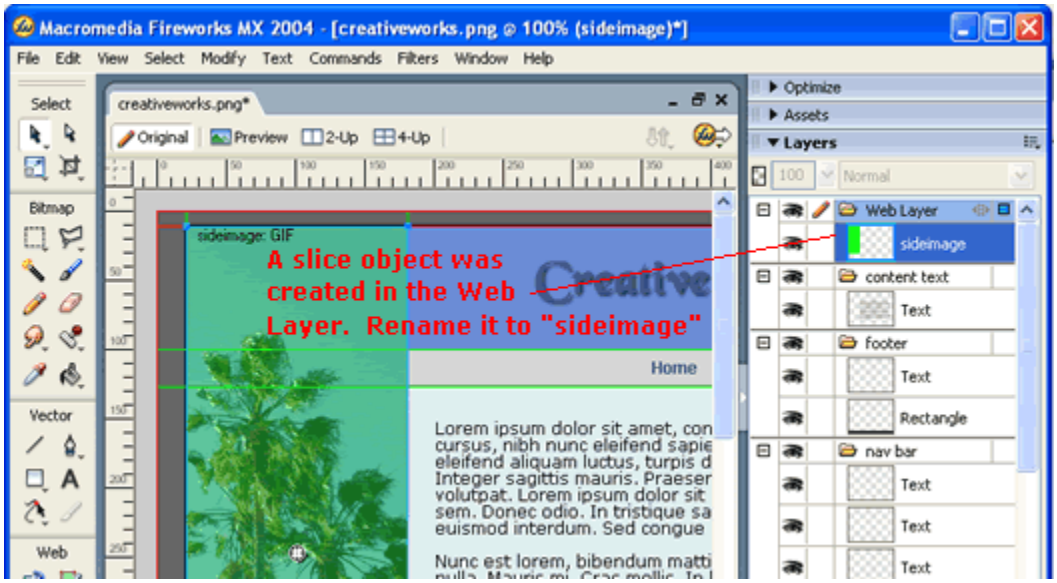


Now that the guides are in place, we are ready to slice. Make sure that **View -> Guides -> Snap to Guides** is checkmarked. Then use the slice tool to slice and drag out the area as shown.



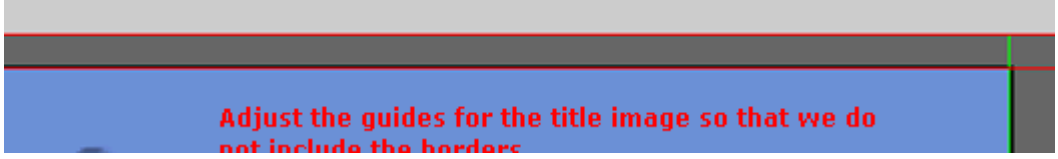


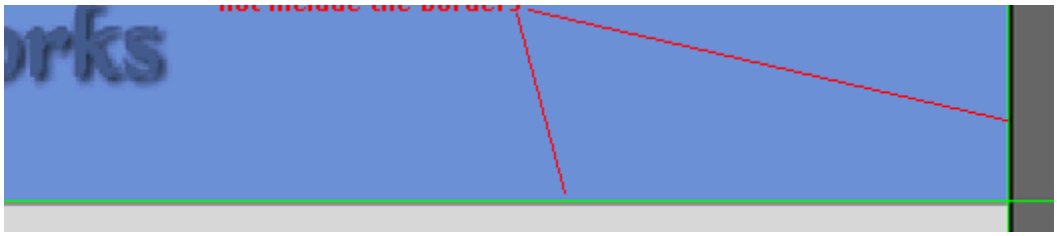
Note that an new slice object was placed in the **Web Layer**. Rename that object to be called "**sideimage**".



Slicing the Title

Similarly, we adjust the guides for the title image...

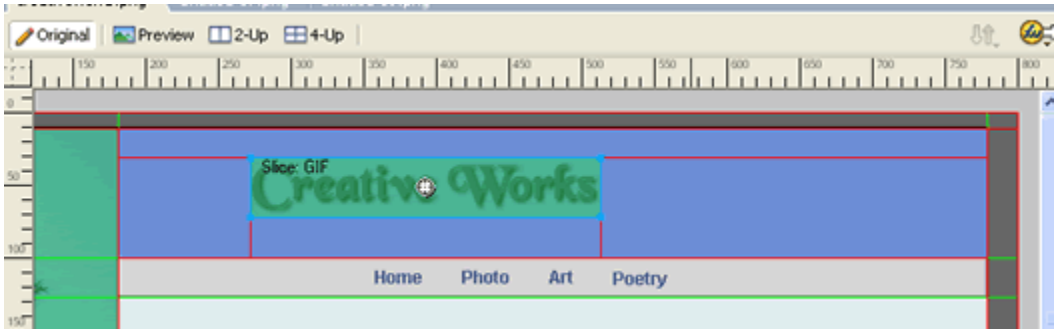




and slice out the below area, renaming it to "titleimage".

You should now have something like my file here: **creativeworks.png**

We had sliced the whole rectangular area of the masthead to be the titleimage slice even though many areas of it is simply solid color that can be easily implemented in HTML. A rule of thumb for quick loading of webpages is to have as few and as small images as possible. So ideally, we should have sliced the title image as ...



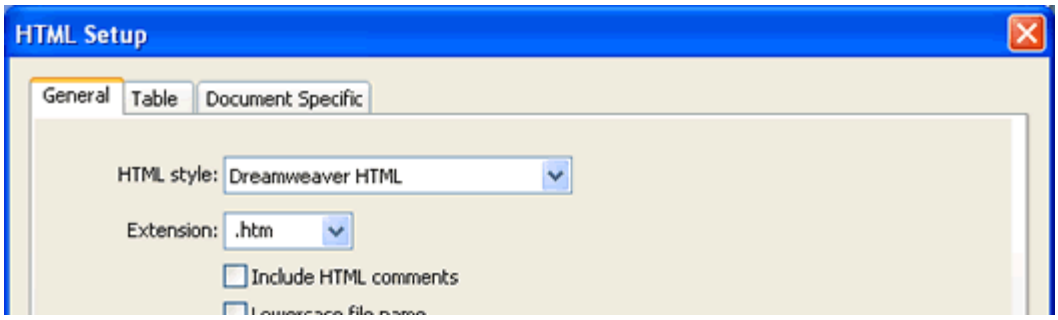
However, that would mean more complicated HTML coding (which we will do in the Intermediate Design Series). But for the Intro Series, let's keep things simple for now and leave the title image to be the whole masthead.

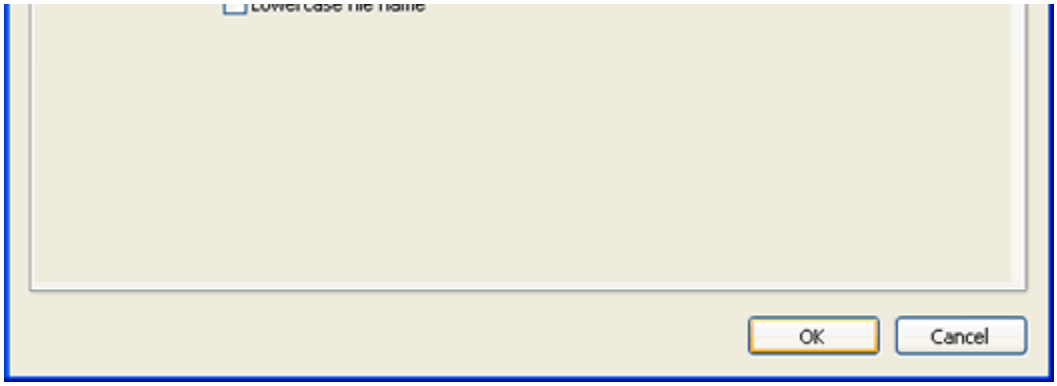
Exporting

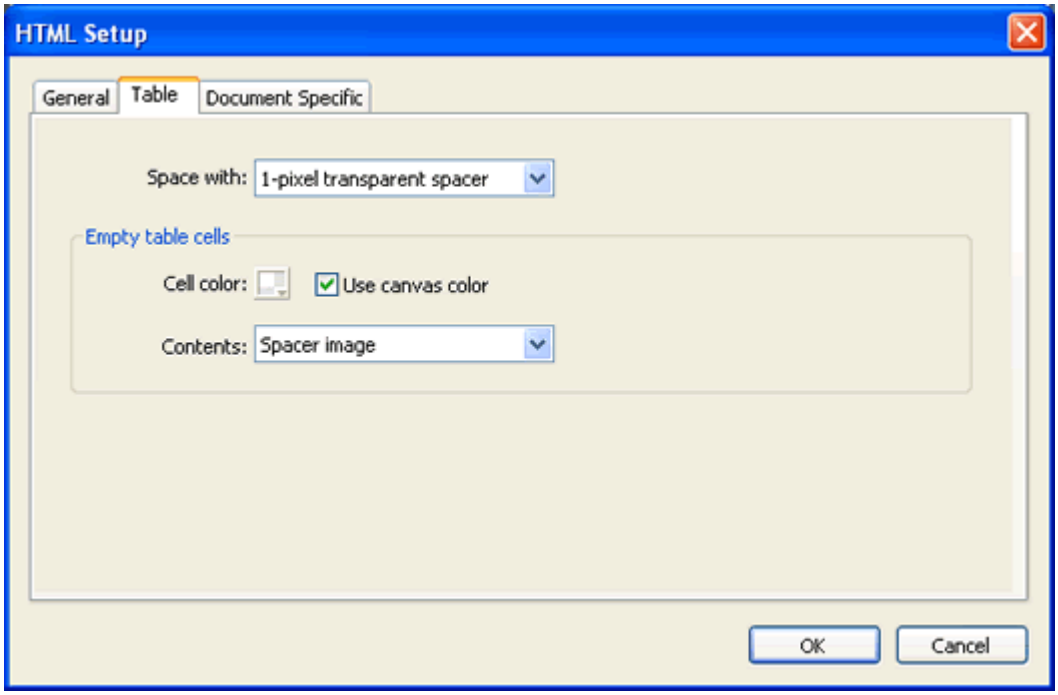
After we have sliced our comp, we now want to export those slices.

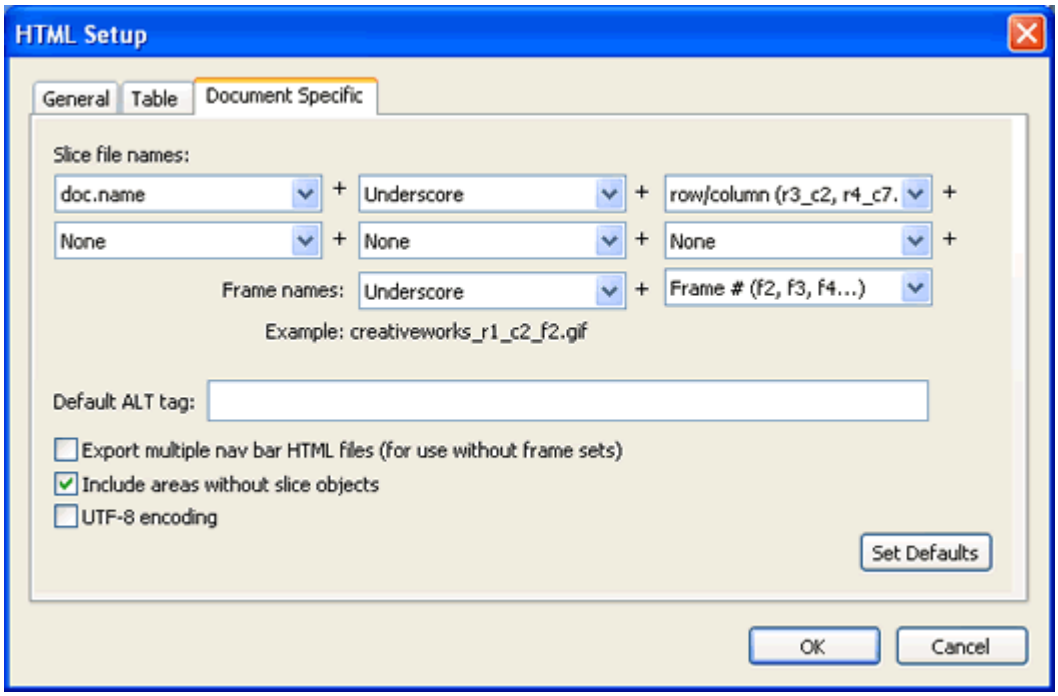
Fireworks does have a facility to export this comp into a HTML page. However, this technique is very rudimentary and is intended for only for the most basic of web pages. In fact, I generally don't recommend this method at all. Because the code that is generated is sub-optimal. And in the Intro HTML/Dreamweaver course, I show you how to hand code much more optimal code. Nevertheless, you should at least know that such as feature exists. And so we do it now.

In Fireworks, with the **creativework.png** file opened, do menu **File -> HTML Setup** and keep the default settings as shown...

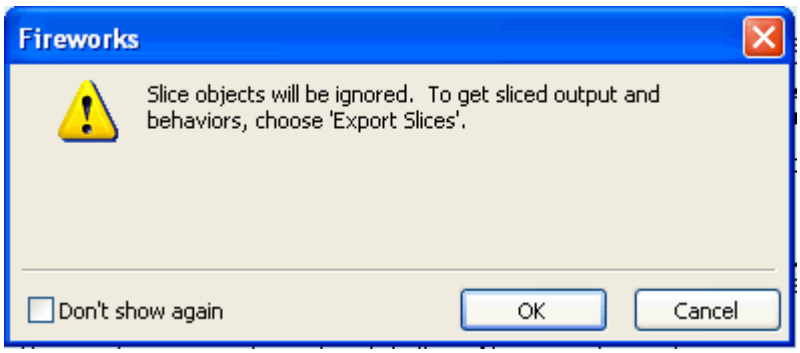



OK
Cancel

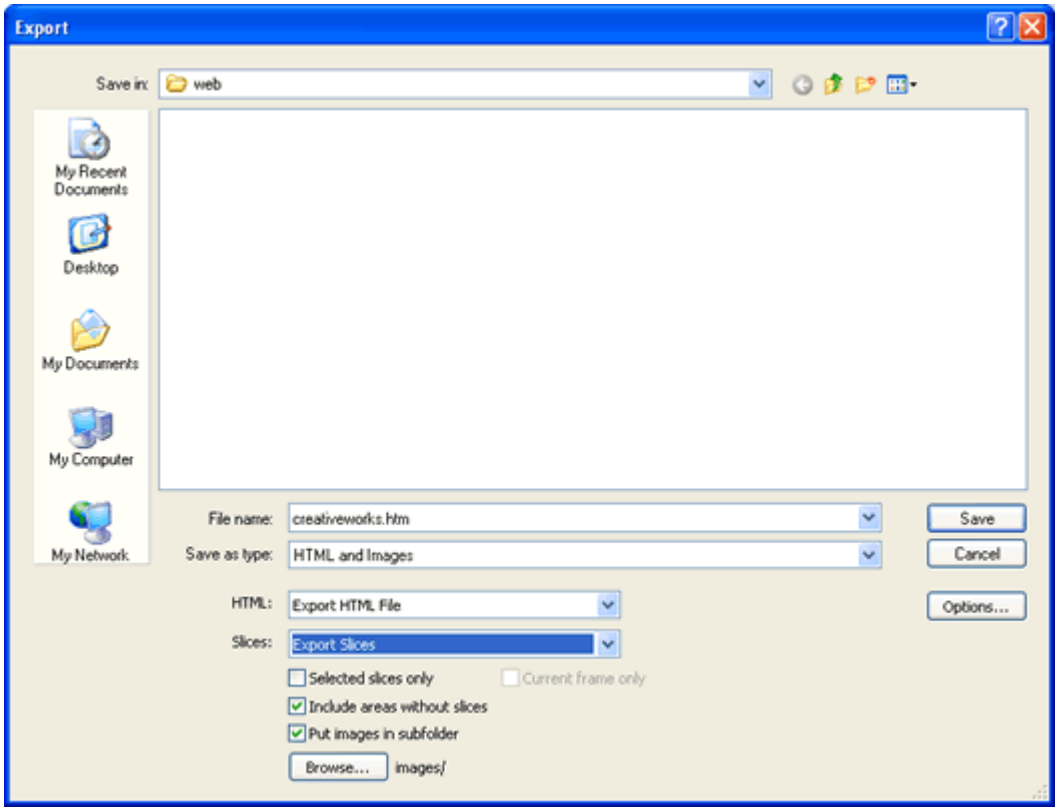

HTML Setup
General
Table
Document Specific
Space with:
1-pixel transparent spacer
Empty table cells
Cell color:
Use canvas color
Contents:
Spacer image
OK
Cancel


HTML Setup
General
Table
Document Specific
Slice file names:
doc.name
Underscore
row/column (r3_c2, r4_c7.
None
None
None
Frame names:
Underscore
Frame # (f2, f3, f4...)
Example: creativeworks_r1_c2_f2.gif
Default ALT tag:
Export multiple nav bar HTML files (for use without frame sets)
Include areas without slice objects
UTF-8 encoding
Set Defaults
OK
Cancel

Now do **File -> Export**. If you get this dialog, click **OK**.

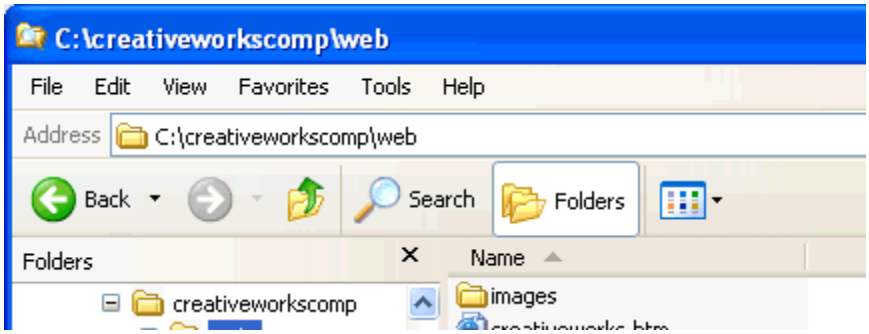


In the **Export** dialog, provide the filename of **creativeworks.htm** in a new folder called **web** under the **creativeworkscomp** folder. Set the other settings of the dialog as shown.

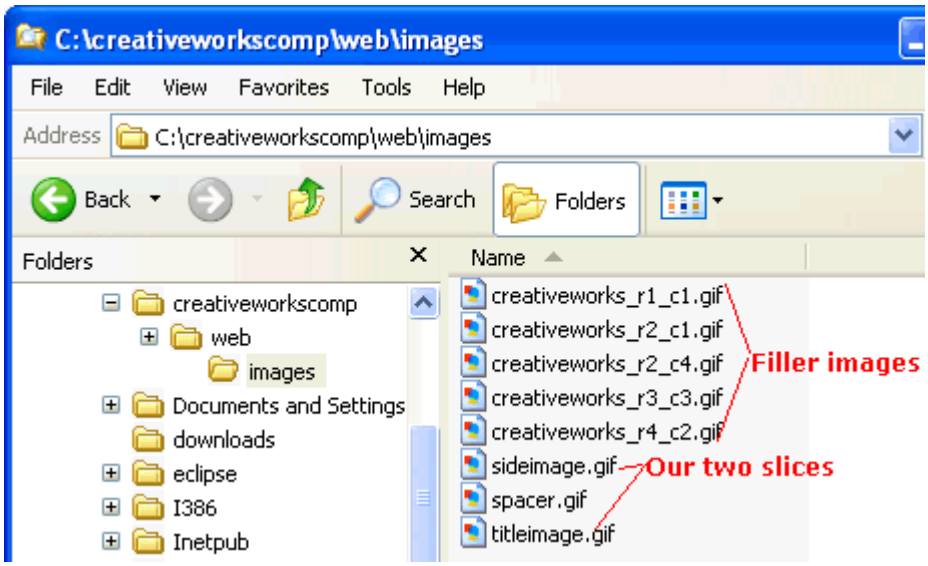


Click **Save**.

Using Window Explorer, you will find that Fireworks has generated the **creativeworks.htm**.

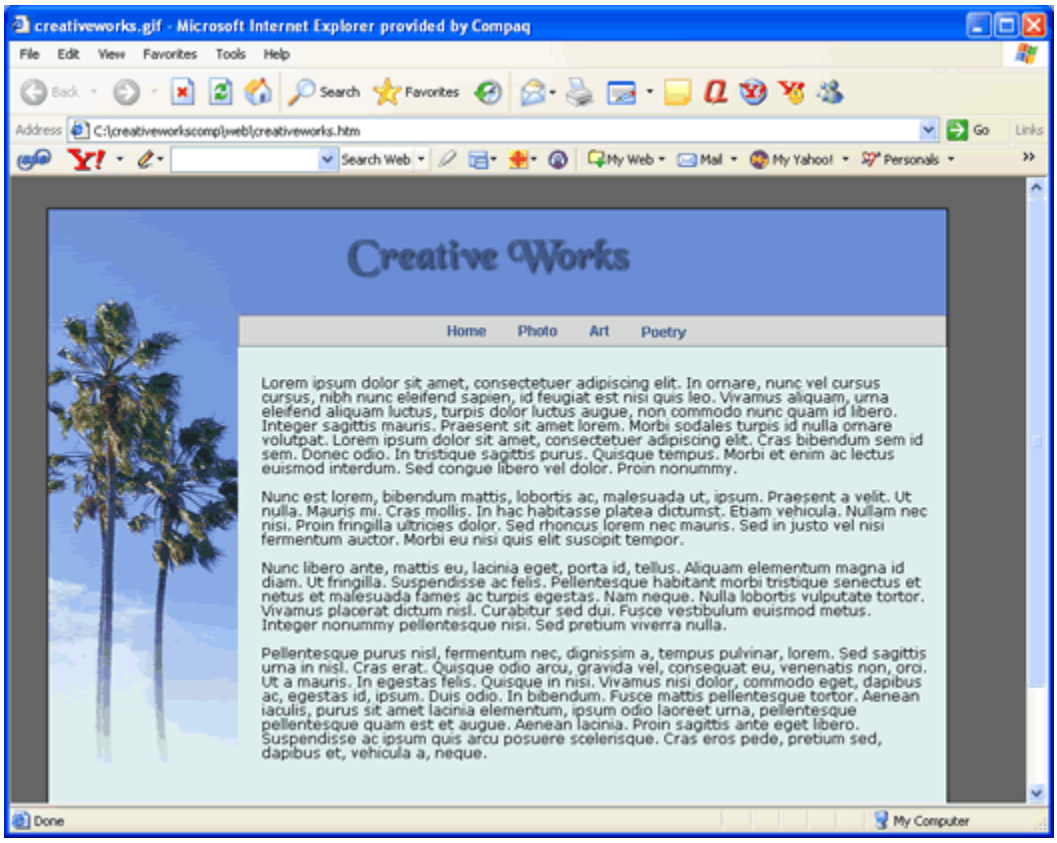


And the images for that web page has been generated inside the image folder ...



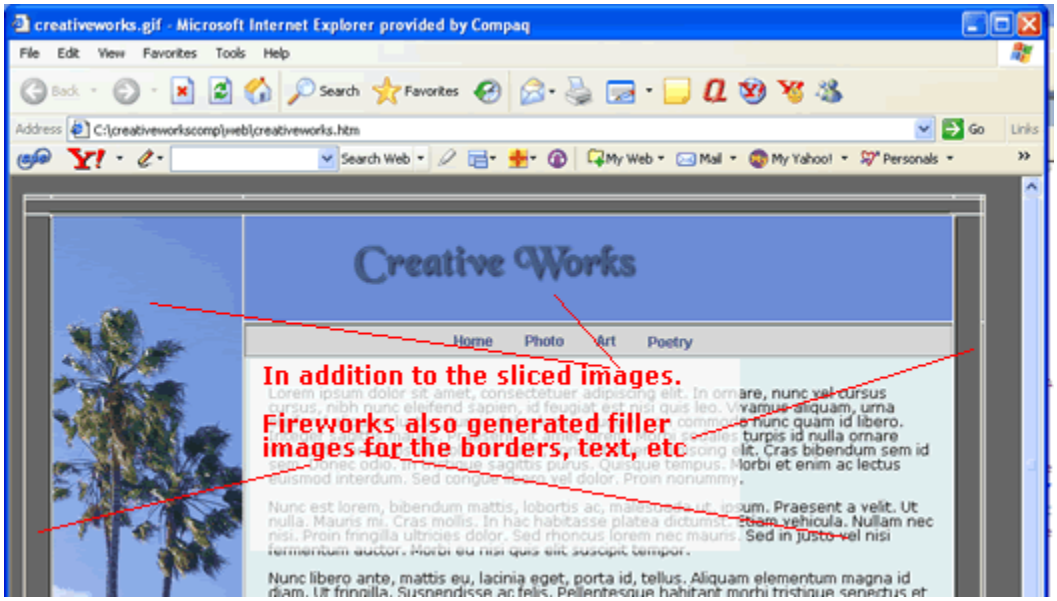
Note that our two sliced images have been given filenames matching their slice names of **sideimage** and **titleimage**. We'll talk about the filler images and the spacer.gif shortly.

But first, run our **createiveworks.htm** web page by double-clicking on it and loading it into our browser.



Well, it looks like a web page. However, it is a very poorly constructed web page. Fireworks does not know HTML very well. It only knows about creating HTML tables and putting images in to the cells of the table.

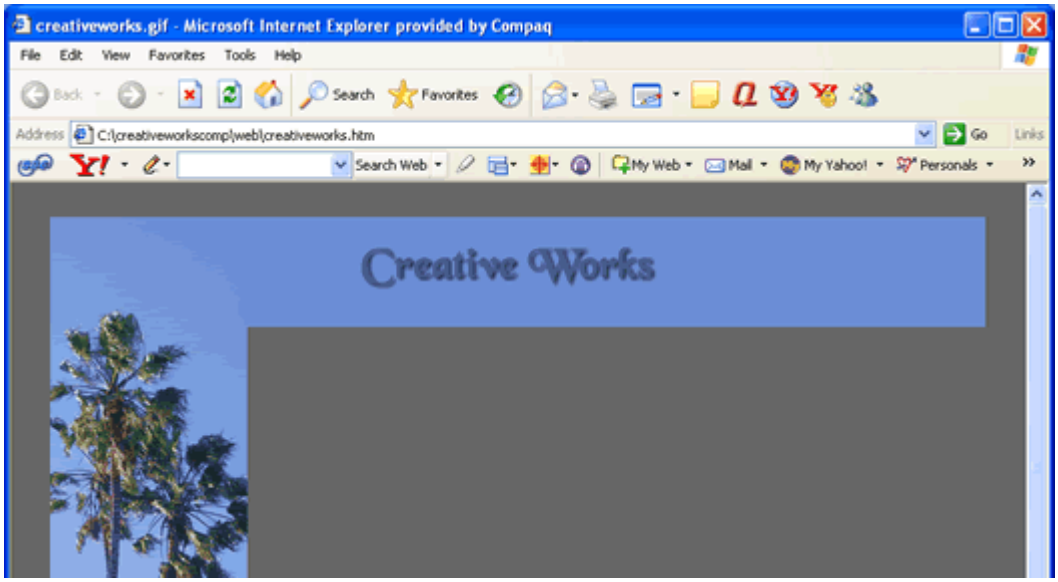
Here is the table structure that Fireworks has created.



So even through we wanted to implement the navigation bar, the content text, and the footer using HTML code, Fireworks does not know how to code them in HTML. So it outputted those elements as images instead. Those images are the filler images such as **creativeworks_r1_c1.gif**, **creativeworks_r2_c1.gif**, **creativeworks_r2_c4.gif**, **creativeworks_r3_c3.gif**, **creativeworks_r4_c2.gif**. For example, the filler image **creativeworks_r4_c2.gif** looks like ...

This is definitely not how web sites are built. Those filler images are virtually useless to us.

In the **Export** dialog above, we mind as well uncheck the "**Include Areas without Slices**" and get the following instead...



Even with this, Fireworks produces table structures that are far from optimal. In fact, it is considered poor. In the Intro HTML/Dreamweaver course, I'll teach you how to create a proper table structure for this webpage.

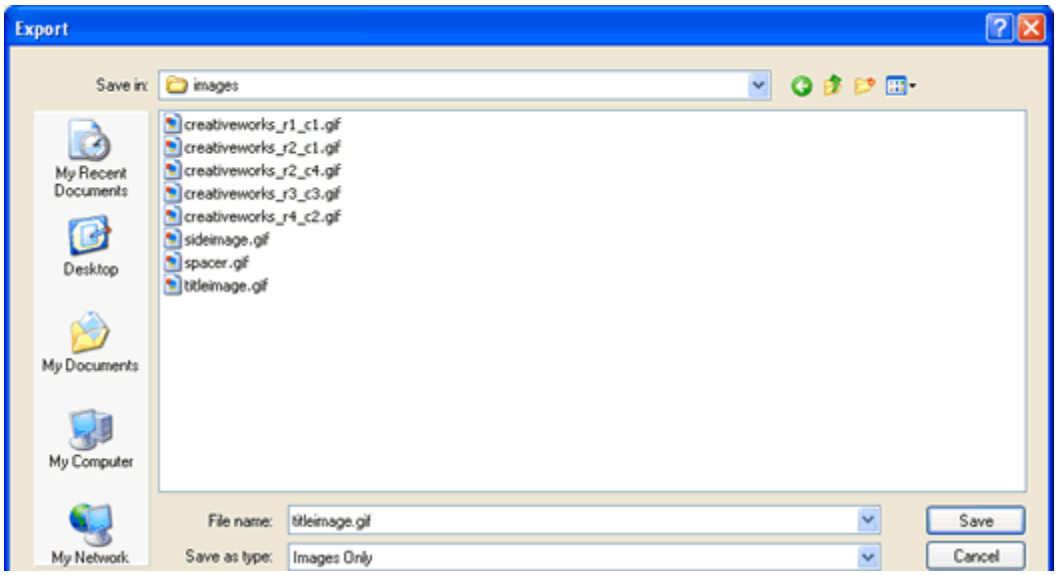
You can get my file for this lesson here: **creativeworks.png** (Click link to open in browser. Then right-click to do "**Save Picture As**").

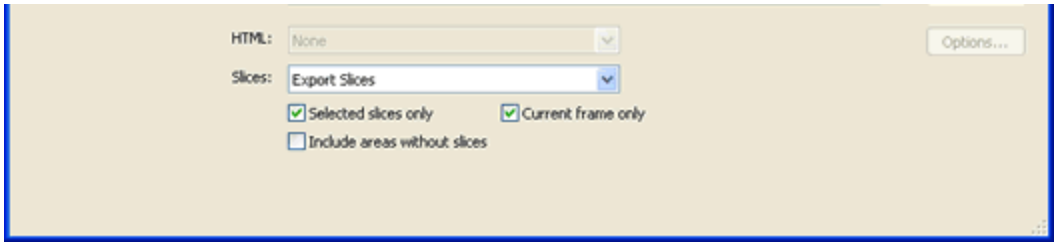
Exporting Slices Only

Now we will learn various export options, which depending on your purposes, you will need to use from time to time.

Typically, we want to export each slice of our layout individually. To do this, right-click on the **titleimage** slice and **Export Selected Slice**...

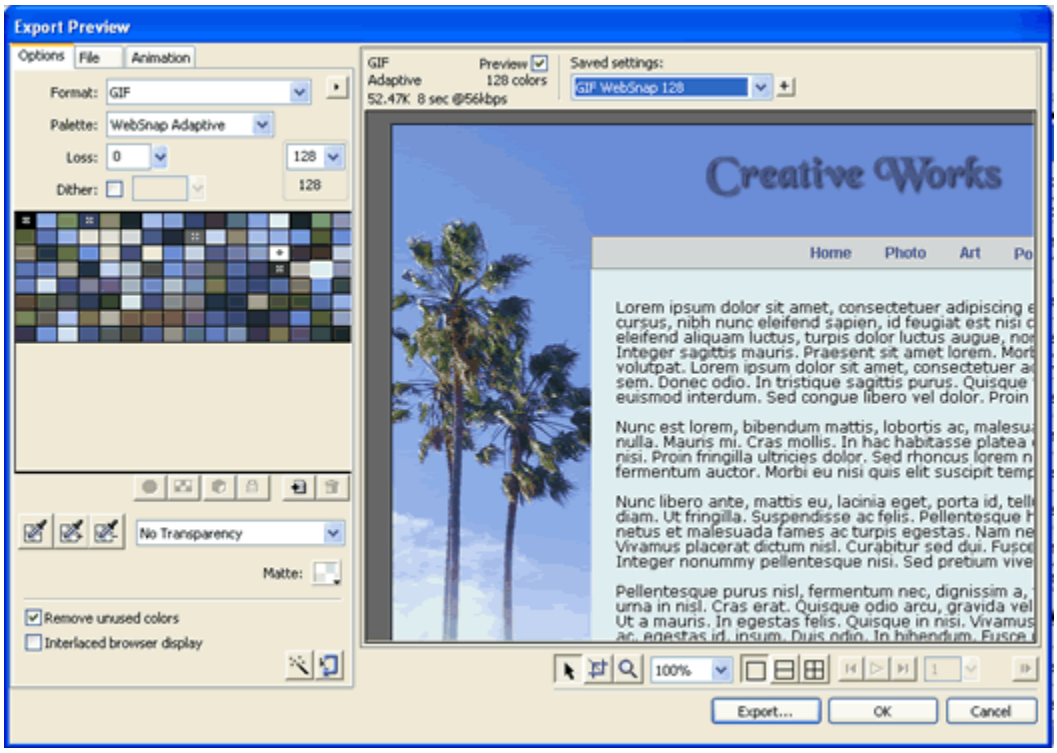
And save the file according to this dialog...





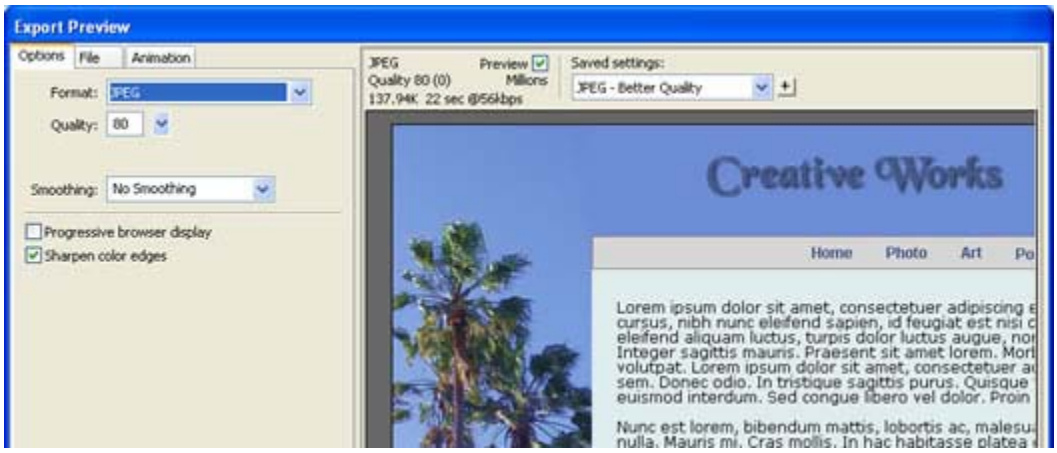
The filename that Fireworks provides to you in this dialog will be the name of your slice. And the format that will be outputted will be whatever is set in the **Export Preview** dialog.

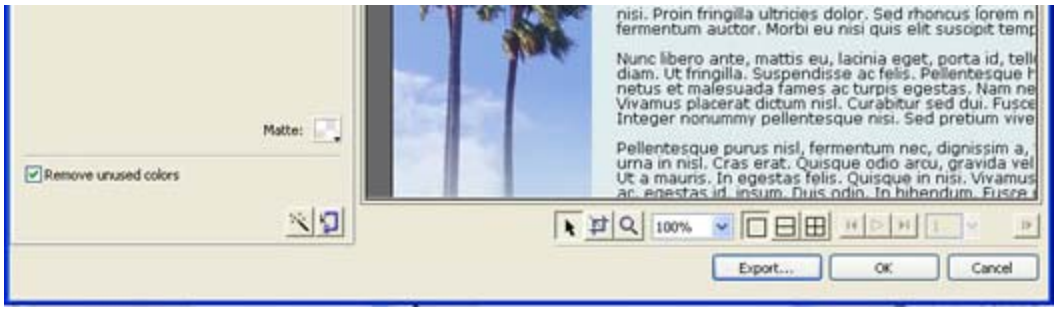
Let's go to menu **File -> Export Preview** now.



As you can see, my dialog had been previously set to GIF format and that is why the saved slice came out as **titleimage.gif**.

For the **sideimage** slice, we are going to export it as a JPG since it is a photograph. In the **Export Preview** dialog, set it to **JPEG** with quality **80** as shown.





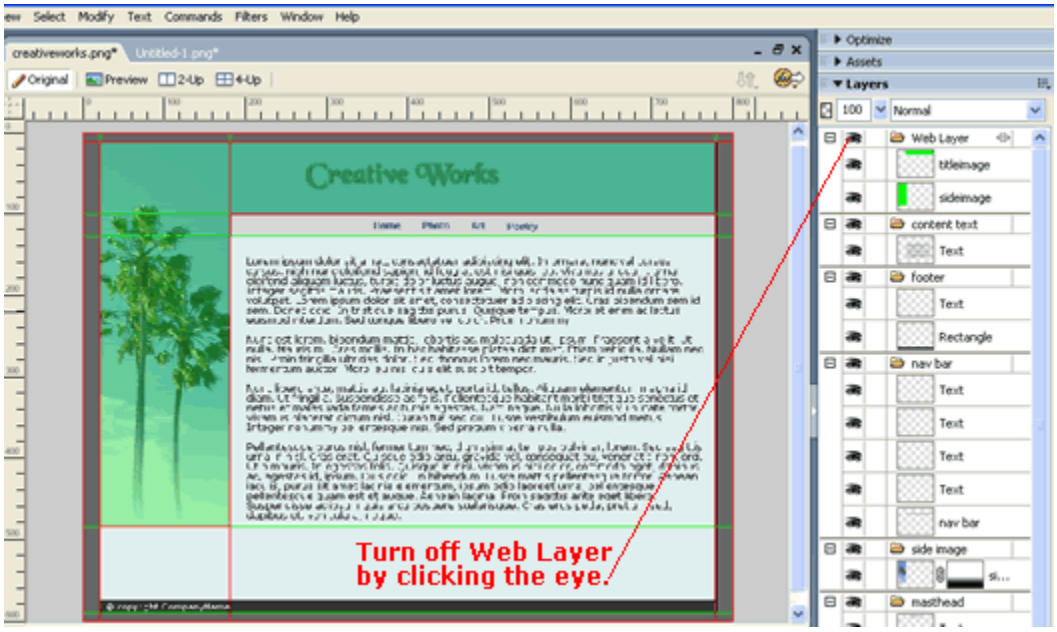
You may also have noticed that by setting to JPG, the image quality (especially just above and to the right of the tree) has improved. Compare the screen preview of this export dialog with the GIF export dialog from previous. But the price we pay is the larger file size. According to the dialog the JPEG will be 137K in size, which translates to about 22 sec for a 56kps modem to download it. In comparison, the GIF image is 52K.

Click **OK** to the export dialog. Clicking **OK** does not perform the export. It only sets the format. So export the sideimage slice by right-clicking on it and selected **Export Slice**. This time, Fireworks should save it as **sideimage.jpg**.

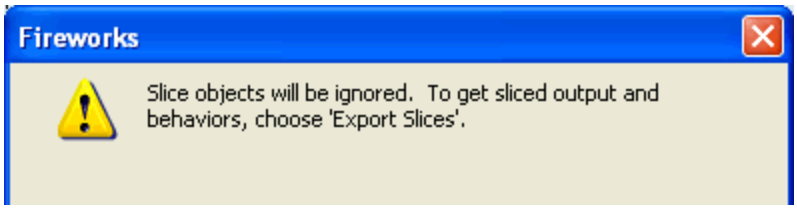
Hold on to the files **titleimage.gif** and **sideimage.jpg**. These will be the two files that will be needed later when we do the HTML coding in Dreamweaver.

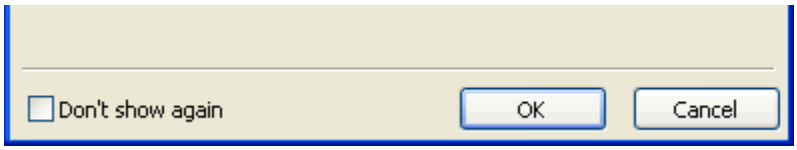
Saving Entire Comp Image

Sometimes you just want an single large image of your entire design to show someone for review, for example. Since you want to export the whole document, you will not be needing to work with the slice pieces. So turn off the Web Layer by clicking on the eye icon as shown...



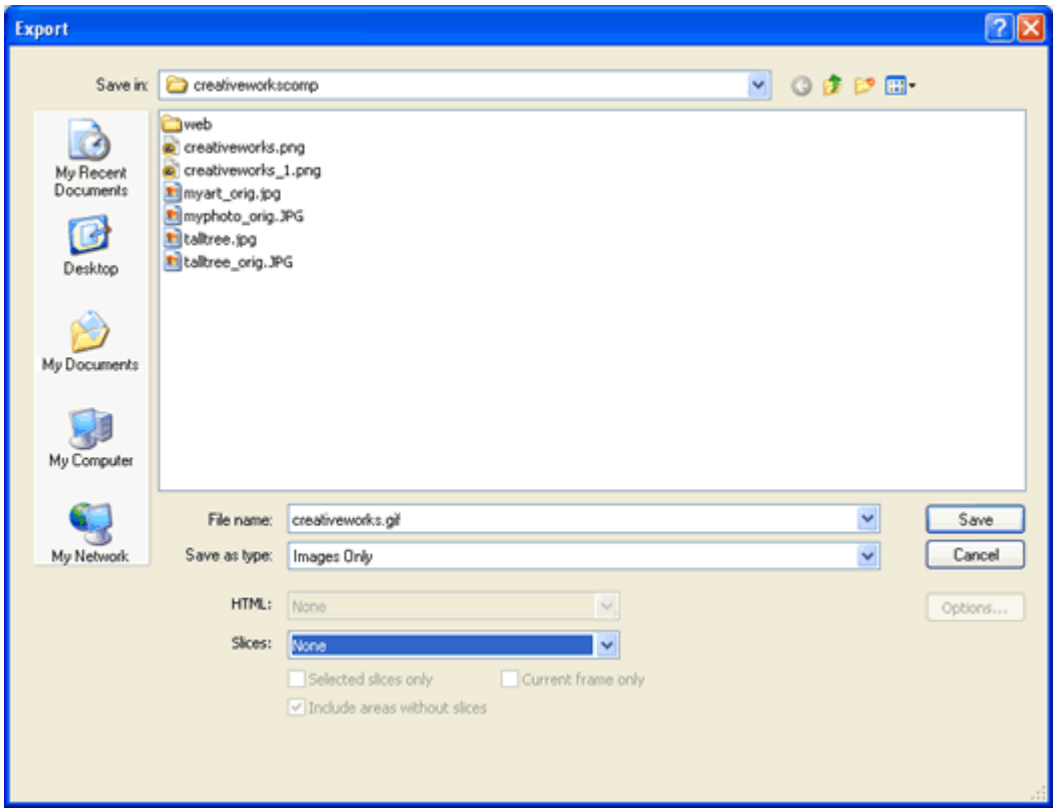
To export the whole document, do **Export Preview** and set it to either GIF or JPG. I typically will do GIF. And then do **File -> Export**. You might get this dialog ...



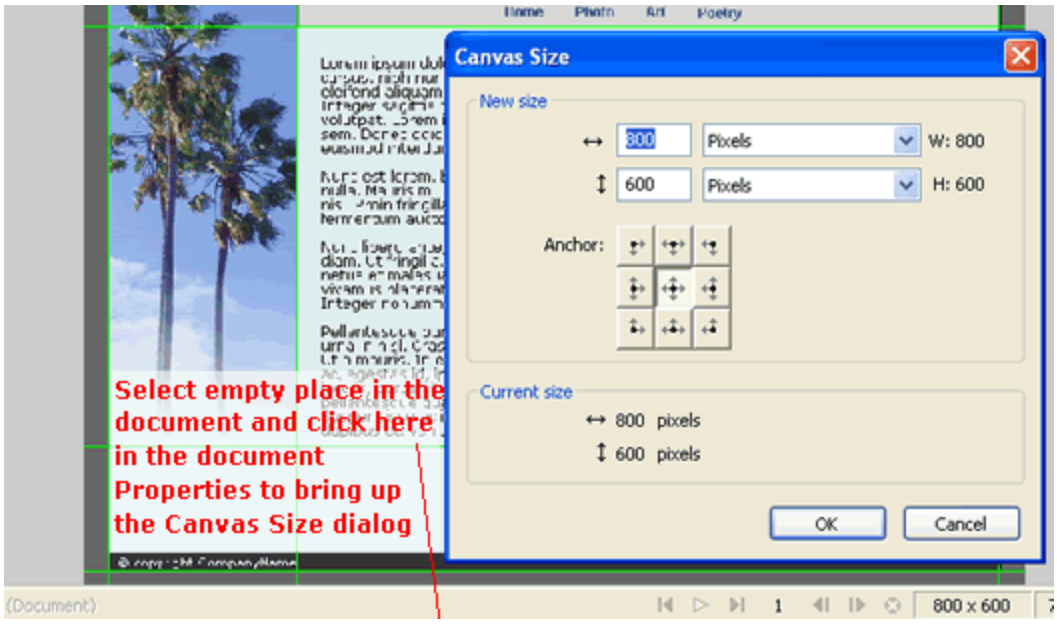


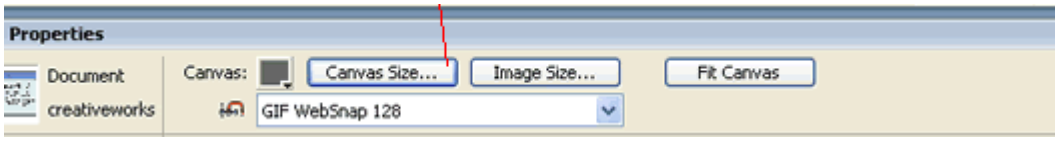
Just click **OK** to it.

Set your **Export** dialog to ...



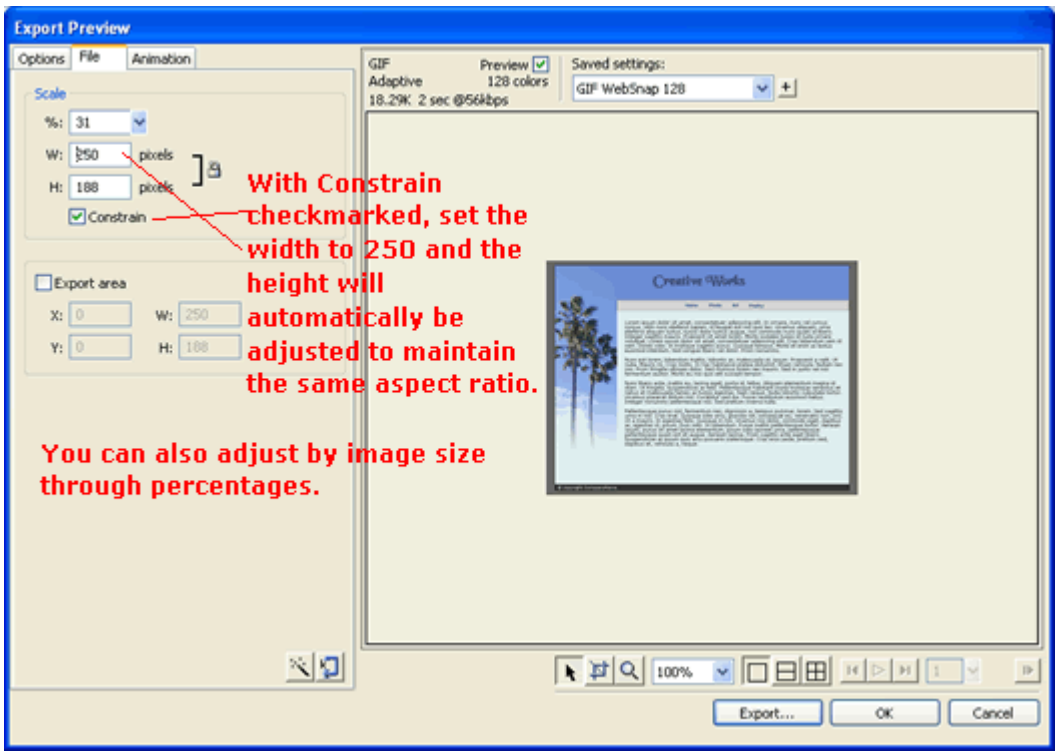
When you click **OK**, you will get an image of your whole design that is 800x600 pixels -- since this is the document size shown in the **Canvas Size** dialog.





Creating A Thumbnail Image

Suppose you don't want such a large image. You might just want a small image for a portfolio showcase that is about 250 pixels wide. You can export a smaller image through the **File** tab of the **Export Preview** dialog...



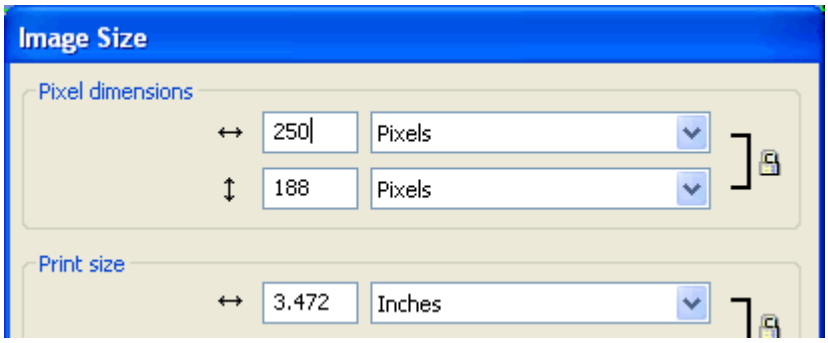
From here, you can export immediately with the **Export** button. Or you can click **OK** to save the setting and export later with **File -> Export**.

Note that this only sets the size of the file that is outputted. As you can see from the **Canvas Size** dialog, the document is still 800x600.

Let's make a backup copy now with **File -> Save A Copy** and give the copy some name such as **creativeworks_2.png**.

Changing the Canvas Size

If you really did want to change the acutal image size, you can do so through **Modify -> Canvas -> Image Size** with the settings shown. But don't make the changes, because we want to keep our comp at full 800x600 size.



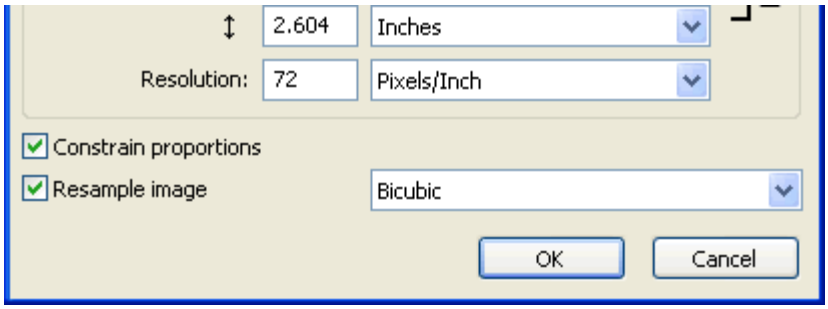
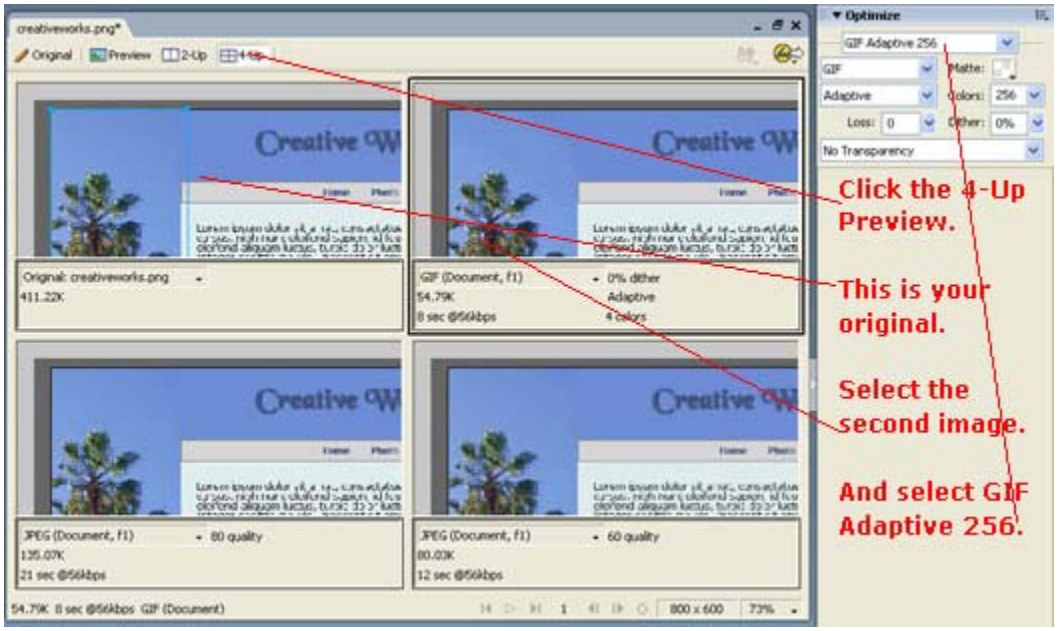
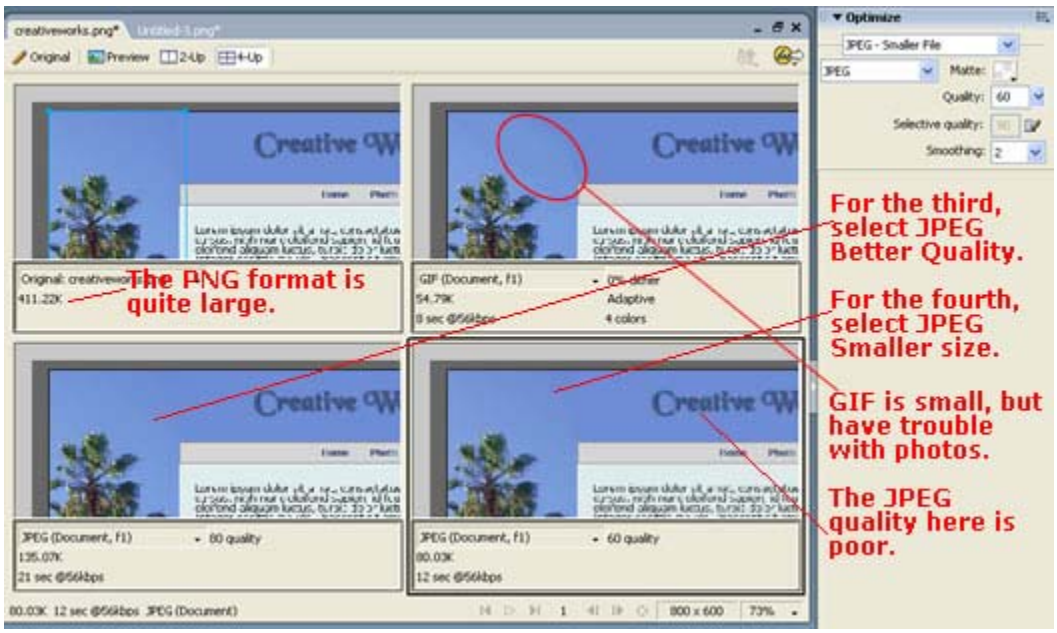


Image Optimization

When exporting an image, you want to find the best combination of small file size and good quality. The **Preview View** in conjuntion with the **Optimize** panel will help you do this.





Now that you can see all four settings side-by-side, it will be easier for you to pick out the one with the right balance of file

size and quality.

Since most modern browser can render the PNG format, you can use Fireworks' native PNG format in your HTML web page. However, it is often not done because the file size is quite large -- the upper left preview shows it to be 411K.

GIF format can support transparency as well as animation, and its file size is quite small. So GIF is most common and is used by default. The upper-left preview shows it to be 54K. However, GIF uses at most 256 colors and therefore may have difficulty with gradients and photographic details. For example, you may notice some pixelation of the sky area marked in the preview.

Since JPEG can support over a million shades of color, it is good if you require photographic quality. And there is a quality setting that you can adjust. The lower-left preivew has a high quality setting that results in a 135K filesize, whereas the lower-right preview is set at a lower quality that results in a filesize of 80K. With the low quality setting, the title text does not appear as sharp.

Note that in this example exercise, we are exporting the whole design comp as one large 800x600 image. That is why the filesizes are so large. This is just an example of the use of the preview views. This is not what we would do in building the actual webpage.

In building the actual webpage, we would be exporting the side image as JPEG and would be exporting the title image as GIF. And the rest of the site is pure HTML. If you look at the filesizes of **sideimage.jpg** and **titleimage.gif** that you had exported previously, they are around 17KB and 4KB respective. These are the right size and quality needed for a web page.

Completed Fireworks Portion

That's it for the Fireworks portion of the lessons. You are ready to move on to the HTML and Dreameaver lessons starting on the next page. You will be using these two images in building the website: **sideimage.jpg** and **titleimage.gif** (Download by clicking the link to bring them up in your browser. Then right-click on the images and do "**Save Picture As**".)

You can download the final **creativeworks.png** file in the same fashion.

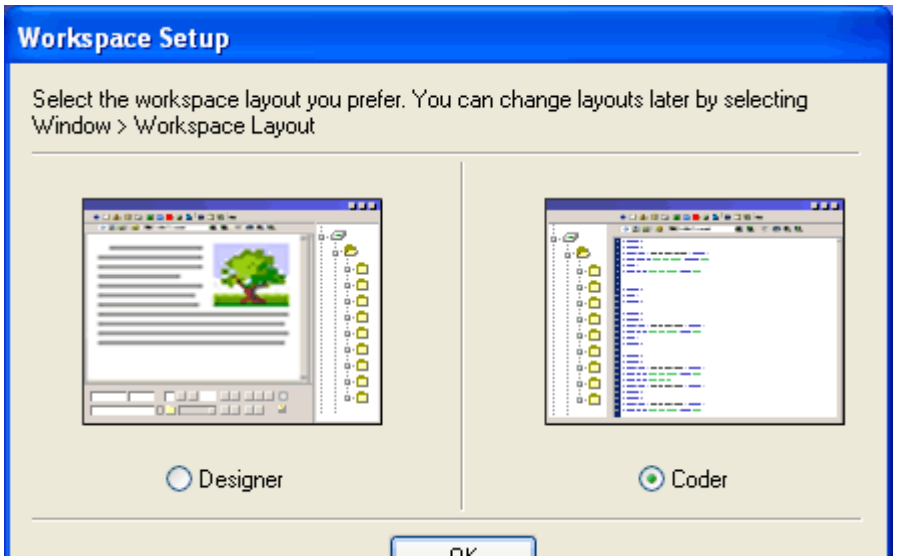
PART II: Introductory HTML and Dreamweaver Lessons

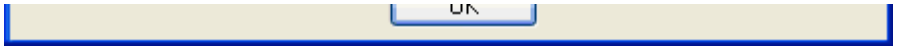
In these lessons, we will learn the very basic of HTML, the language for creating web pages and web sites. No experience needed. Just follow the step-by-step instructions as we use Dreamweaver 8 to implement the website based on the Fireworks design comp created in the first part of the book. You can get the **creativeworks.png** comp file by clicking on the link to open it in a browser. Then right-click and do "**Save Picture As**".

Starting Dreamweaver

In these lessons, we will create the **Creative Works** website from the design comp that we had created in the Fireworks lesson. But before we do, let's learn the fundamentals of Dreamweaver and HTML first.

When you start up Dreamweaver for the first time after installing it, it might as you ...

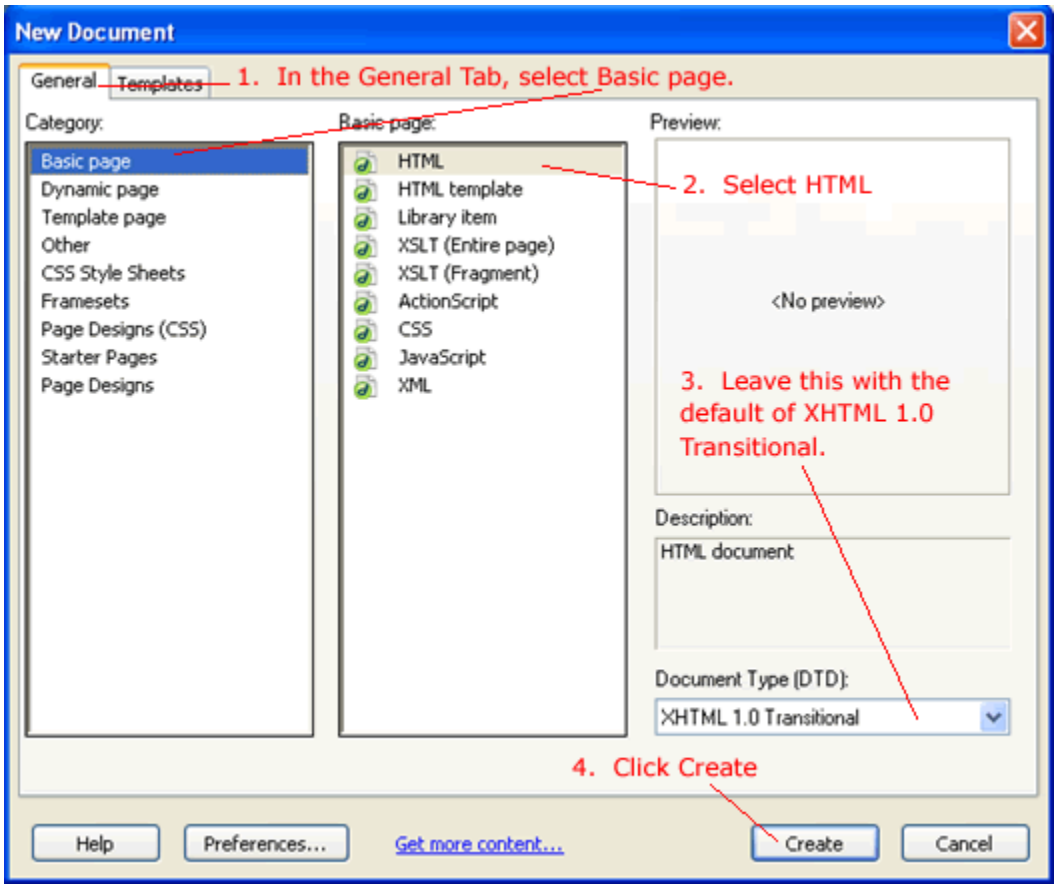




Since we will be learning HTML code and will be doing hand-coding as well as the design features of Dreamweaver, select **Coder**. In actuality, you can choose either and you will still have all the same functionality. Except that the panels will be laid out differently. That's all. If you had chosen **Designer** previously, you can switch to **Coder** by menu **Window -> Workspace Lyout -> Coder** so that you Dreamweaver will look somewhat like the screen shots in this course.

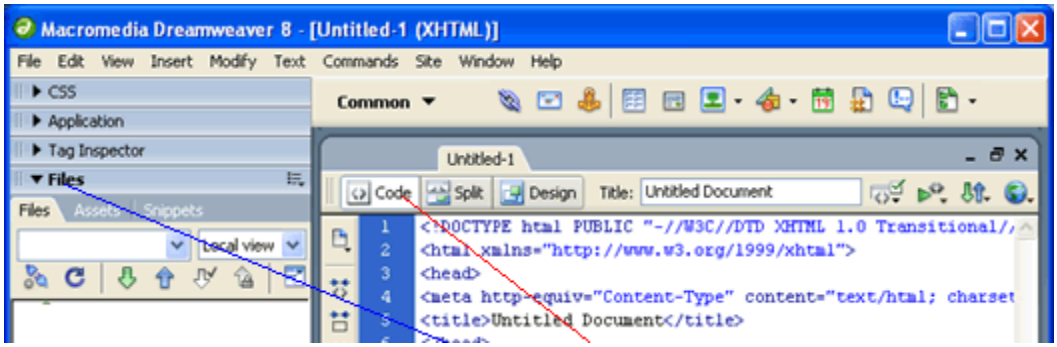
New Document

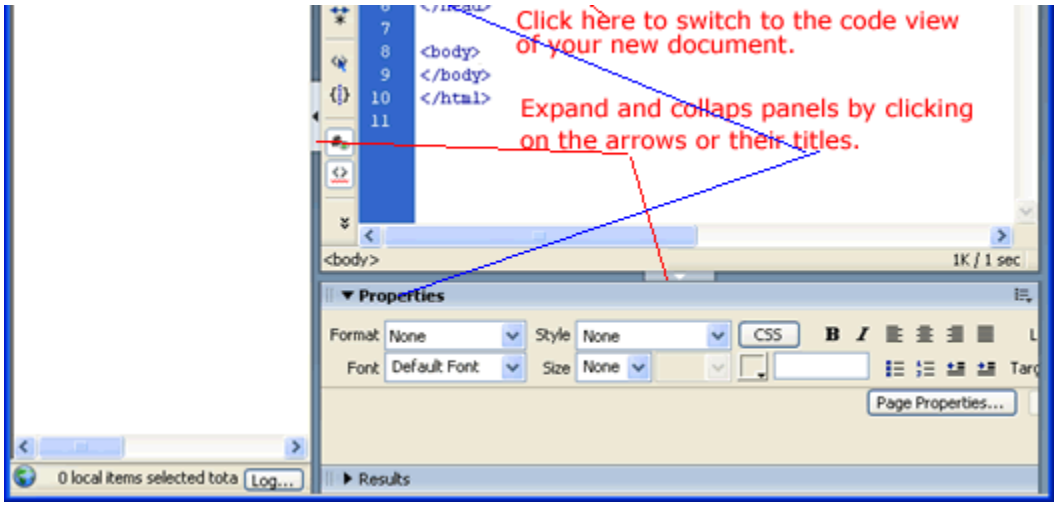
Start by creating a new HTML document in Dreamweaver with **File -> New**...



XHTML is the new standard. Think of XHTML 1.0 as the version of HTML markup that is after HTML 4.0. We won't be looking at the **XHTML spec** now (because the reading is too dry and technical). But just know that this link in W3C's website (W3 Consortium's website) is the ultimate reference point since they are the ones who defined XHMTL.

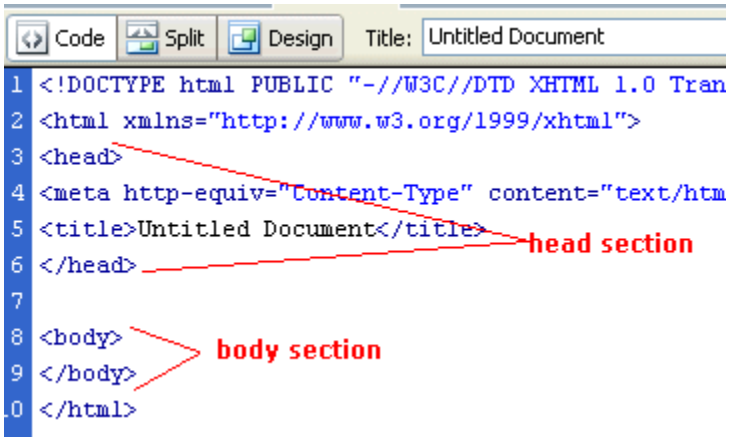
After clicking **Create** in the above dialog, you will see that the development environment in Dreamweaver is very similar to that of Fireworks in that you have collapsable and tear-able panels.





In Dreamweaver, there is an extra document bar with three buttons "**code**", "**split**", and "**design**" to enable you to switch between these three views. If you don't see this bar, select menu **View -> Toolbar -> Document**.

Go to the code view now by pressing the **code** button now. Here you see the default HTML skeleton code that Dreamweaver has generated in your new document.

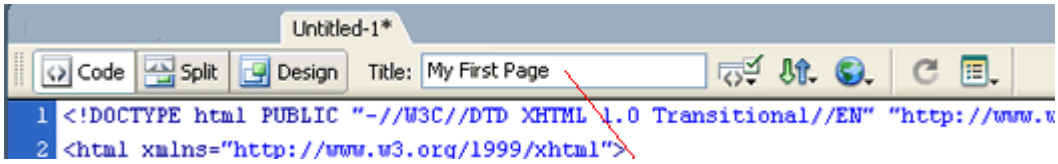


These are basically the code that you would write every time you create a new XHTML document. Dreamweaver is nice enough to write them for you. The things in angle brackets are ***tags***. The very first line that has **doctype** in it. This basically indicates to the browser what version of HTML code we are using.

The **<html>** tag indicates the start of the HTML document. And the **</html>** tag indicates the end of the document. Although this is really an XHMTL document, we are going to use the term HTML loosely. Note that tags often comes in pairs: **<html>** is the start tag, and **</html>** is the end tag.

You can nest additional tags inside tags. For example, inside the body of the **<html>** tag are the **<head>** tags and the **<body>** tags as indicated. You can say that an HTML document has two sections: a head section and a body section as shown in the above diagram.

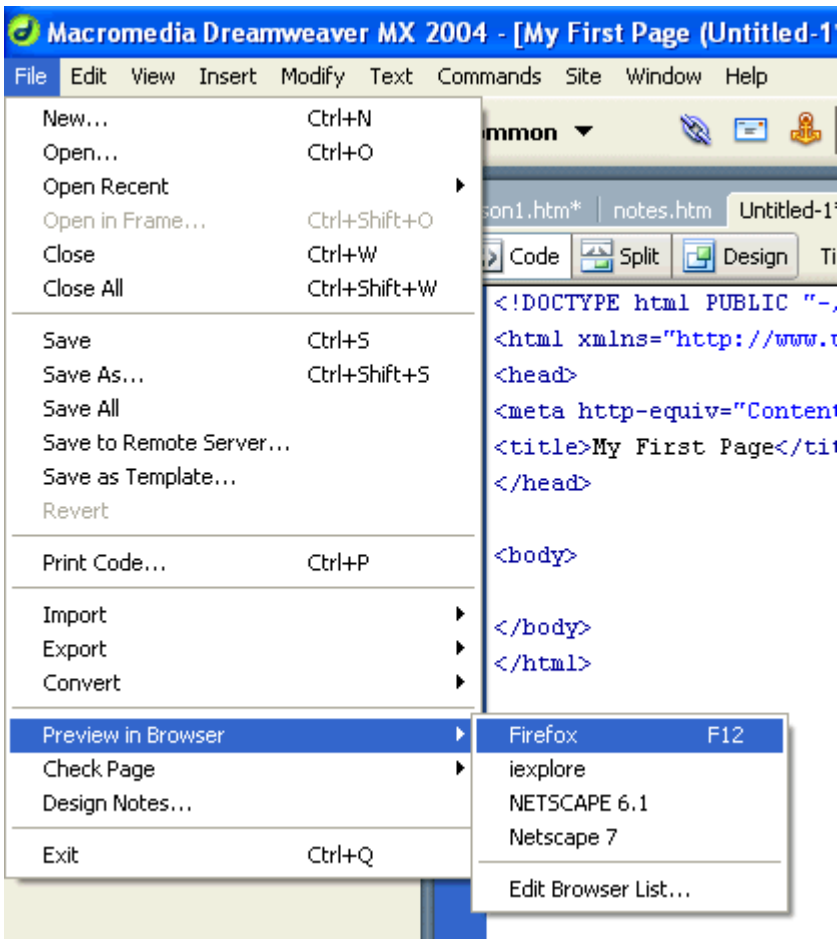
The head section contains things like the title of you web page for example. Give our page a title by adding a title of **My First Page** by doing as shown.



```
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5  <title>My First Page</title>
6  </head>
7
8  <body>
9
10 </body>
11 </html>
12
```

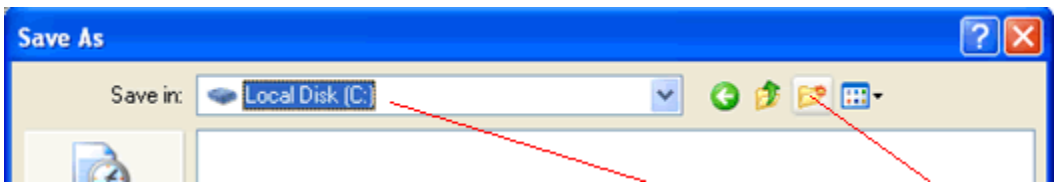
Type our title here and press Enter.
This write out the title in the code here.

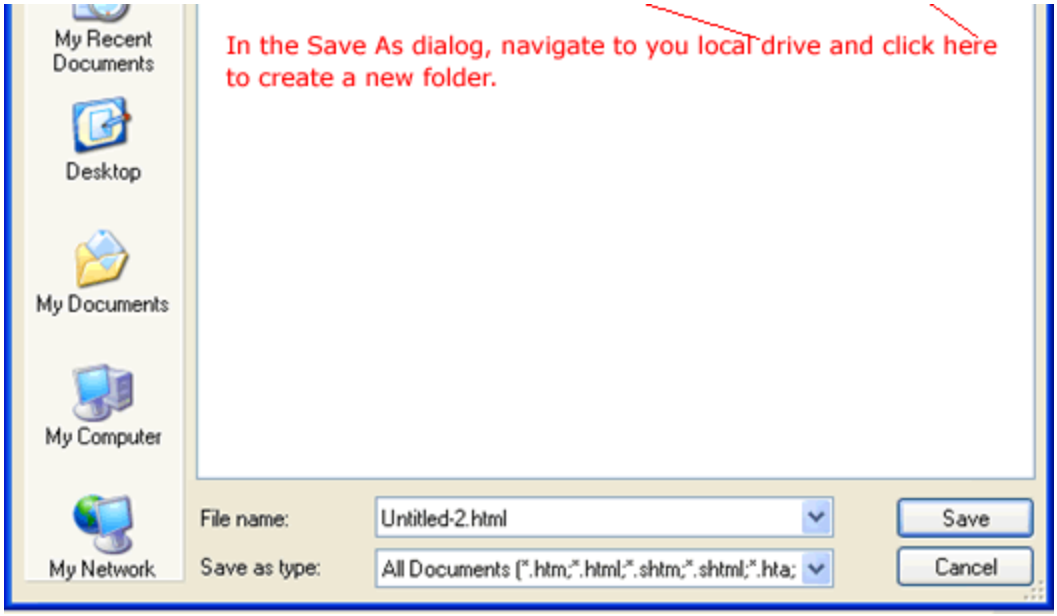
We will now test our page in our browser to see what it looks like. Dreamweaver can launch your browser for you and load the page if you do menu **Menu -> Preview In Browser**.

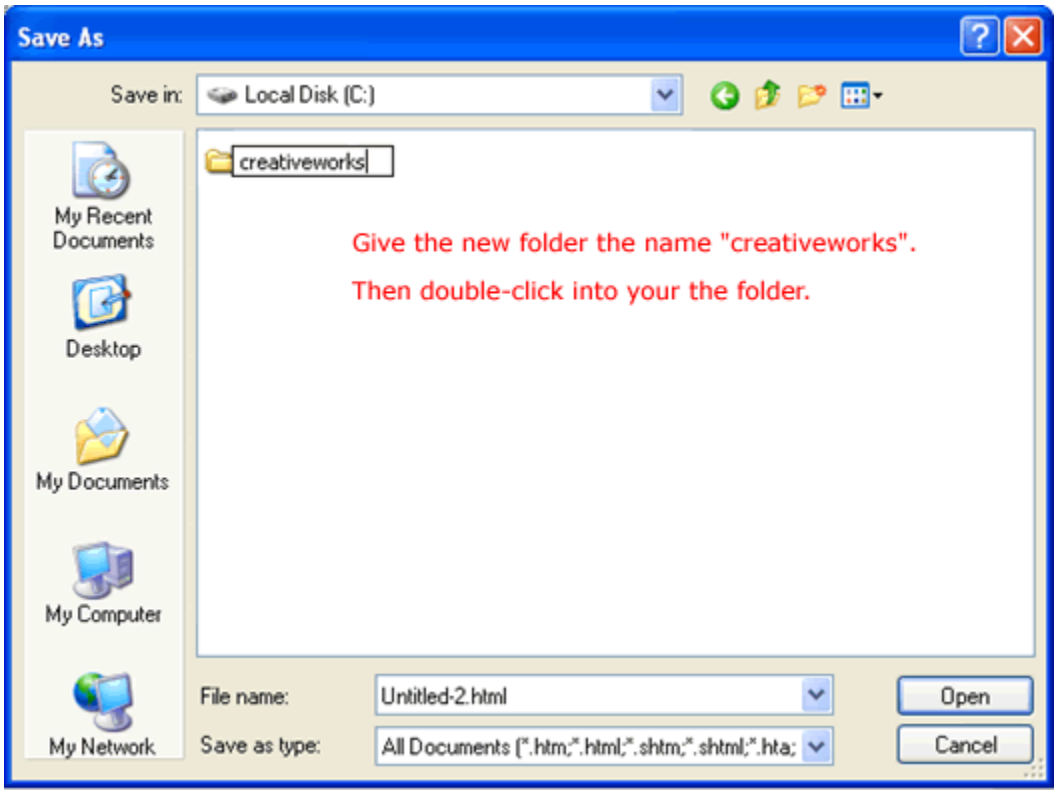


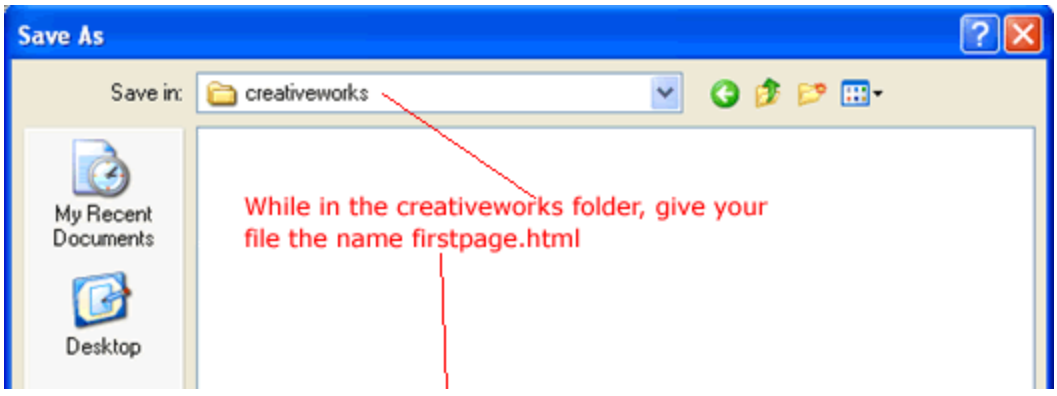
If your menu does not have any browsers listed, you can select **Edit Browser List** in menu above to setup Dreamweaver to become aware of the browsers that you have on your computer.

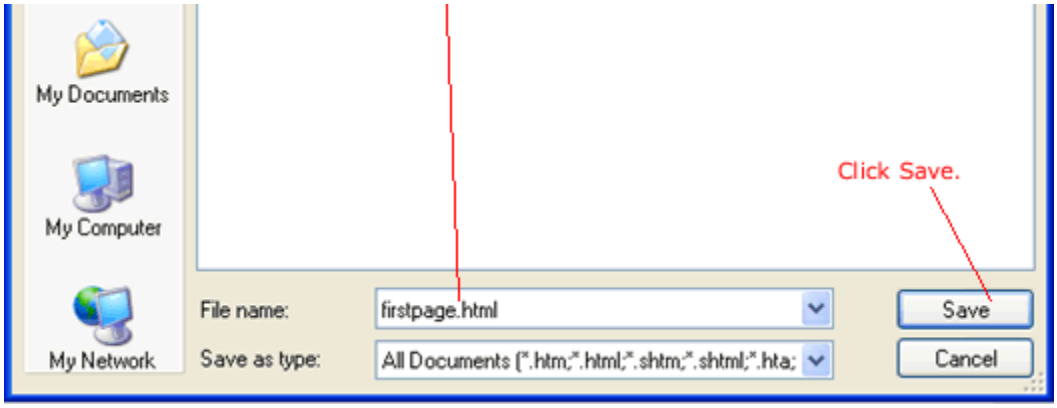
Dreamweaver needs to have a saved document in order to preview, so it might ask if you want to save the changes made to the current document. Say "**yes**" and save this file as **firstpage.html** in a new folder called **creativeworks** by following these steps...



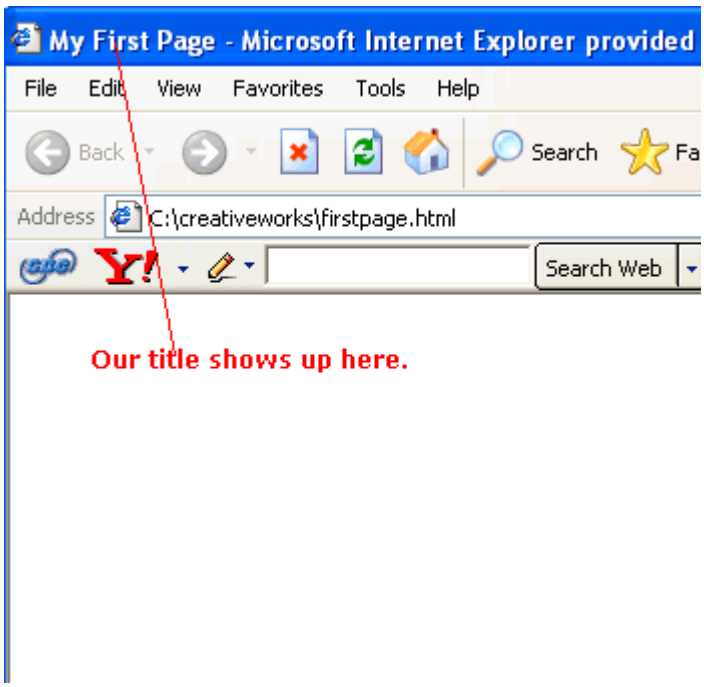

In the Save As dialog, navigate to you local drive and click here to create a new folder.
My Recent Documents
Desktop
My Documents
My Computer
My Network
File name:
Untitled-2.html
Save as type:
All Documents (*.htm;*.html;*.shtm;*.shtml;*.hta;
Save
Cancel


Save As
Save in:
Local Disk (C:)
creativeworks
Give the new folder the name "creativeworks".
Then double-click into your the folder.
My Recent Documents
Desktop
My Documents
My Computer
My Network
File name:
Untitled-2.html
Save as type:
All Documents (*.htm;*.html;*.shtm;*.shtml;*.hta;
Open
Cancel


Save As
Save in:
creativeworks
While in the creativeworks folder, give your file the name firstpage.html
My Recent Documents
Desktop

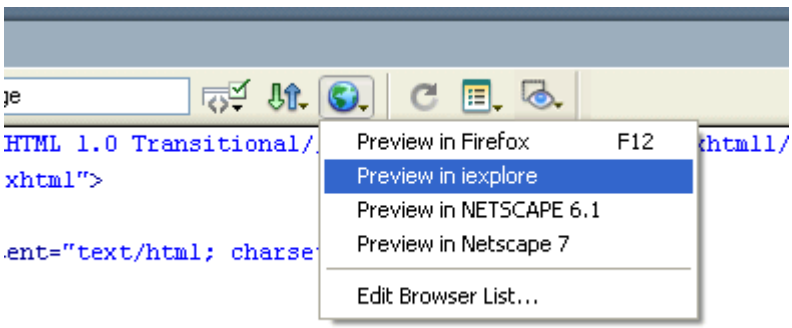


After Dreamweaver launches your page, you should see the following come up. Notice our title in the window header.



Right now, it doesn't matter what browser you use. However, in the ***Intro CSS*** course, you will be using both Internet Explorer and Firefox. So you might just want to get those two browsers to start with. If you do not have Firefox, it is a **free download**.

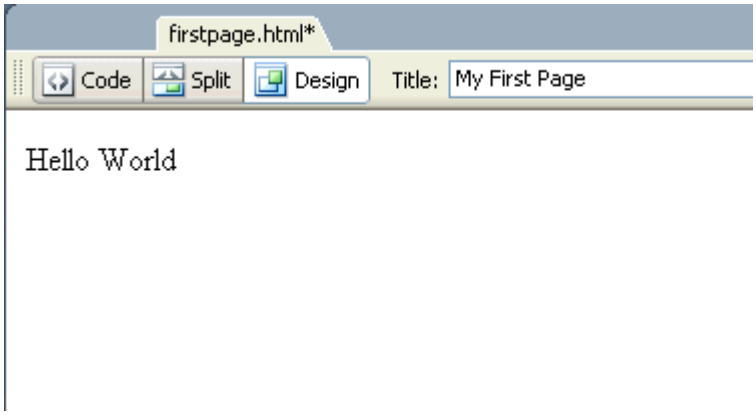
Dreamweaver has made this testing task easy for you by giving you a shortcut key of **F12** and a toolbar button as shown.



So there is no excuse for not testing frequently.

Adding Content

So far our page is empty. We want to add some content to our page. Click the **Design** button to go to the design view and start typing the following.

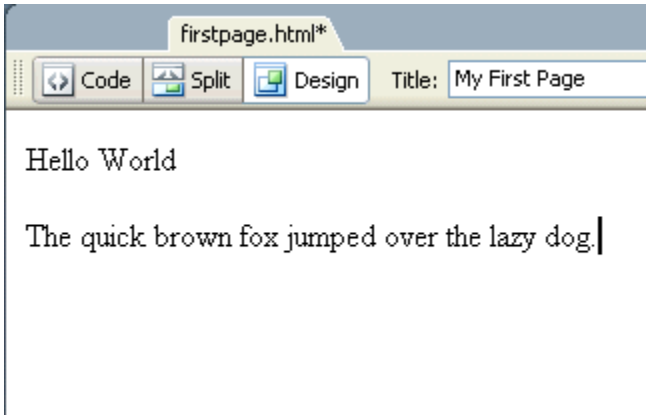


Switch back to the code view. See how your content is added within the **<body>** tag.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1
<html xmlns="http://www.w3.org/1999/xhtml"
<head>
<meta http-equiv="Content-Type" content="t
<title>My First Page</title>
</head>

<body>
Hello World
</body>
</html>
```

The **body** tag is what contains the content of our web page.

Go back to the design view to add another paragraph.



Switch back to code. See how the **<p>** tag is used to represent paragraph.

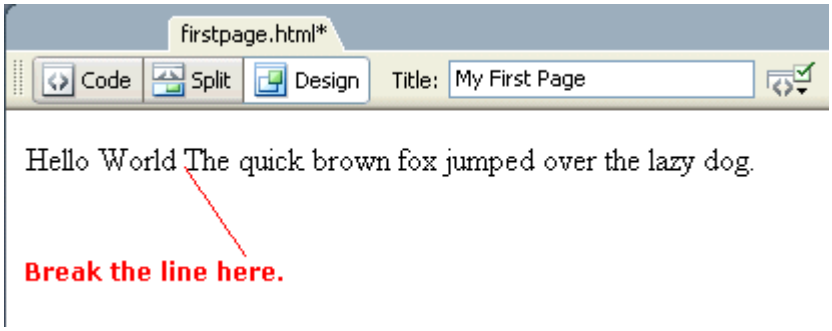
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; chars
<title>My First Page</title>
</head>

<body>
<p>Hello World </p>
<p>The quick brown fox jumped over the lazy dog. </p>
</body>
</html>
```

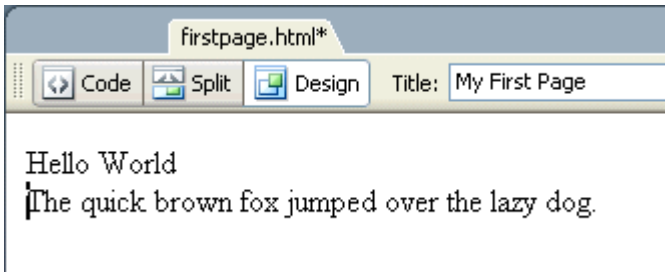
 tag indicates the start of paragraph. indicates the end.

There are two tags. One ** ** tag (known as the ***start tag***) that indicates the start of the paragraph and one ** **(known as the ***end tag***) that indicates the end of the paragraph. The ***end tag*** is the one with a slash right after the first angle bracket. The text "The quick brown fox jumped over the lazy dog." in between the two tag is known as the ***body*** of the tag, or the ***content*** of the tag.

What if you don't want two paragraphs, but two lines by themselves? Go back to the design view and remove the paragraphs so that you have only one line.



Now you want to ***break*** the line at the location shown. Just do **Ctrl-Enter** at that location. (**Ctrl-Enter** is equivalent to the menu **Insert -> HTML -> Special Characters -> Line Break**. This menu item is buried so deep that virtually eveyone uses **Ctrl-Enter** instead.)



And see that the line break tag **
** was generated...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitior
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; cha
<title>My First Page</title>
</head>

<body>
<p>Hello World <br />
```

line break tag consisting of both a start and end tag combined.

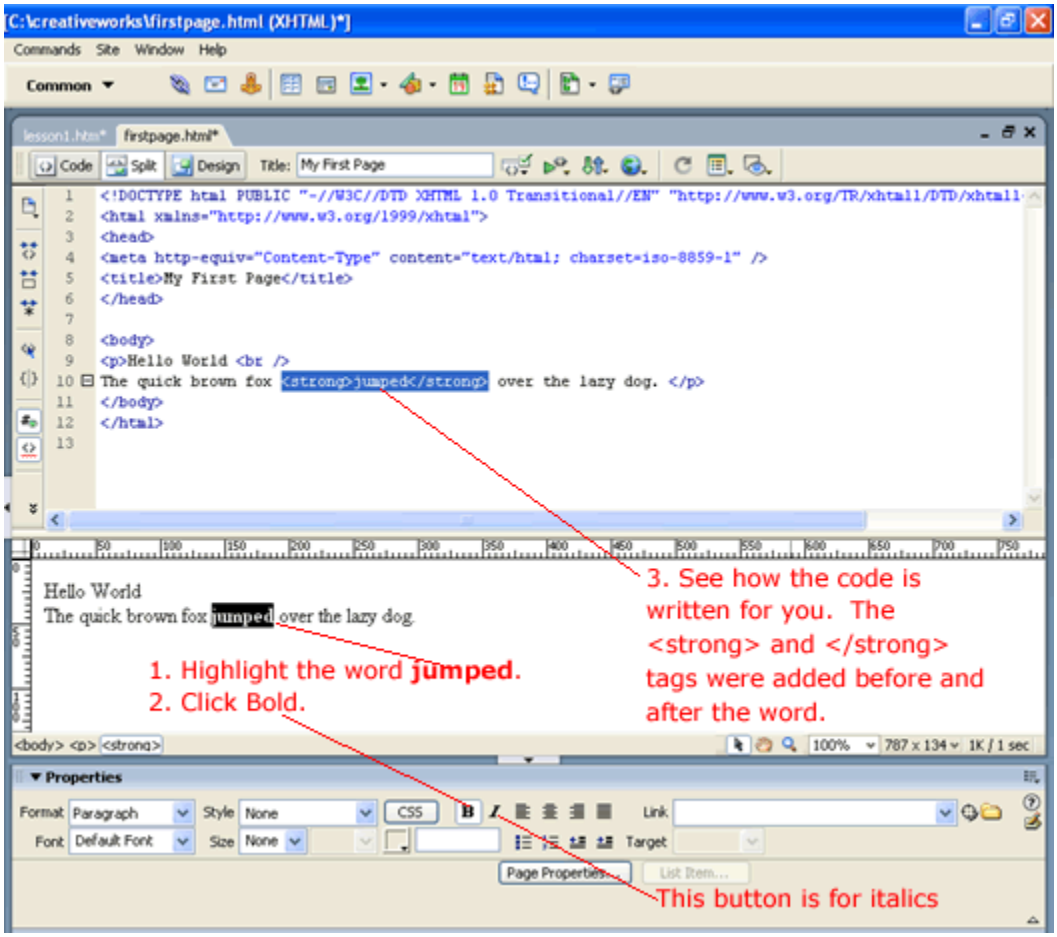
```
The quick brown fox jumped over the lazy dog. </p>
</body>
</html>
```

The line break tag looks a little bit different from the other tags. What is it? It has a slash **/** just before the second angle bracket. If you look around in the above code, there is not other tag that looks like this.

All tags comes in pair: a start tag plus an end tag. For example, the paragraph tags: ** ** and ** **. But where is the end tag for the line break tag? The line break tag does not have content in its body and therefore its start and end tag can be combined into this notation: **
**

Round-Trip HTML

Are you tired of switching back and forth betwween code and design view? Try the split view by clicking the **Split** button.



In the lower design view, bold the word "**jump**" as shown above and see how the change is immediately reflected in both views. This is Dreamweaver's famous ***round-trip HTML*** at work. You make changes in any of the views and it is automatically reflected in the other views. And as you format the code the way you like it, Dreamweaver tries its best to not alter your code format as it performs its update your code or generate new ones. In my opinion, Dreamweaver does this better than any other HTML editor that I have seen.

In addition to bolding the word "**jump**" also click the **italics** button right next to the bold. See that Dreamweaver writes out the following code...

jumped

Now you just learned the tags for bolding text and italics. They are **** and **** respectively. In the old days, people used **<b>** and **<i>**. But for semantics reasons, **** and **** are now preferred. Note how they properly nest one another. One of the XHTML requirements is proper nesting of tags. Hence it would be **incorrect** to do ...

jumped

Good time to save your document.

XHTML Syntax

XHTML is more strict syntax and has a greater requirement than HMTL. HTML lets you get away with sloppy code. Although the browser is able to render your page properly if you don't follow proper XHTML syntax, it is best to following them by habit so that you don't get into the habit of sloppy code.

At this point, I want to point out some of the proper syntax of XHTML so that you can recognize what are considered bad habits as you look at older code or read outdated tutorials on the internet.

Proper XHTML syntax requires the following...

1. Proper nesting of tags. You have just seen this example with the bold and italics.

2. All tags and attributes in lowercase letters and attribute. For example, write ** ** and not **<DIV ID="wrap">**.

3. All tag attributes must be in quoted. For example, write **<table width="80">** and not **<table width=80>**.

4. All tags must have matching closing tag. For example write ...

** Hello World **
** The quick brown fox. **

and not ...

Hello World
** The quick brown fox.**

(Even though the two code will look the same on the browser)

5. The **doctype** statement that we saw at the top is required.

6. All XHTML documents must have the following tags: **html**, **head**, **title**, **body**.

7. No shortcut notation in attributes. Write **<option selected="selected>** and not **<option selected>**.

Here are further references on XHTML syntax:

http://www.htmlgoodies.com/beyond/xml/article.php/3473511
http://www.w3schools.com/xhtml/xhtml_syntax.asp

Viewing Source

Here is my version of the code so far: **firstpage.html**

You can do one of several things with this link. The following

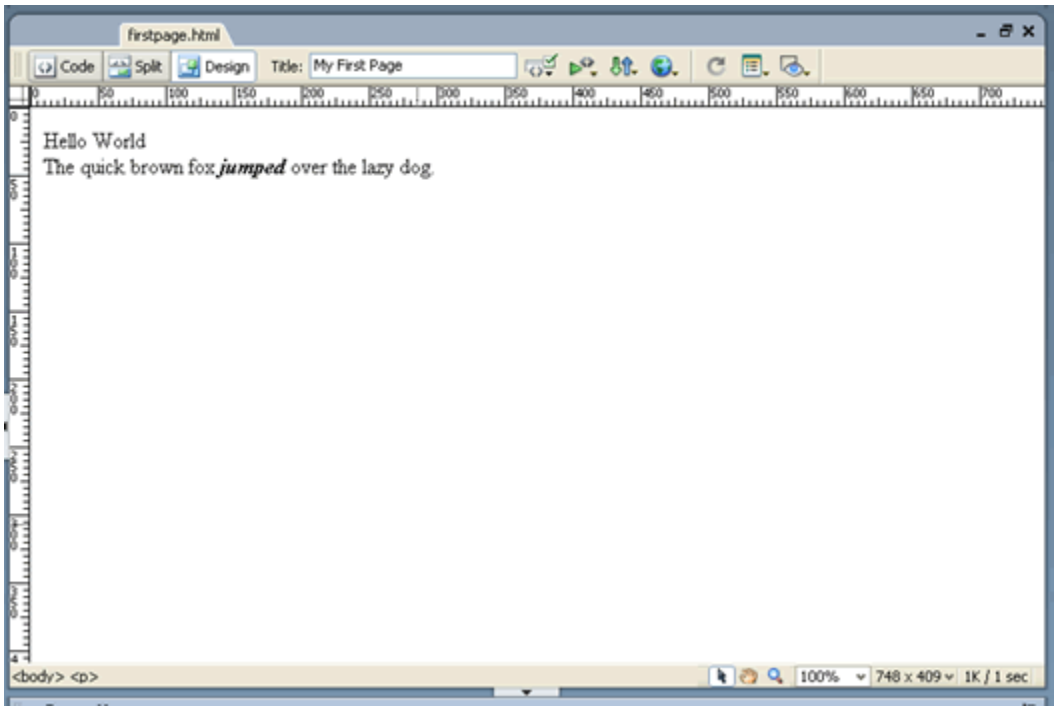
1. Click on the link to see how my page renders in the browser.

2. If using Internet Explorer (IE), right-click on the link and select **Save Target As** to download the file to your local disk. And then you can view it in Dreamweaver or any other text editor. In Firefox, the menu is **Save Link As**.

3. If using IE, after clicking on the link to view the page, you can do menu **View -> Source** to see the source code of my page. In Firefox, the menu is **View -> Page Source**.

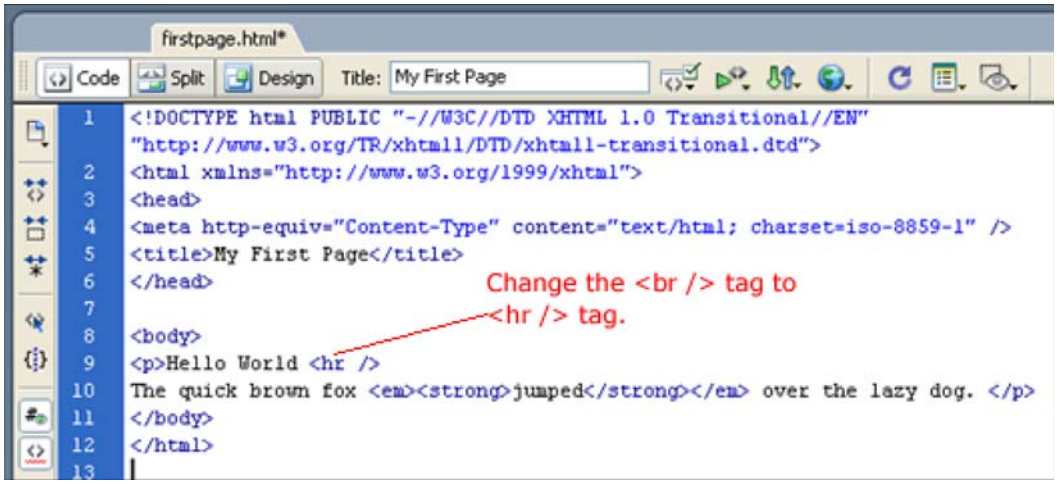
The Horizontal Rule

Do **File -> Open** in Dreamweaver to open up **firstpage.html** in the design view to refresh your memory as to what the page looks like...

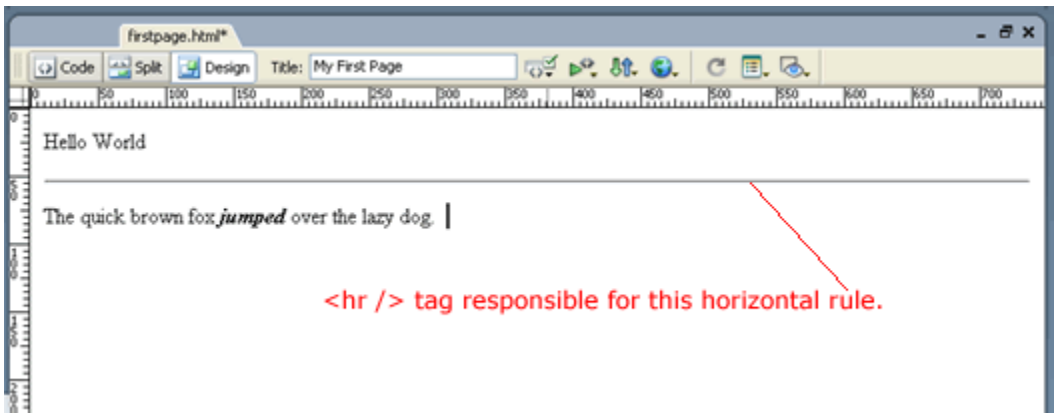


As a quick quiz, what was the tag that caused the word jumped to be bolded? To be italicized? What was the tag that caused the second sentence to be in a new line?

Switch to the code view now to see if you got the answers correct. It was indeed the **
** tag that caused the second sentence to be on a new line. Now change the **
** tag to the **<hr />** horizontal rule tag as shown below. And why does these two tags end in **/>**?

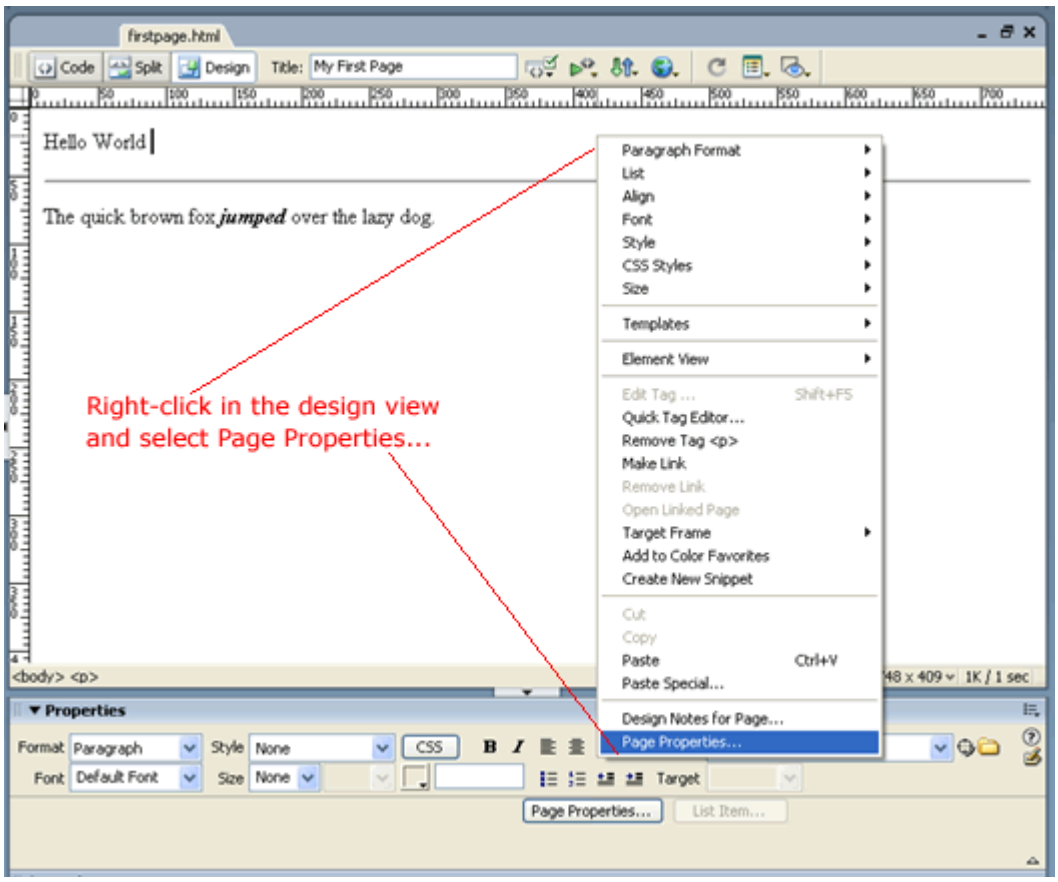


Switch to the design view to see what effect the **<hr />** tag had...

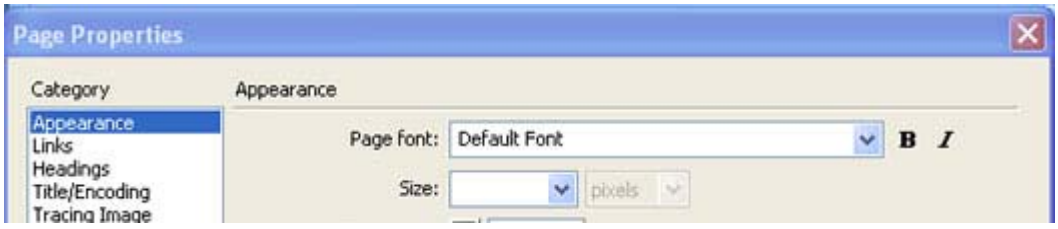


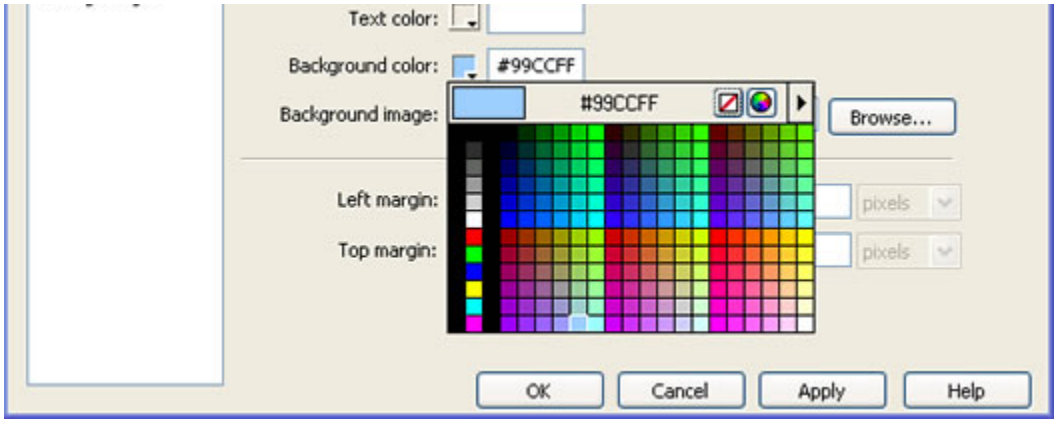
Changing the Background Color

Let's change the background color of the page by right-clicking and selecting **Page Properties**



In the **Page Properties** dialog, set the background color to #99CCFF by picking the square indicated and click **OK**.





Go to the code view and see what Dreamweaver has added.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>My First Page</title>
<style type="text/css">
<!--
body {
    background-color: #99CCFF;
}
-->
</style></head>

<body>
<p>Hello World <hr />
The quick brown fox <em><strong>jumped</strong></em> over the lazy dog. </p>
</body>
</html>
```

I don't like the way Dreamweaver has formatted this code.

There is no harm in reformatting it so that the **</head>** tag is on its own line. And that is what I do...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.o
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>My First Page</title>
<style type="text/css">
<!--
body {
    background-color: #99CCFF;
}
-->
</style>
</head>

<body>
<p>Hello World <hr />
The quick brown fox <em><strong>jumped</strong></em> over the lazy dog. </p>
```

I prefer it with this tag on a separate line.

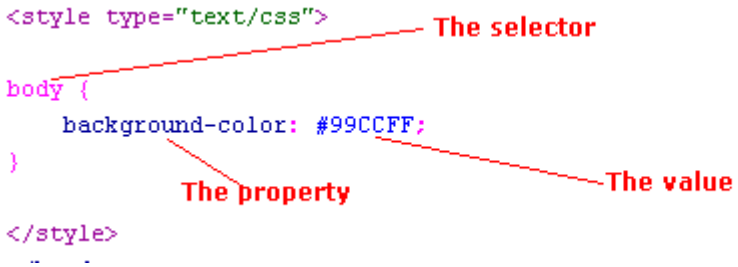
Internal style sheet with a CSS rule.

```
</body>
</html>
```

Dreamweaver has added an ***internal style sheet***, or ***internal styles*** indicated by the **<style>** and **</style>** tags.

(Don't worry about the gray **<!--** and **-->** just yet. They are HTML comment tags to hide these code from older browsers who don't understand style sheets. In fact, I am going to remove those, because they are irrelevant since all modern browser will understand style sheets now-a-days.)

Inside this style sheet is an ***CSS*** (Cascading Style Sheet) rule. A CSS rule consists of the **selector**, the **property**, and the **value**.



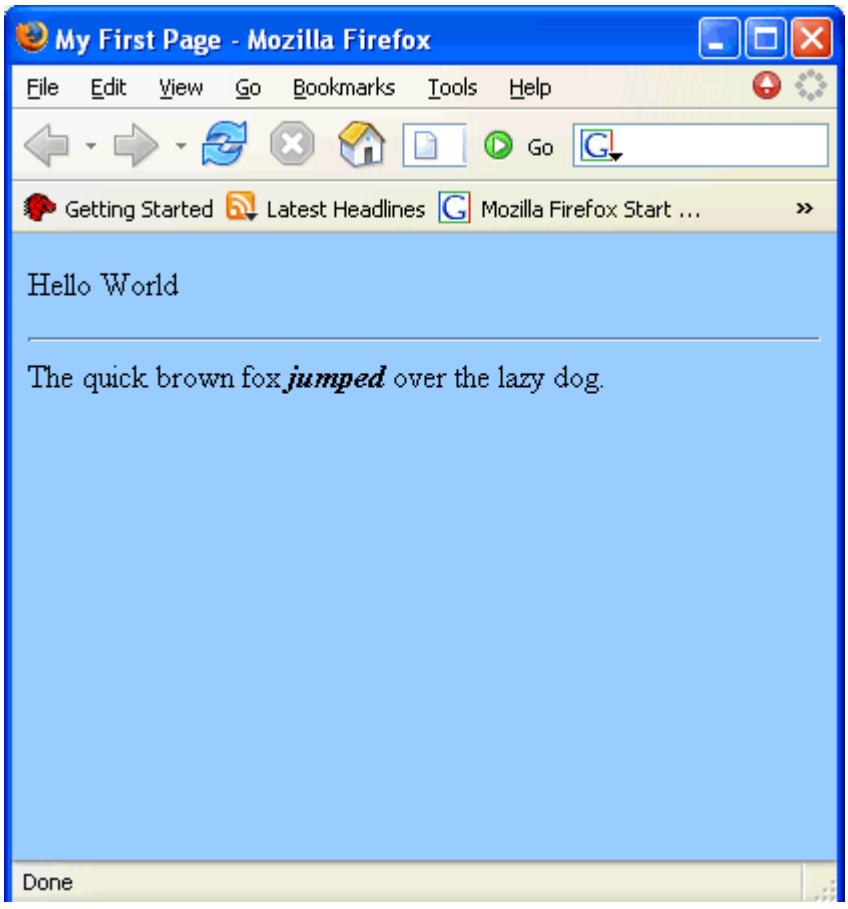

```
<style type="text/css">

body {
    background-color: #99CCFF;
}

</style>
```

This rule tells the browser ...

"for everything that is inside the <body> tag, please set their background color to the hexidecimal color of #99CCFF"

Because this rule is inside an internal style sheet, this rule only applies to this one page, and not to every page of our site.

And when the browser carries out this order, you get...



This color is a bit bright and would not be a color that I would use in a real design. However, I picked this color in order to bring about a point.

In our CSS rule, change **#99CCFF** to **#9cf** ...

```
<style type="text/css">

body {
    background-color: #9cf;
}

</style>
```

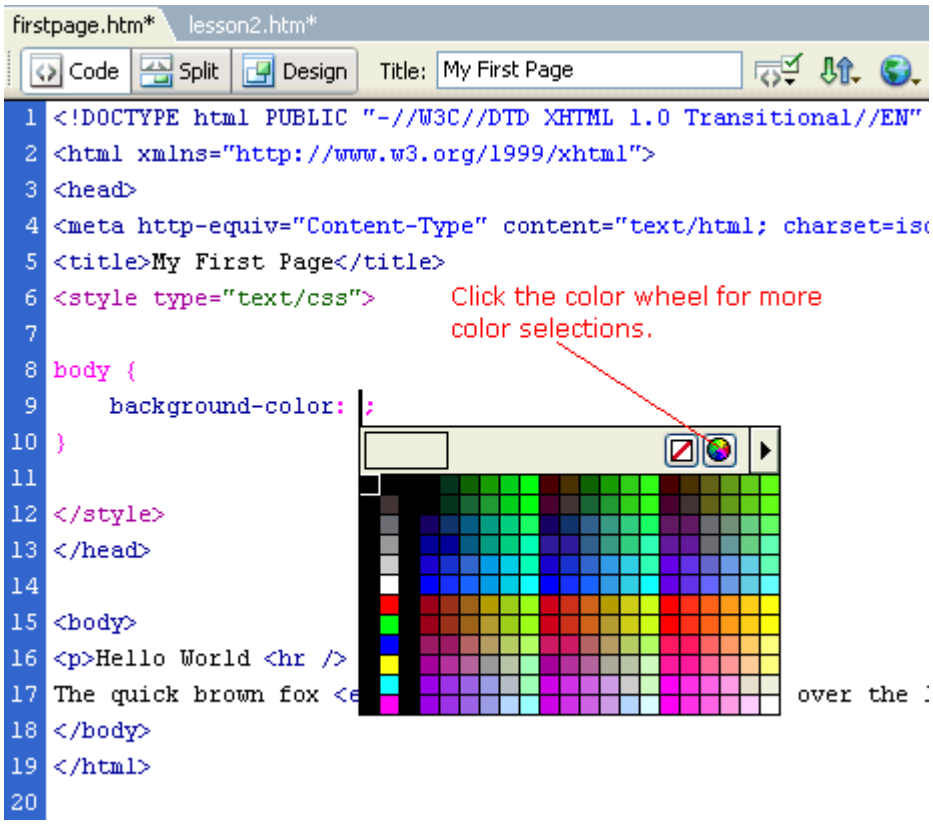
And you get the same bright color. Hence, **#9cf** this is a short-cut notation for **#99CCFF** when writing CSS colors. Also the upper and lower case letter both works. If the first two digits (the red component) are the same, **and** the middle two digits (the green color component) are the same, **and** the last two digits (the blue component) are the same, then you can use a single digit for each of the three pairs.

Adjust the background color

Okay, now that I have made the point about the shorthand notation, let dim the background color so that we don't hurt our eyes.

Go to the code view this time and delete the **#9cf** from our CSS rule.

At the location shown, press **Ctrl-Space** to bring up Dreamweaver's ***code-hinting.***



Click the color wheel and pick a color of your choosing.

In fact, if your color is one of the 17 colors listed **here**, then you can write it using the colorname instead. For example,

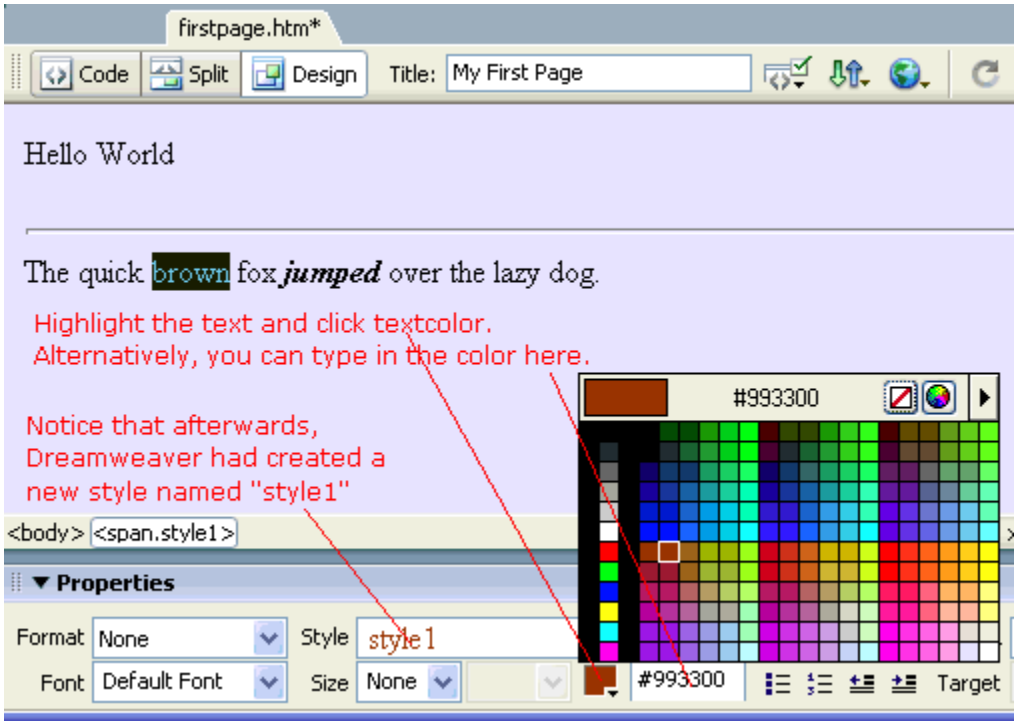
```
body {
  background-color: blue;
}
```

But the color that I happened to pick was a light gray of #E6E0FE;

```
<style type="text/css">
<!--
body {
    background-color: #E6E0FE;
}
-->
</style>
```

Stylizing the Text

Going back to the design view, change the text color of the word "**brown**" to the color brown...



In the code view, we see below that Dreamweaver has a **** tag around the word "brown".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Tra
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/ht
<title>My First Page</title>
<style type="text/css">

body {
    background-color: #E6E0FE;
}
```

```
.style1 {color: #993300}
</style>
</head>

<body>
<p>Hello World <hr />
The quick <span class="style1">brown</span> fox
</body>
</html>
```

Attribute name

attribute value

This **** tag has an ***attribute*** of ...

class="style1"

The **class** is the attribute name and the **style1** is the attribute value.

This tells that browser that the text in between the span tags are a little bit special. They are of a special ***class***. The class that this item belongs in is named **style1**. **style1** is an arbitrary name that was given by Dreamweaver. (It is not that descriptive of a name; so we will give if a better name latter.)

In addition, Dreamweaver has added another CSS rule.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Tra
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/ht
<title>My First Page</title>
<style type="text/css">

body {
    background-color: #E6E0FE;
}
.style1 {color: #993300}
</style>
</head>

<body>
<p>Hello World <hr />
The quick <span class="style1">brown</span> fox
</body>
</html>
```

The very last semicolon before the } can be omitted, but I like to add one here anyways.

The content of our particular **** tag will be styled by the **style1** CSS rule.

Look at the CSS rule, this time the rule does not start out with the name of an element. Instead the rule starts with the classname preceded by a dot. The dot indicates to the browser that what is to follow is a classname. In plain English, the rule says ...

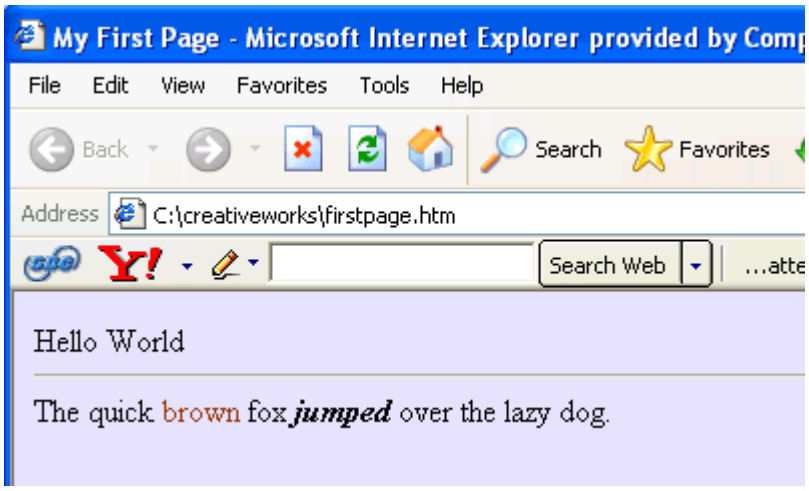
"For all elements with the class "style1", please style the color to #993300."

"Style1" is an arbitrary name that Dreamweaver has given us. It is best to give it our own more informative name such as "browntext".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transition
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; cha
<title>My First Page</title>
<style type="text/css">

body {
    background-color: #E6E0FE;
}
.browntext {color: #993300}
</style>
</head>

<body>
<p>Hello World <hr />
The quick <span class="browntext">brown</span> fox <em>
</body>
</html>
```

Changed our style name

And see that it still works just the same when we preview it in our browser...



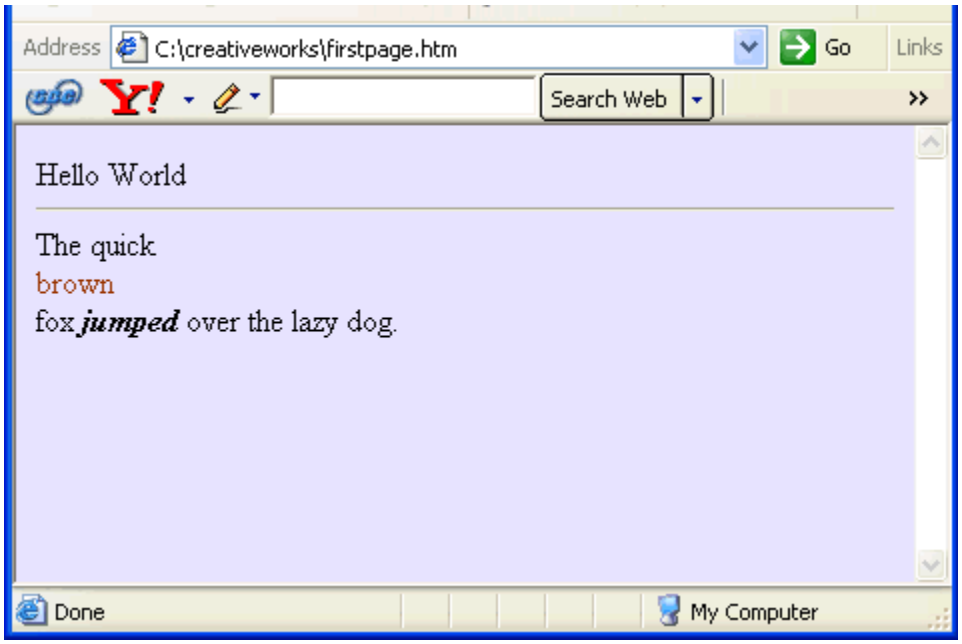
Span versus Div

What is the difference between **** tag and a ** ** tag? The main difference is that a **** tag is an ***inline*** element and ** ** tag is a ***block*** element. To demonstrate what this means, replace the **** with a ** ** tag...

```
<body>
<p>Hello World <hr />
The quick <div class="browntext">brown</div> fox <em><strong>jumped</str
</body>
```

replaced with div tag

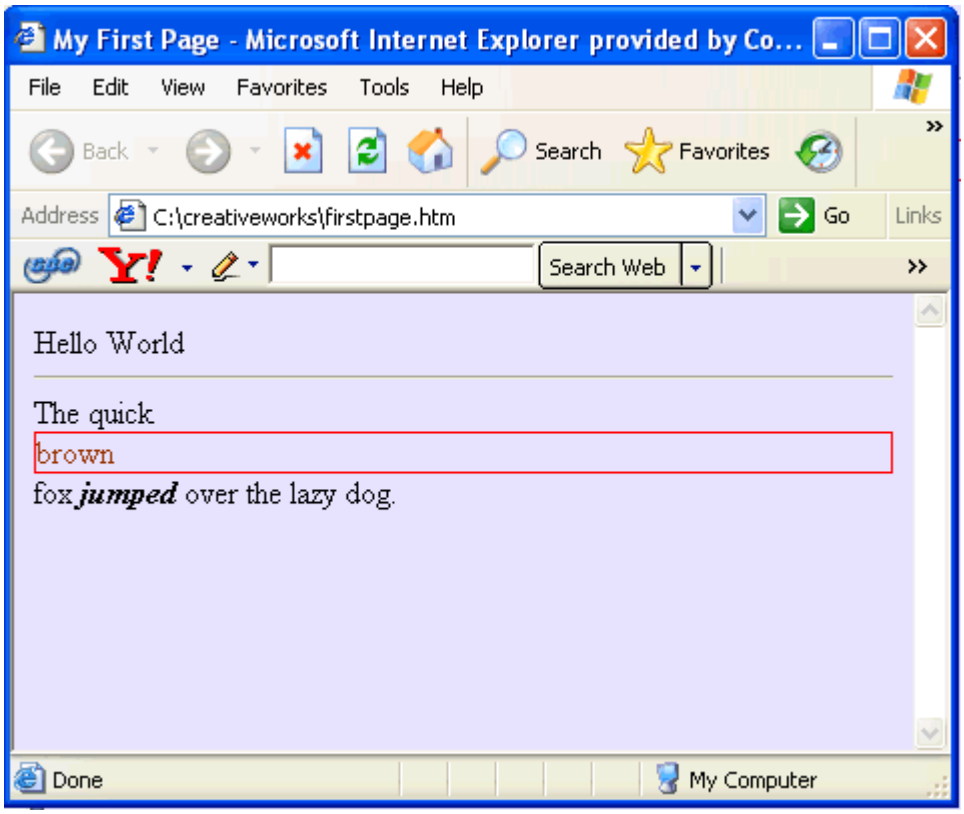
And see that you get ...



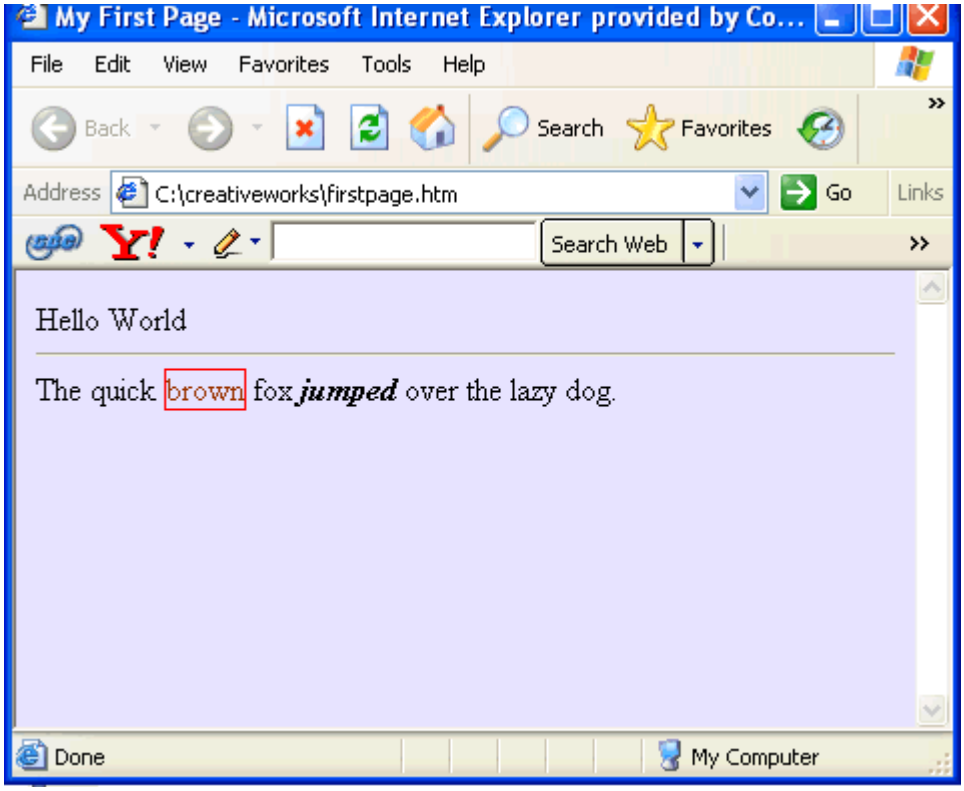
The ** ** tag takes as much of the width of the page as it can. It blocks out the whole section of the page, so that anything following it has to go to the next line beneath it. The **** tags takes as little space as possible, only enough to contain its contents.

For the purpose of illustration, I have put a border around the element so that you can see the difference.

Here it is using a div tag...



And here it is using a span tag...



I like it better with the span tag, so remove the div tag and put back the span tag so that your code looks like ...

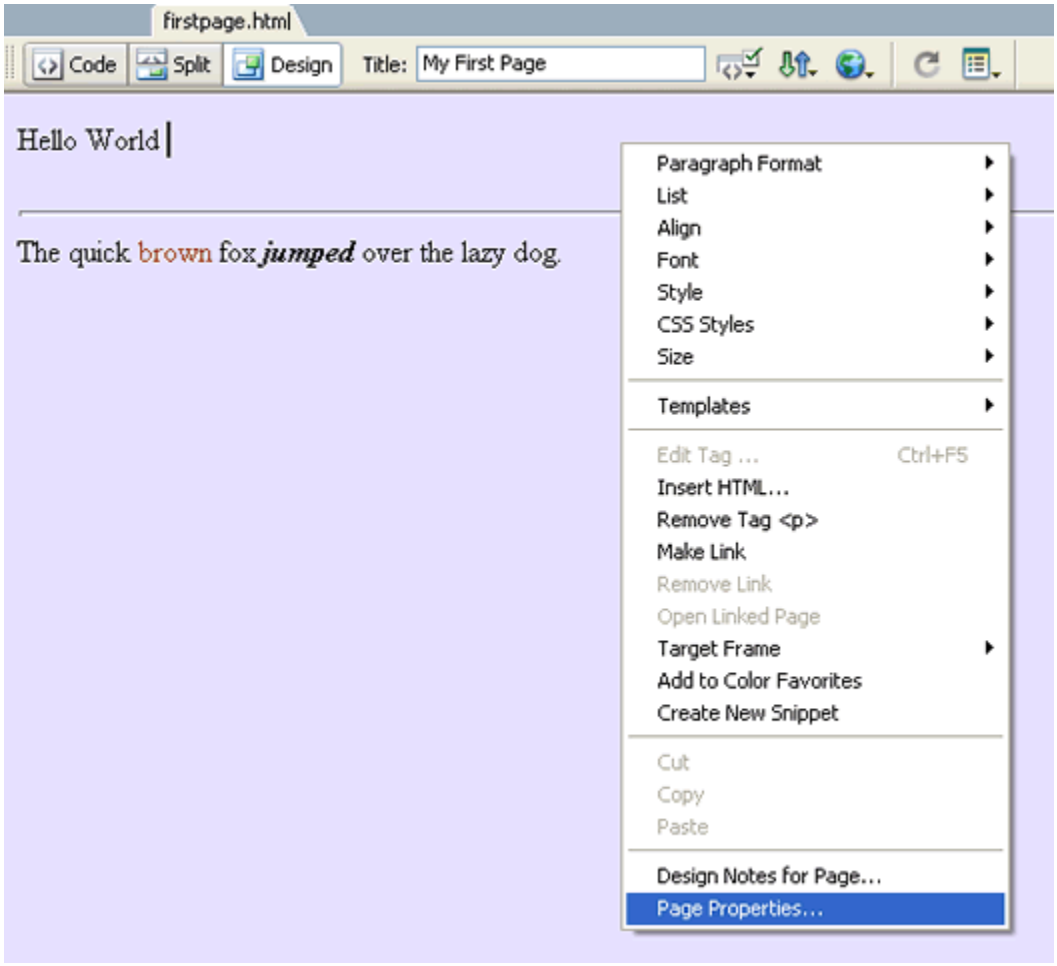
```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html;
   charset=iso-8859-1" />
5  <title>My First Page</title>
6  <style type="text/css">
7  <!--
8  body {
9      background-color: #E6E0FE;
10 }                                   Put back the span tag.
11 .browntext {color: #993300}
12 -->
13 </style>
14 </head>
15
16 <body>
17 <p>Hello World <hr />
18 The quick <span class="browntext">brown</span> fox <em><strong>jumped
   </strong></em> over the lazy dog. </p>
19 </body>
20 </html>
```

As another example, note that the **<p>** tag is a block element; while the **** tag is an inline element. So now you can see the difference between an inline element and a block element.

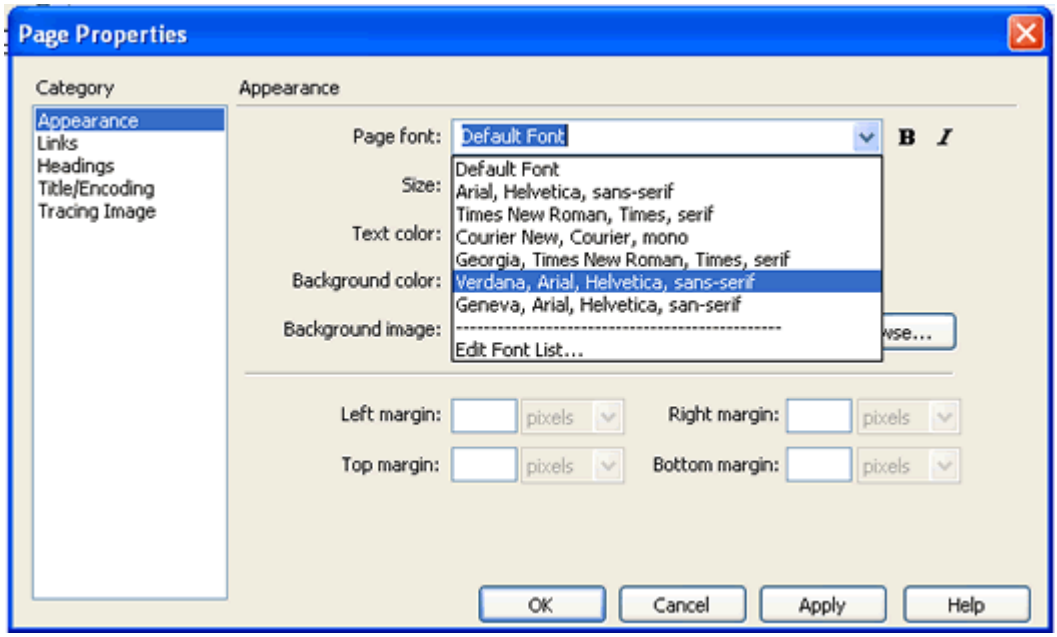
Here is what we have so far: **firstpage.html**

Font-Family

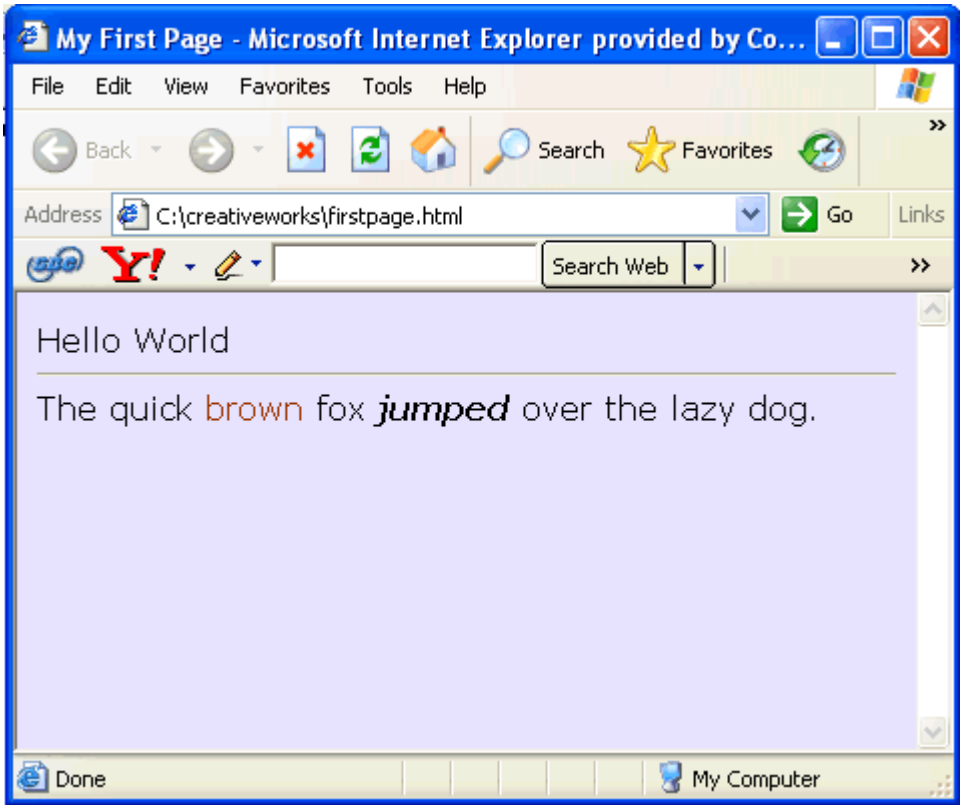
Open up **firstpage.html** in the design view. The default font the you see currently on your page is the default ***serif*** type font. Let's change our font to a ***non-serif*** font such a Verdana. In the design view, right-click and select **Page Properties**...



In the **Page Properties** dialog under the **Appearance** category, set the **Page Font** to Verdana and click **OK**.



When you test the page, notice how the non-serif font below is different from the serif font above.



What you set in the **Page Properties** dialog applies to the whole page. So when you look at the code view, it would not be too surprising that a CSS rule was added to the **<body>** tag.

```
<style type="text/css">

body {
    background-color: #E6E0FE;
}
.browntext {color: #993300}
body,td,th {
    font-family: Verdana, Arial, Helvetica, sans-serif;
}
</style>
```

rule applies to all three of these elements: body, td, th

Actually, this CSS rule applies to the **<body>** tag, the **<td>** tag, and the **<th>** tags. The **<td>** and **<th>** tags are used by **<table>** elements which we don't have on our page. So we will get rid of it, and you are left with ...

```
<style type="text/css">

body {
    background-color: #E6E0FE;
}

.browntext {color: #993300}

body {
```

These two rules applies to the same body element and thus can be combined.

```
    font-family: Verdana, Arial, Helvetica, sans-serif;
}

</style>
```

Since we now have two rules that applies to the same element (see above), we can combine them into one (as below) ...

```
<style type="text/css">

body {
    background-color: #E6E0FE;
    font-family: Verdana, Arial, Helvetica, sans-serif;
}

.browntext {color: #993300}

</style>
```

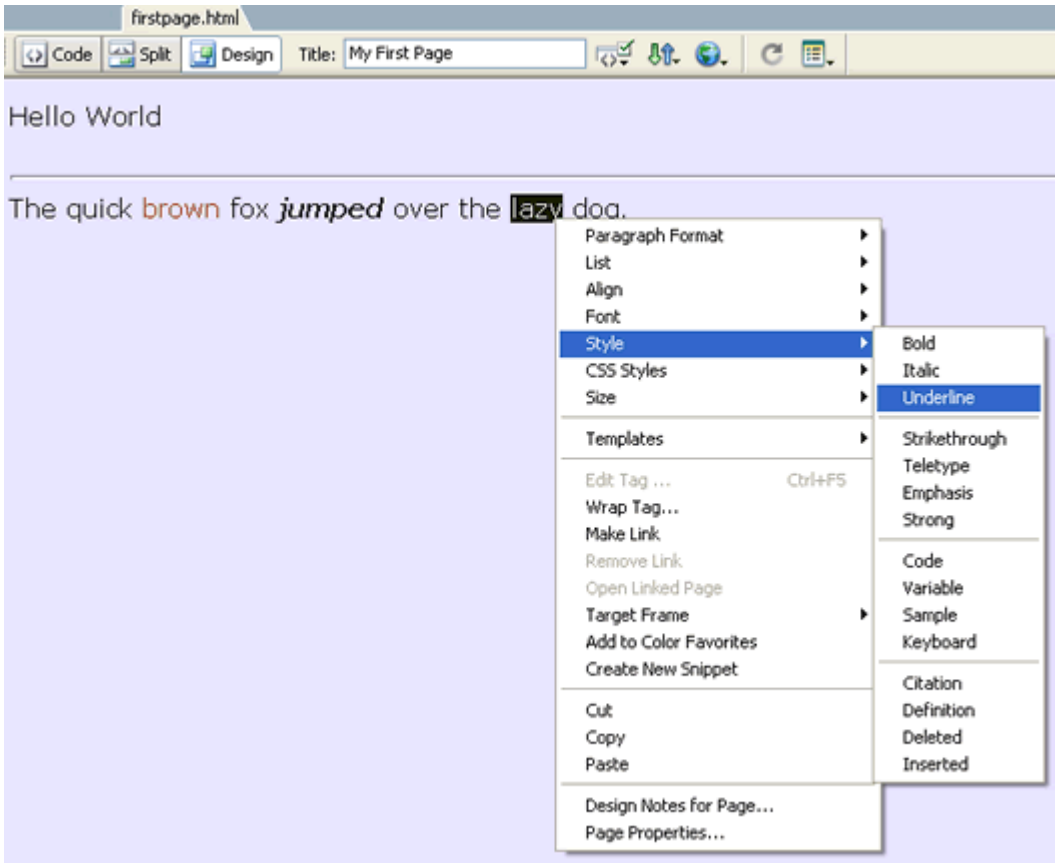
The rules for an element is separated by semicolons.

Combining rules is prefered because less text in your file means faster download of your pages.

For the value of our **font-family** property, we have serveral fonts listed: **Verdana, Arial, Helvetica, sans-serif**. This means that if the user's computer has the Verdana font installed, the browser will render the text in that font. If not, the browser will try Arial font, and then try Helvetica. If all else fails, the browser will render it in a generic sans-serif font.

Underlining Text

To bring up a point, let's now try underlining the word "lazy" by highlighting and right-clicking it design view.



And see that it generates the **<u>** tag.

```
<body>
<p>Hello World <hr />
The quick <span class="browntext">brown</span> fox
<em><strong>jumped</strong></em> over the <u>lazy</u> dog. </p>
</body>
```

The <u> tag is deprecated

This is not good, because the **<u>** tag is **deprecated**, which mean that "it will work, but you should not use it because there are better alternatives ". For example, you can use the following CSS style instead...

```
<style type="text/css">
body {
    background-color: #E6E0FE;
    font-family: Verdana, Arial, Helvetica, sans-serif;
}

.browntext {color: #993300}

.underlinedtext {
    text-decoration: underline;
}
</style>
</head>

<body>
<p>Hello World <hr />
The quick <span class="browntext">brown</span> fox
<em><strong>jumped</strong></em> over the <span class="underlinedtext">lazy</span>
</body>
```

where **underlinedtext** is an arbitrary name that I came up.

You really shouldn't underline normal text anyways; because from an usability standpoint, they get confused with links. If it really is a link, then it gets underlined by default without us needing to stylize it. I'll show you how to create links in a later section.

Notice the dot in front of the words **underlinedtext** and the dot in front of the **browntext** in the above code. The dot is known as the ***class selector***. Those CSS rules that has these dot will look for element of that ***class***. In other words, the **.underlinedtext** selector rule applies to all elements of class "**underlinedtext**". And looking at the code above, the **** tag is of class "**underlinedtext**" because it has the "**class**" attribute of value **underlinedtext**.

More Deprecated Elements

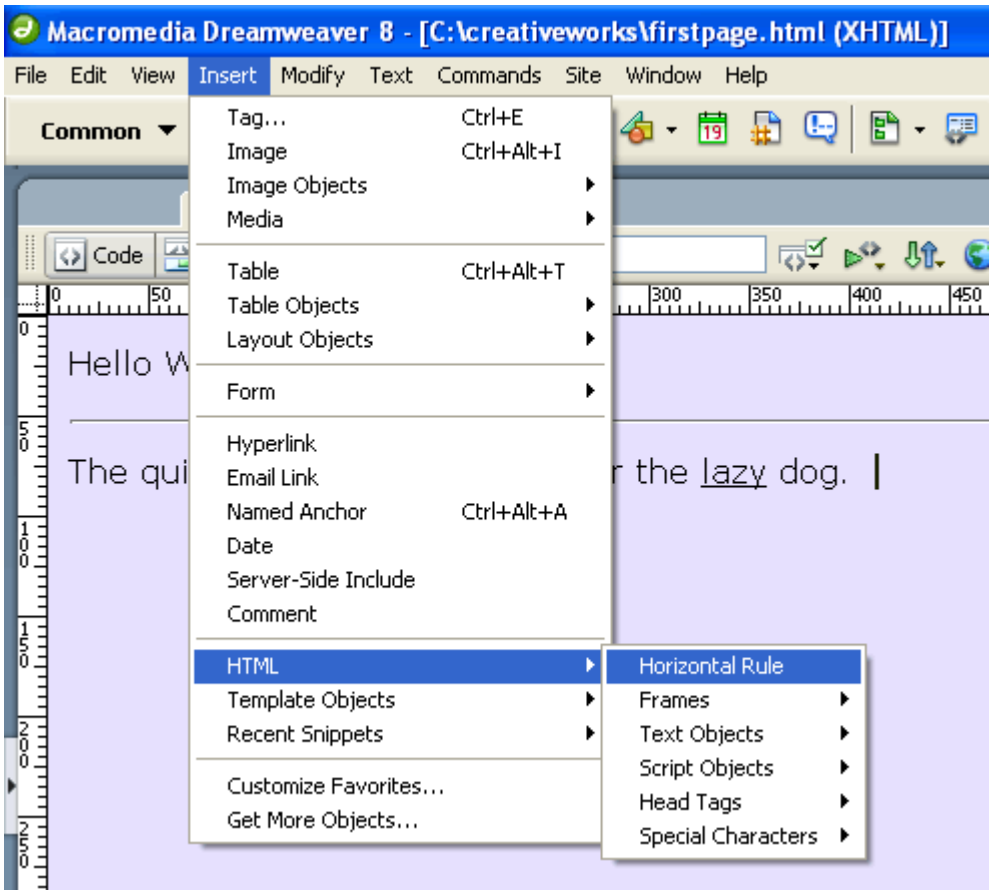
Here are more deprecated elements that we should not use.

The **<font>** tag is a deprecated tag. Before the days of CSS, the **<font>** tag is used to stylize text. But now it is such bad practice to use the **<font>** tag that I will not even show you an example of it (least you fall into its trap). You have already seen the preferred way of stylizing text via CSS properties such as font-family, color, etc. And we will learn more later.

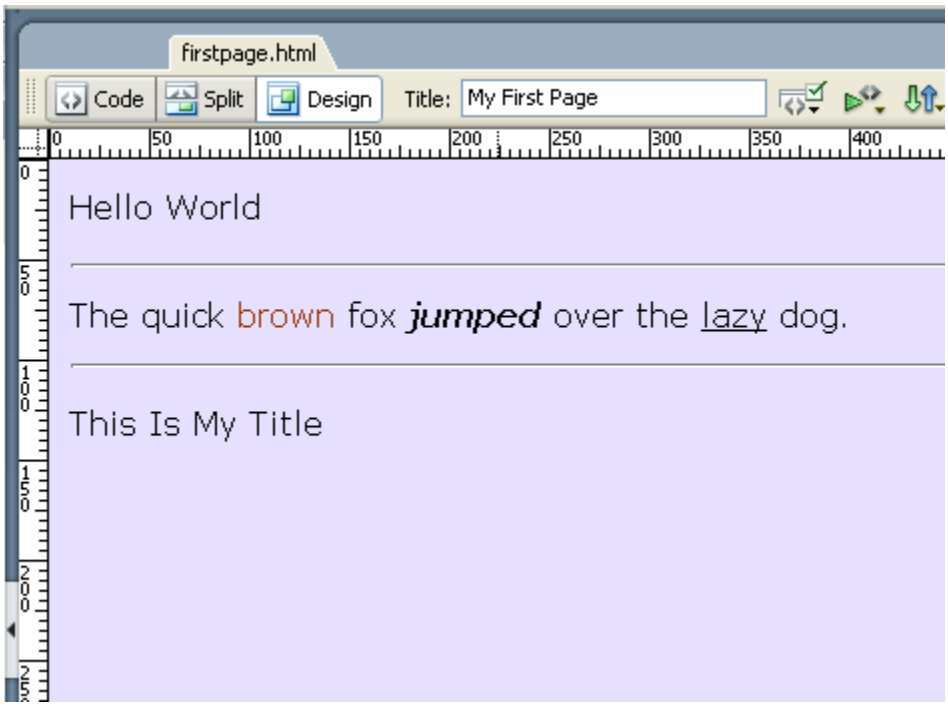
Lorem Ipsum Text

We are going to add another section of text below what we have on the page. So let's add another horizontal rule. This

time use the menu as shown...

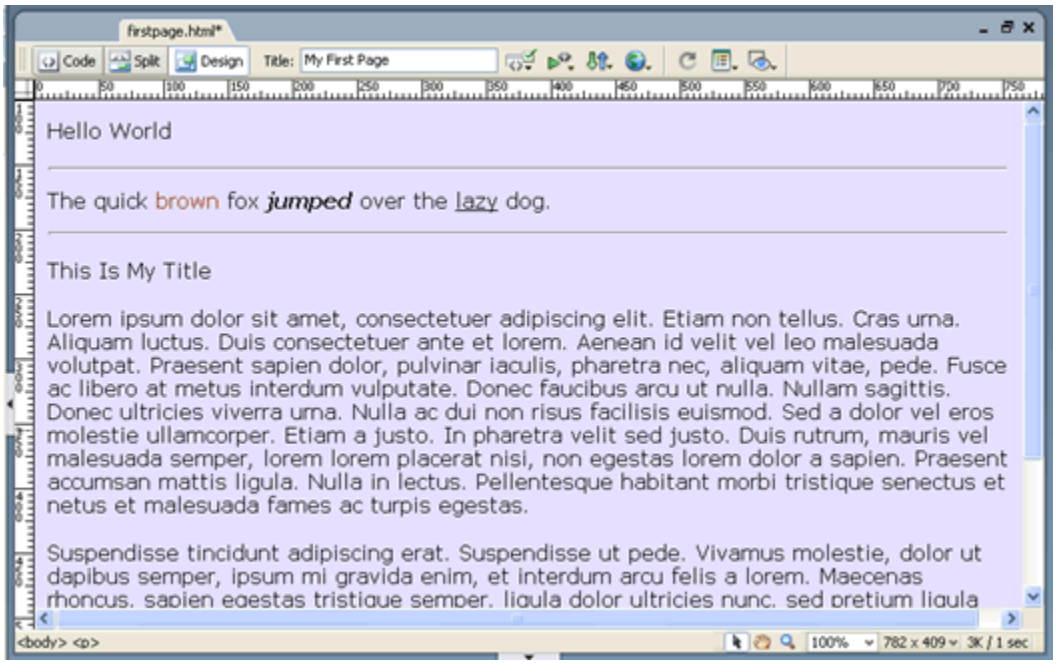


The new horizontal rule will now be selected. Press the **right arrow** key so that it is no longer selected and press **Enter** for a new line. Then type in a heading as shown below...



Lorem Ipsum text are filler text that web designers uses to provide a look-and-feel of real text without actually having to

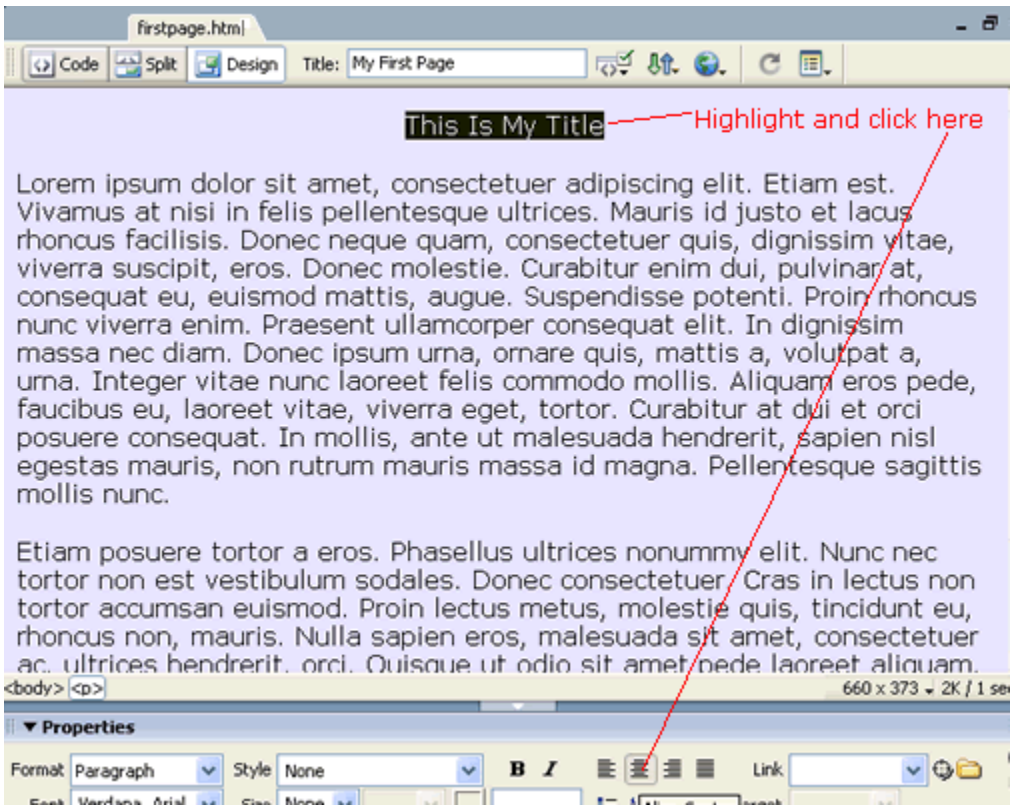
compose a story themselves. They are randomly generated letters that are statistically choosen to make it look like a language. Generate two paragraphs from the **Lorem Ipsum Generator** and copied and pasted it to Dreamweaver's Design view as shown.



If you quickly look at the code view, you will see that paragraphs are marked off by the ** ** and ** ** tags. Switch back to the design view.

Centering the Title

The ** ** text is also deprecated. So how do we center text? To center this title heading on this page, we do...



And the code you get is ...

```
<body>
<p align="center">This Is My Title</p>
<p>Lorem ipsum dolor sit amet, consectetuer adipiscir
<p>Etiam posuere tortor a eros. Phasellus ultrices nc
</body>
```

attribute name

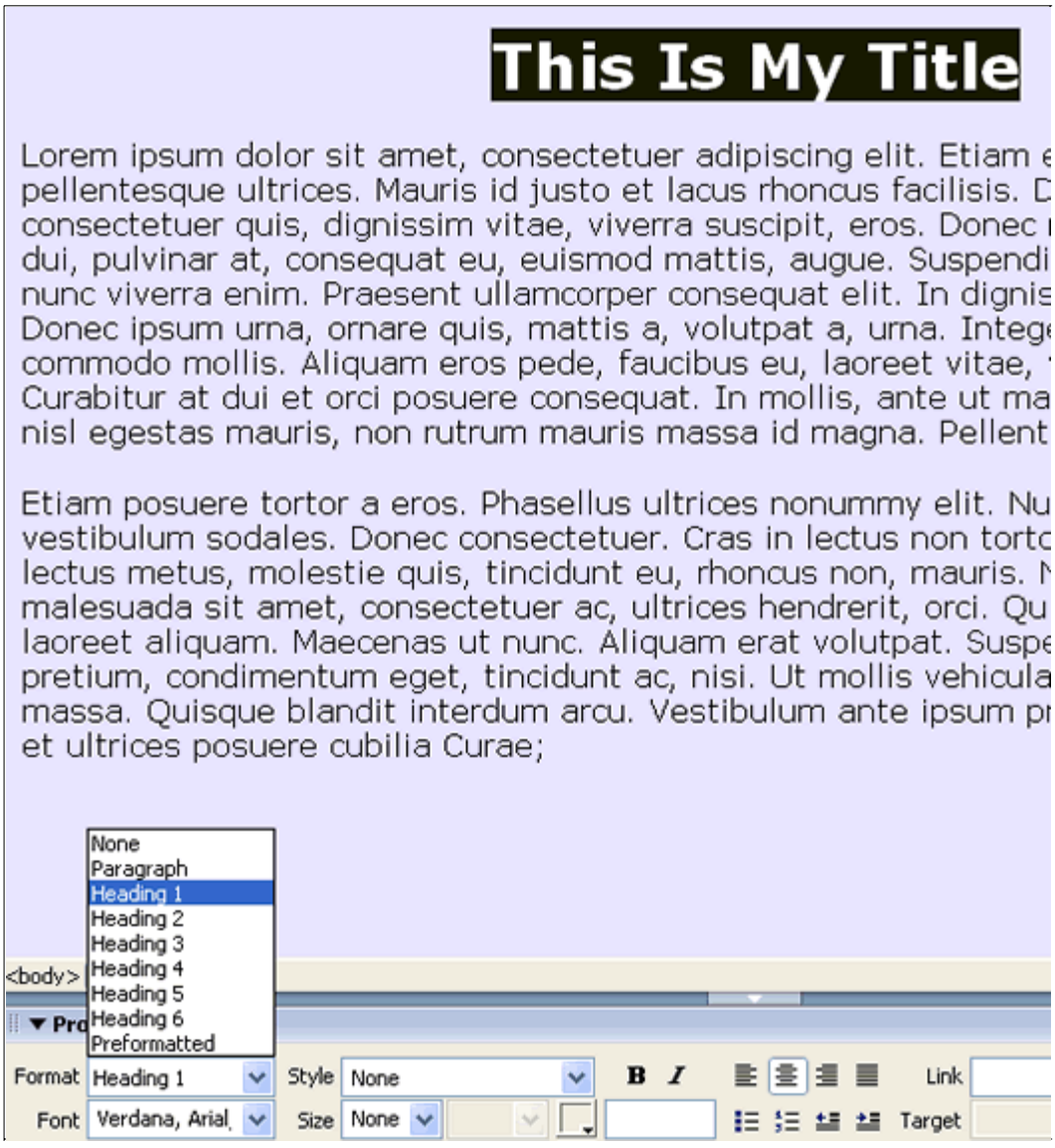
attribute value

This time, Dreamweaver didn't give us a CSS rule. Instead it added the **align** attribute to the paragraph tag.

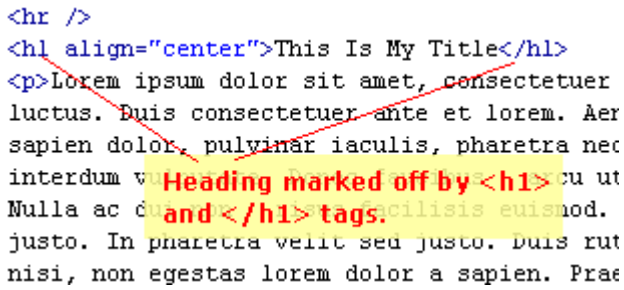
Headings

Normally, the heading title should be a bit larger than the other text on the page. Let's make it so. That is exactly what the heading tags are for. There are six possible heading tags for you to choose from: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>. <h1> is the top-level heading. It will render the font the largest. <h2> is the next subheading with a smaller size font. <h3> is a sub-sub heading. Etc.

In the design view, set our heading to **heading 1**.

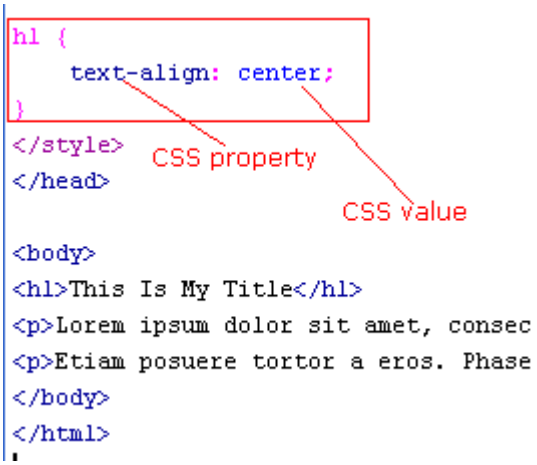


and see that our **<p>** tag has become the **<h1>** tag...



First CSS

Instead of using an attribute (as in the above) to center our tag, some would argue that it is better to use CSS (Cascading Style Sheet) instead. To do this, remove the attribute and add the following CSS rule for our <h1> element ...

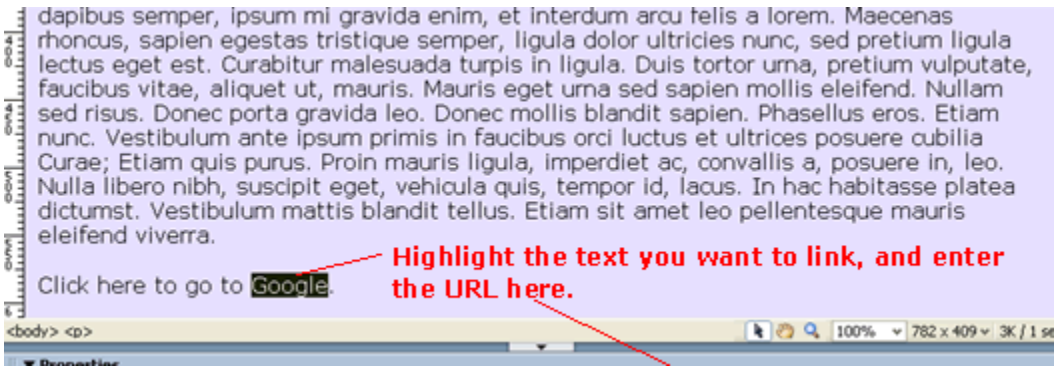


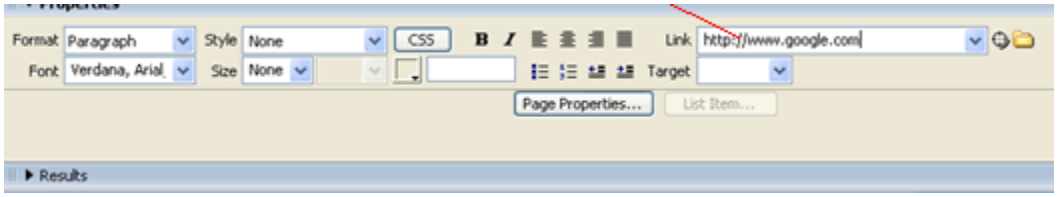
The two code snippets seen above are equivalent. Note that attributes uses an equal sign (=) to separate the name and value; whereas CSS uses the colon (:) to separate the property and value. The attribute name does not need to be the same as the CSS property. For example, the attribute name was "**align**". The corresponding CSS property is "**text-align**". Similarly, the attribute values need not be the same as the CSS values.

It is often hard for beginner to remember all the properties and values and attributes. This is where Dreamweaver's code-hinting comes in handy. The code hints works for attributes just as well as for CSS. Just press **Ctrl-Space** at the location where you are about to type something, and Dreamweaver will help you out.

Creating Links

Here is an example of how to create a link that goes to Google when clicked.





This is the code for it...

```
<p>Click <a href="http://www.google.com">here</a> to go to google.  </p>
```

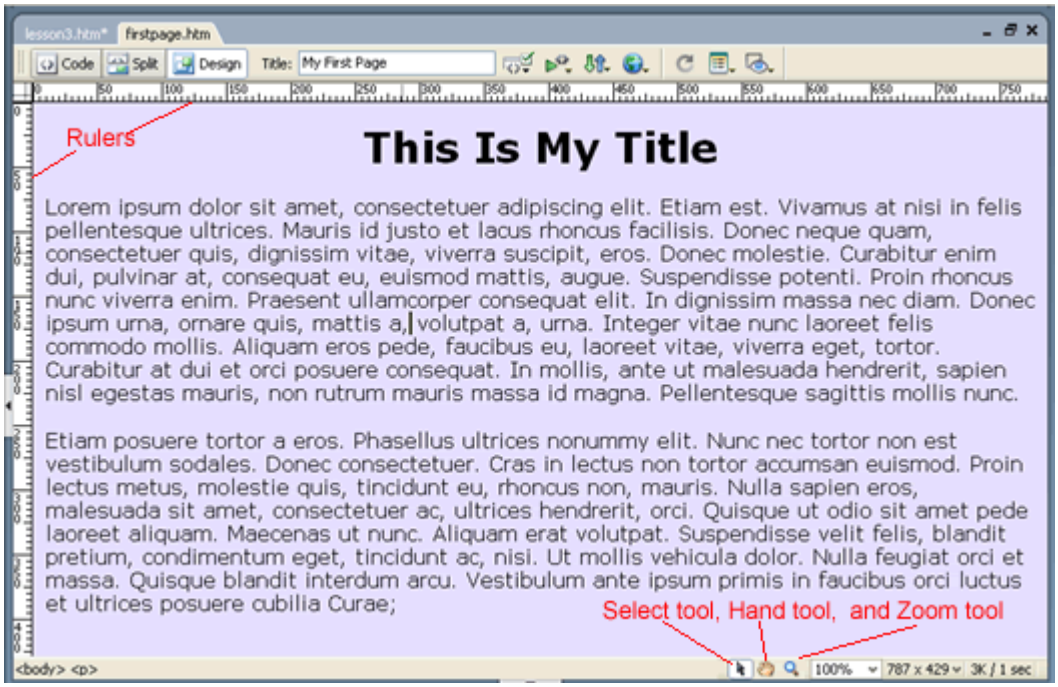
The body of the **<a>** tag is the text of the link and the **href** is the URL where the link goes to.

Here is my **firstpage.htm** file so far: **firstpage.htm**

You can view the page in your browser by clicking on the link. Then do a **View -> Source** in your browser to see the HTML code. Alternatively, you can download the entire file, by right-clicking on the link and select **Save Target As**.

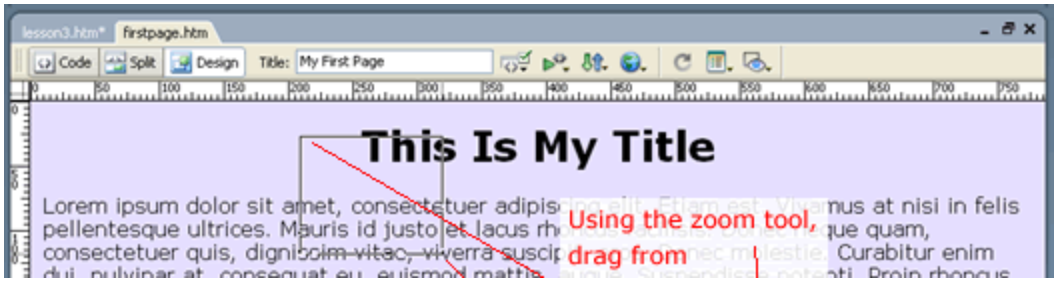
More on Dreamweaver's Design View

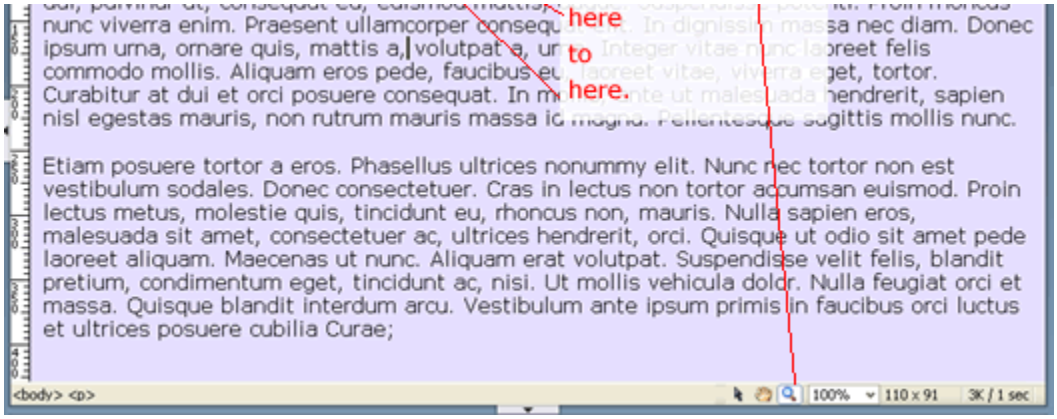
After opening the **firstpage.htm** and then going to **Design View**, you will notice that it has rulers.



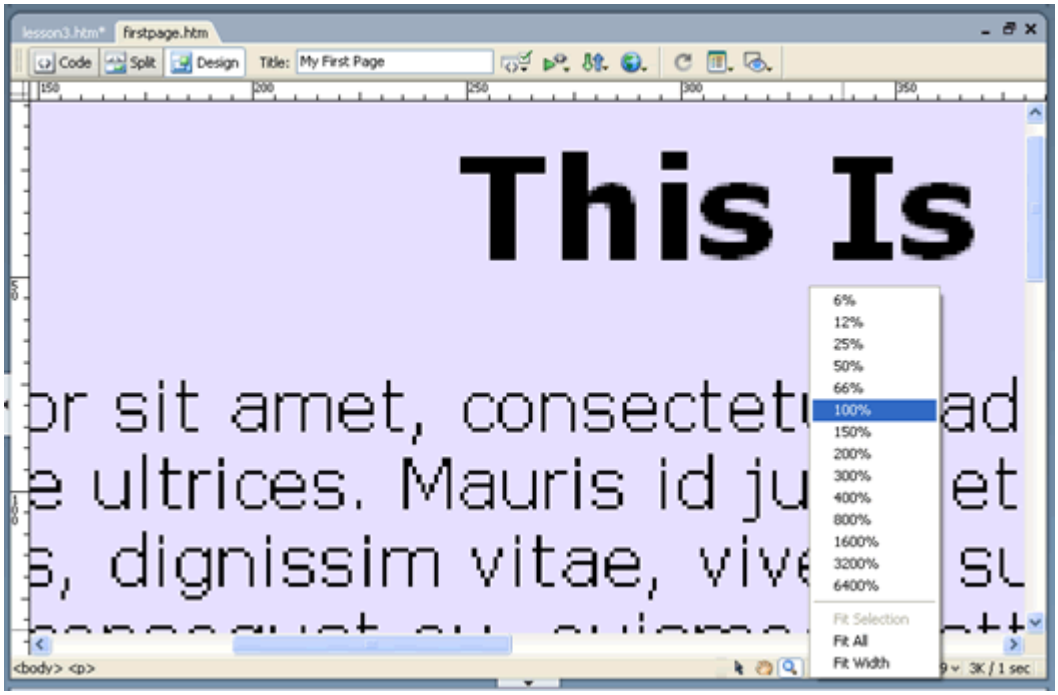
There is also the **Select tool**, which should be the tool that you would normally be using and is selected by default. Then there is the **Hand** tool and the **Zoom** tool.

First play with the **Zoom** tool by ...





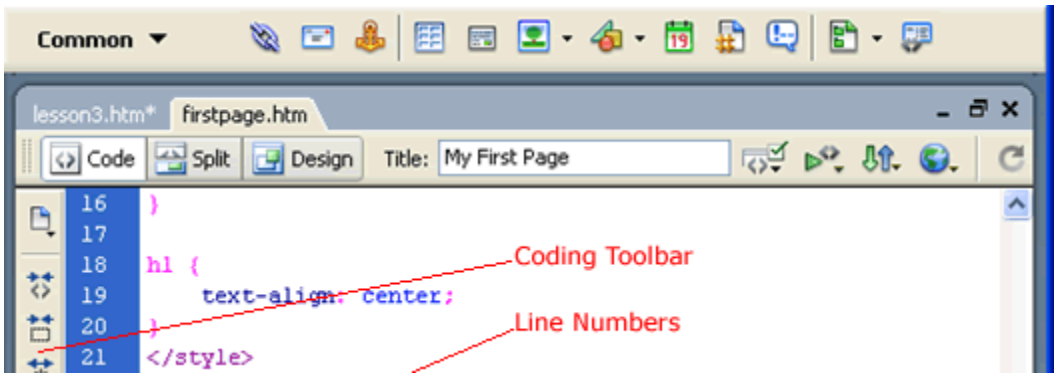
Look, you get zooming in Design View! Now you can use your Hand tool and move the page about. Watch as the rulers continues to tell you what scale and where you are at all times.

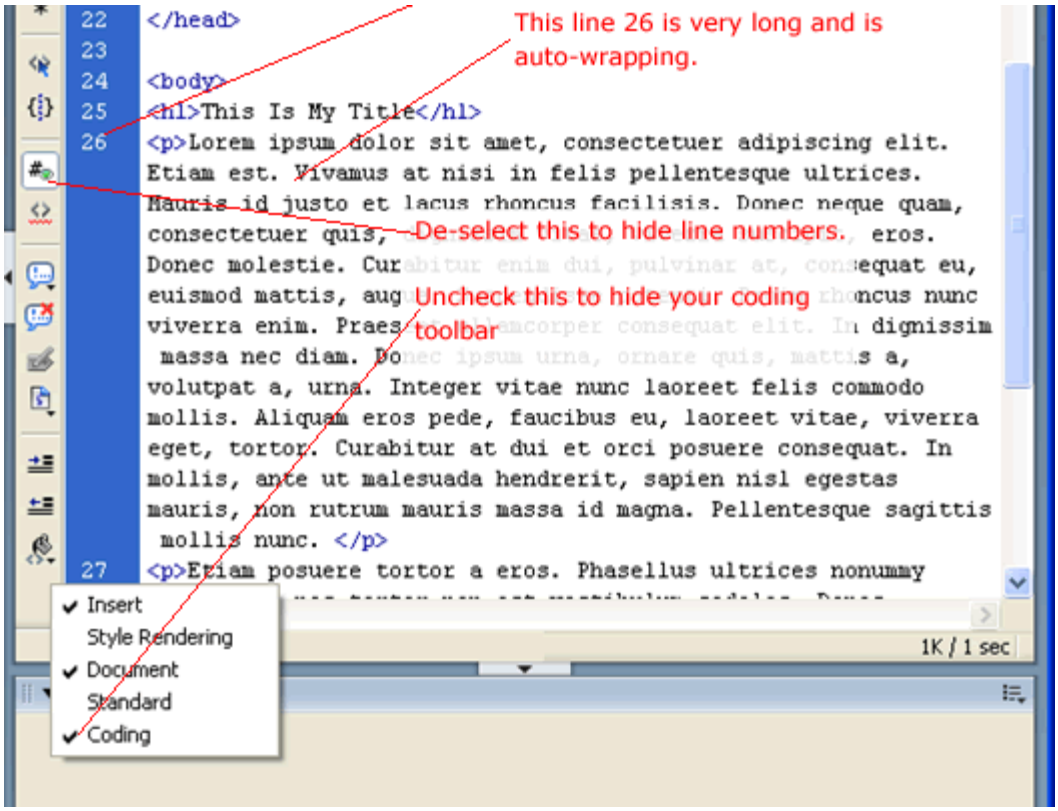


Okay, enough fun. Click the Magnification droplist and go back to 100% from the menu shown above. And then click on the Select tool -- that is the tool that you should normally be in.

Code View of Dreamweaver 8

Now switch to the code view to see the new Coding toolbar and line numbers which you can turn on and off.





If you had turned off your coding toolbar and want it back, just right-click on any other toolbar to bring up the context menu and checkmark **Coding**.

You see that I had typed in (actually pasted-in) the Lorem Ipsum paragraph as one long line (see line 26 above). Dreamweaver had auto-wrapped that line. Personally, I rather that Dreamweaver display long lines as they are and not wrap them. So I turn off auto-wrap by **View -> Code View Options ->** uncheck **Word Wrap**. Make sure that you are in Code view when you do this.

That is a good enough of a start on HTML. We are now done *playing* with **firstpage.html**. In the next lesson, you will start a *real* website based on the comp created in the Intro Fireworks course.

Starting a Website

We are now ready to build our **CreativeWorks** website. In the Fireworks lessons, we had created the comp for the website that looks like this ...

You can get the Fireworks file by opening the **creativeworks.png** link in your browser and then right-click to **Save Picture As**.

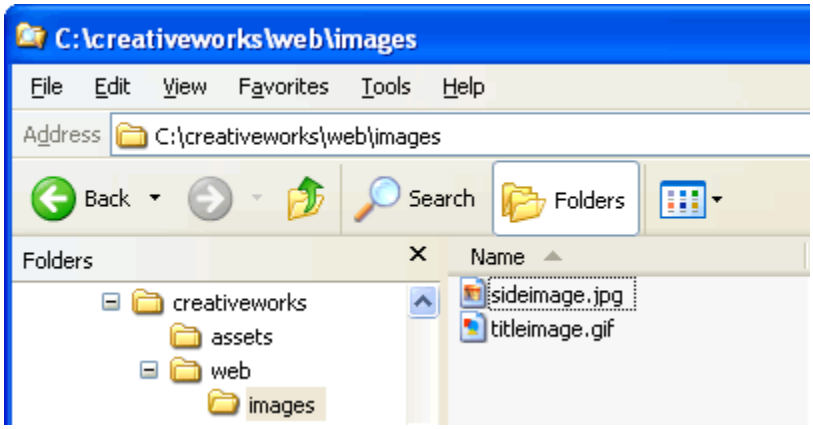
Save this **creativeworks.png** file in a subfolder called **assets** under the **creativeworks** folder that you had created in the previous lessons.

We now create another subfolder called **web** under the **creativeworks** folder. The web folder will be the actual files of our website that would be uploaded to the web server.

In Fireworks, we had also sliced out and exported these two images:

You can download them (**sideimage.jpg** and **titleimage.gif**) by opening the links in your browser and right-clicking on the images to do a **Save Picture As**. Save them in a subfolder called **images** under the **web** folder.

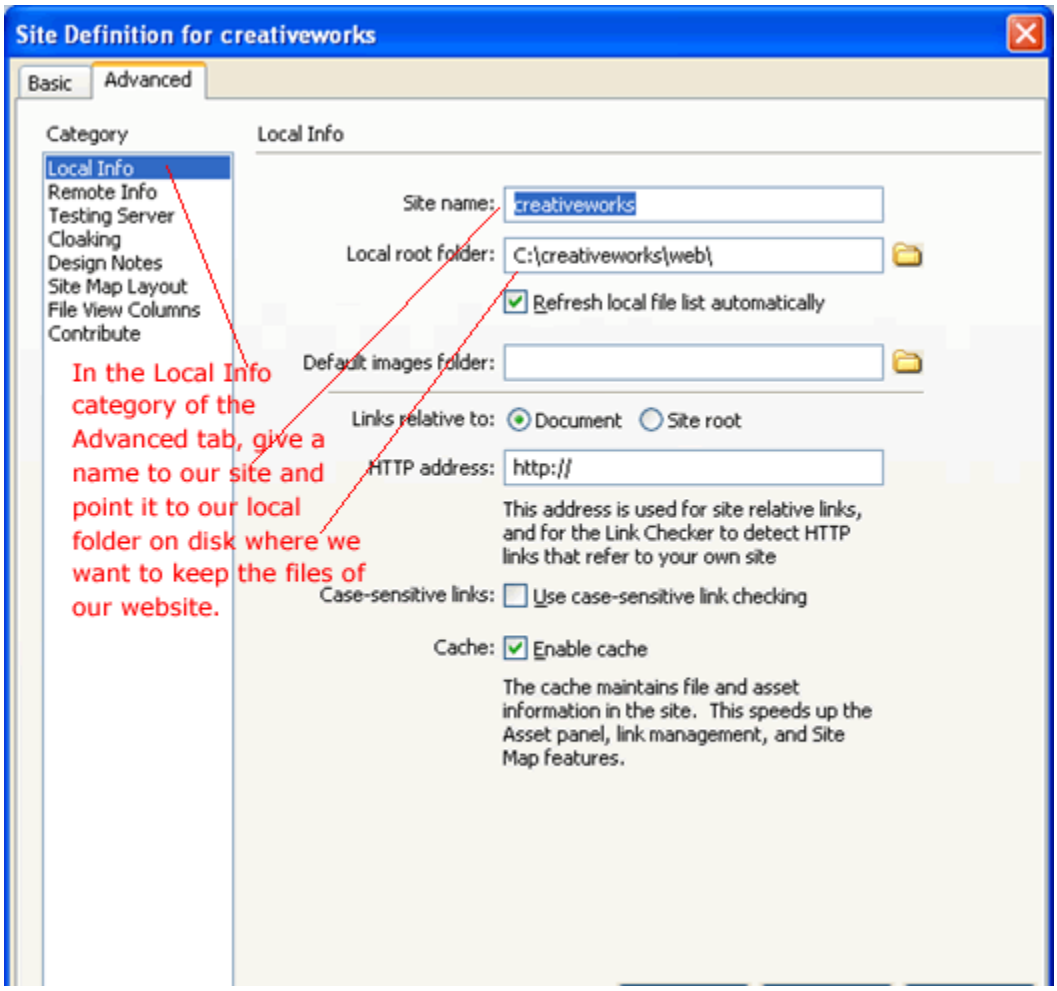
So your directory structure should look like ...



Setting Up Dreamweaver Site

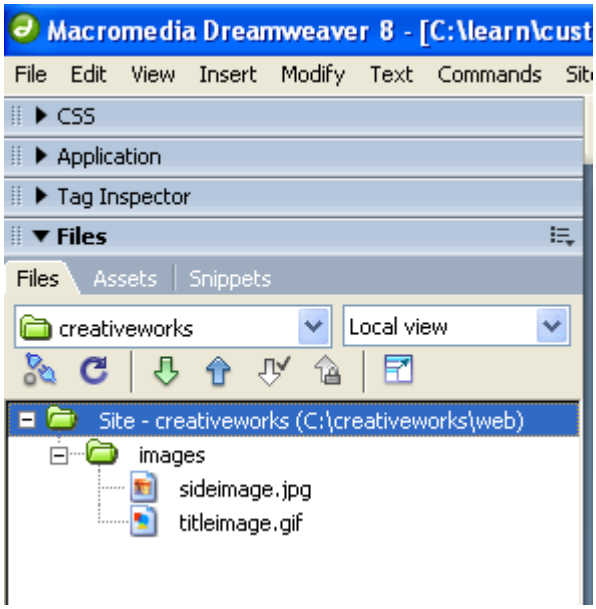
The first thing to do in Dreamweaver in building a website is to setup a Dreamweaver site. Do **Site -> New Site**...

In the **Local Info** category of the **Advanced** tab, give our site the name **creativeworks** and point it to our **web** directory as shown.



We will not worry about the other advanced settings for now. Click **OK**.

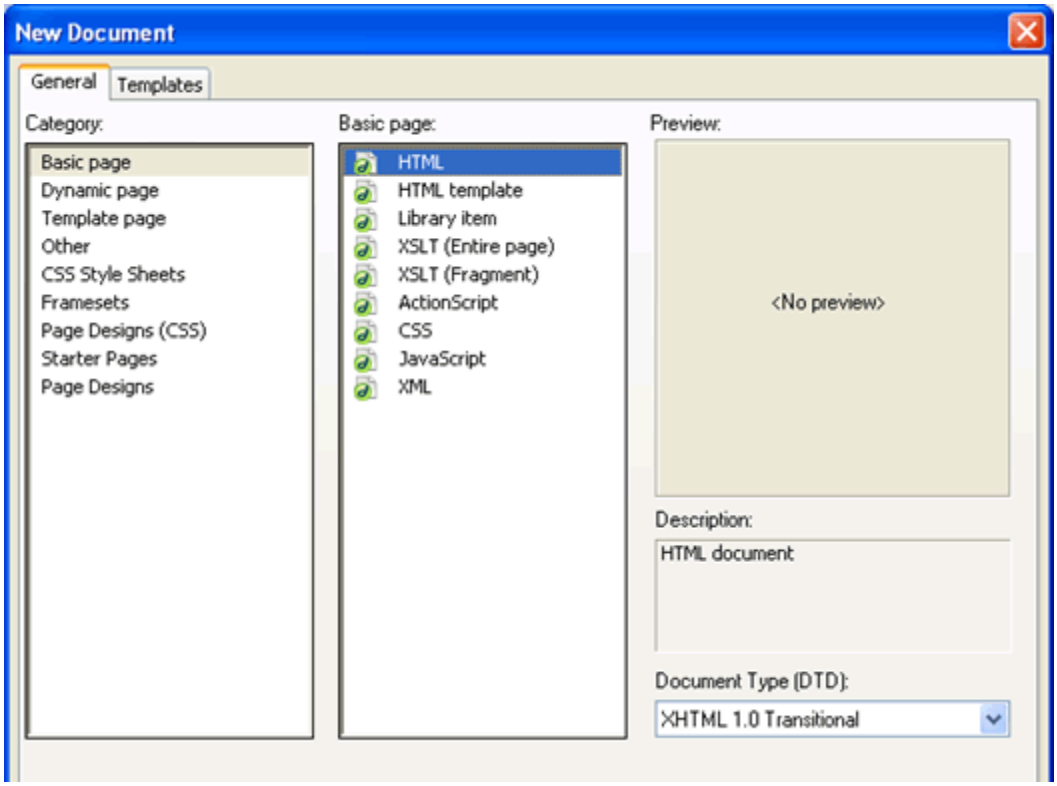
If you look at the Files tab of the Files panel, you will see that it shows the contents of our **web** folder.

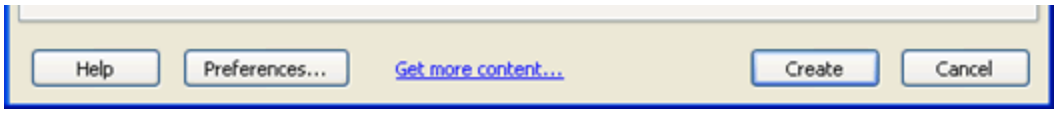


This is where we will manage and create the files for our website. We already have two images of our website in the **images** folder. And it is typical in industry to place website images under the **images** folder of our website.

Creating the Home Page

To create the first page of our website, do **File -> New**



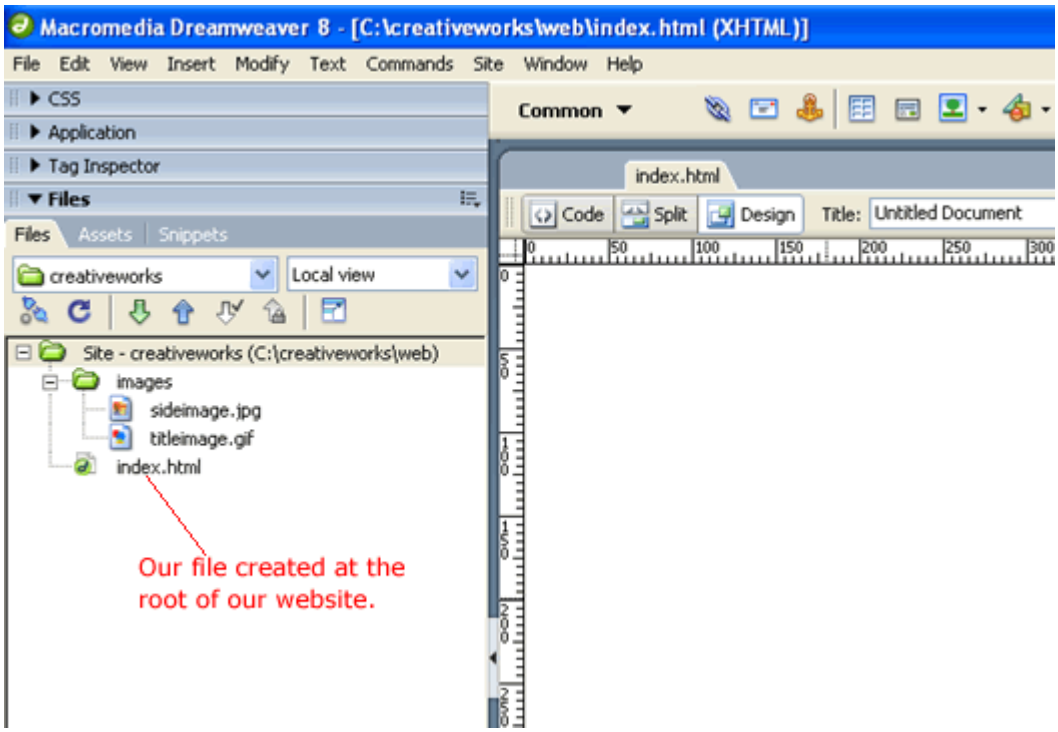


and click **Create** to the setting shown.

The front page of a website, also known as the home page, is the first page that the users will see. This page is ususally named **index.htm** or **index.html** because most web servers are set up so to show this page by default if the user does not specify a page in the URL.

We save our new page by doing **File -> Save** and give it the name **index.html**

In the Files panel, we see the file created in the ***root*** folder of our website.



Supposing that we had obtained the domain name **creativeworks.com** and had uploaded the **index.html** file to our web host, then any body in the world can view our page by typing in **http://www.creativewworks.com/index.html**. If you omitted the **www** as in **http://creativeworks.com/index.html**, it will still work because then the webserver will assume the **www** subdomain by default. And if the webserver is setup to know **index.html** as the default page (and virtually all webserver are set up this way), then the user will still be able to get to our page by typing only **http://creativeworks.com** in the browser.

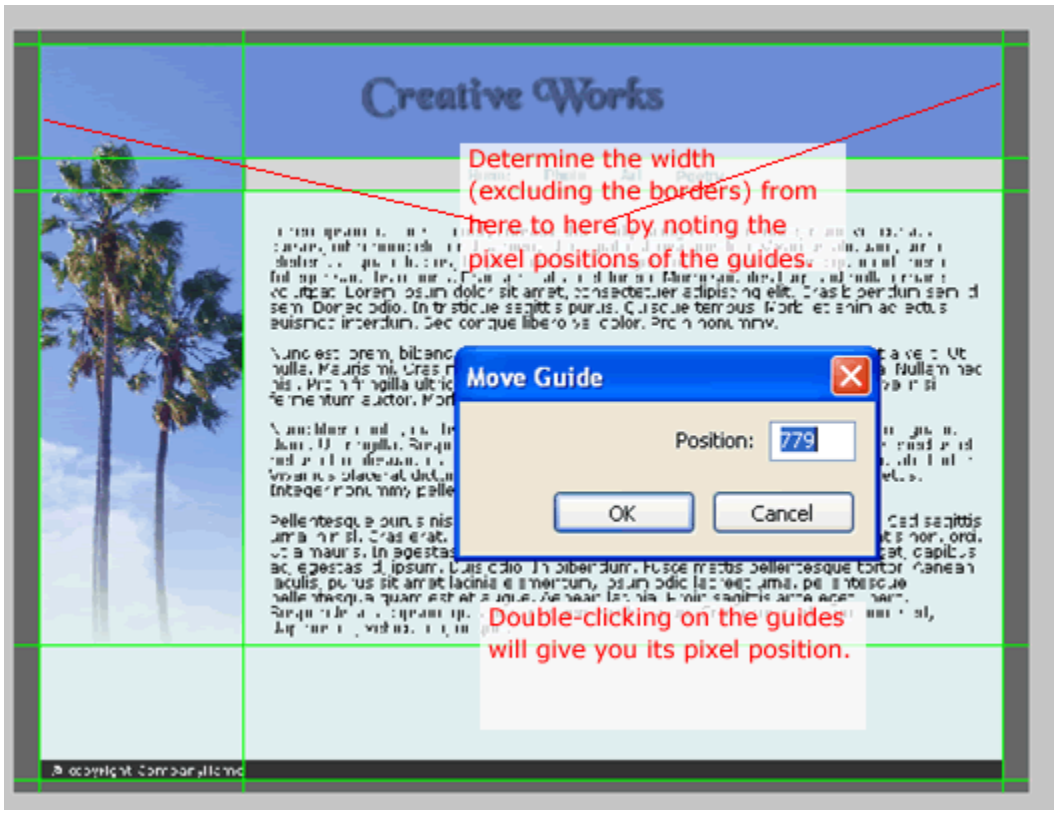
Table Layout

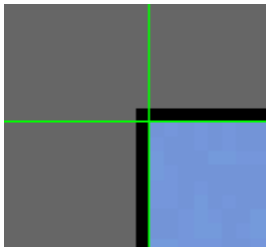
For your first webpage, we will be using a table layout. In other words, we construct a table with rows and columns of cells. Inside those cells, we will be placing images and content. In the early days of web design, all pages are laid out using this table layout method. In today's web design, many sites are no longer using this technique and are using CSS layout instead. Because both method has its pros and cons, I believe that one should know both method and use the method best suited for the individual design. I do teach the CSS layout method in the Intermediate design series. However, in the Intro Design Series, I think it is best to learn the table method first because it is easier learn and it is faster to build and you don't run into all the cross-browser compatibility problems with the CSS layout technique. When you become more comfortable with building web pages, then I would encourage you to learn the CSS technique in my Intermediate Design Series courses.

Using the Comp File

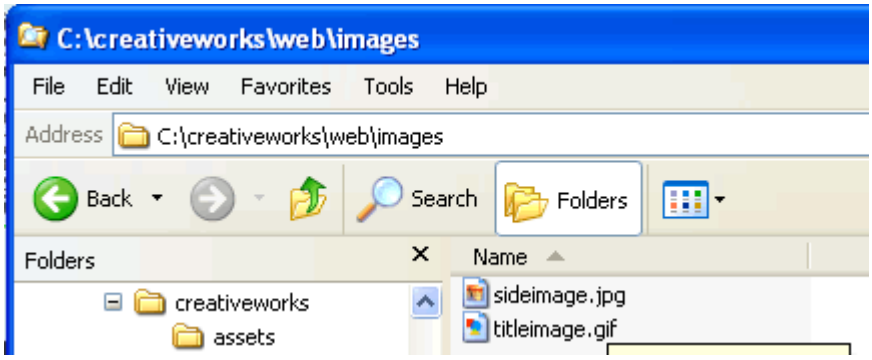
In order to build the site with the exact dimensions as the Fireworks comp, we need to open up the **creativeworks.png** file that is in the **assets** folder using Macromedia Fireworks. And in order for our images to fit together perfectly, we need to have pixel perfect accuracy.

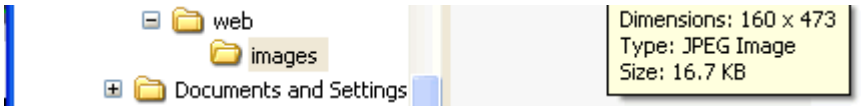
From this Fireworks file, we need to determine the width of our web page and that will determine the width of our table.




Make sure guides are set inside the borders.

With guides turned on and set to be inside the borders, we double-click on them to see that the right guide is at 779 and the left guide is at 21. This gives us an inner width (excluding the borders) to be 758.

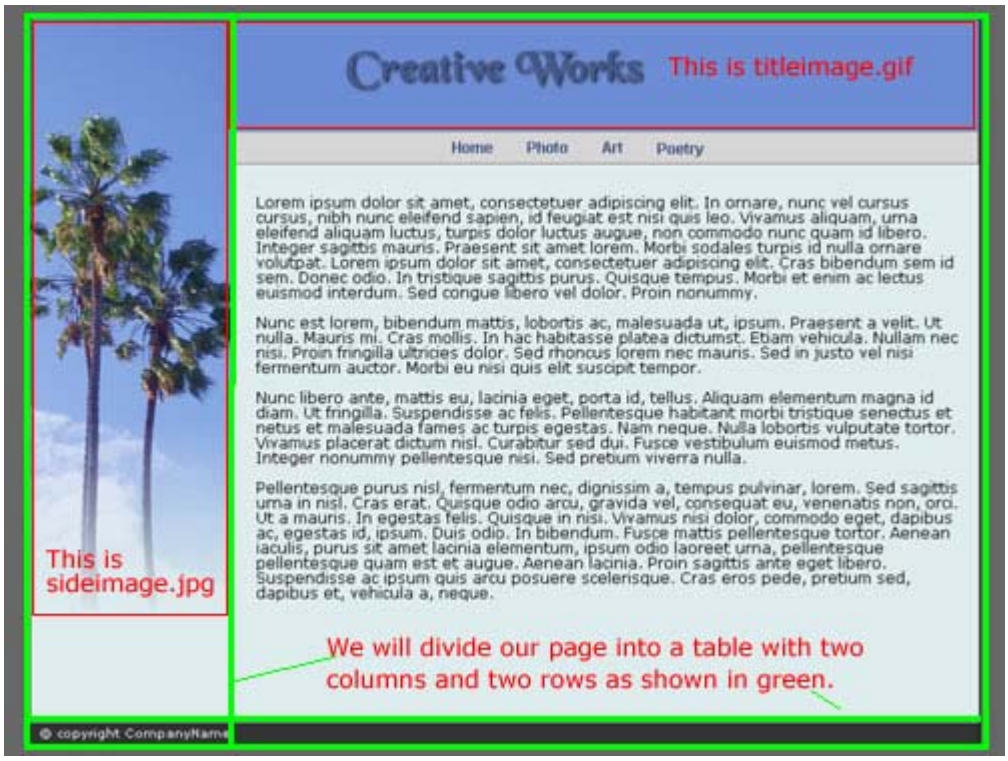




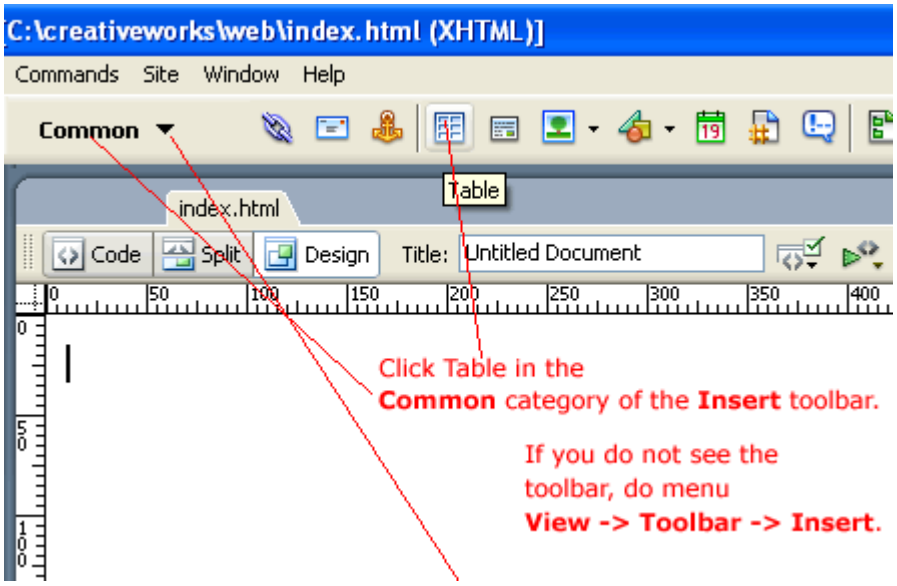
If you pause your mouse over our two image files in Window Explorer, you will see that the **sideimage.jpg** has a width of 160 and **titleimage.gif** has a width of 598. So the two images laid side-by-side will be 758 pixels wide -- exactly the width of our page. The confirms the correctly of the width of our page.

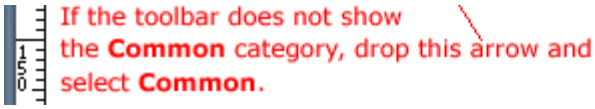
Dividing the Page

Since we want our table cells to hold our images, we divide our page into table cells as shown.

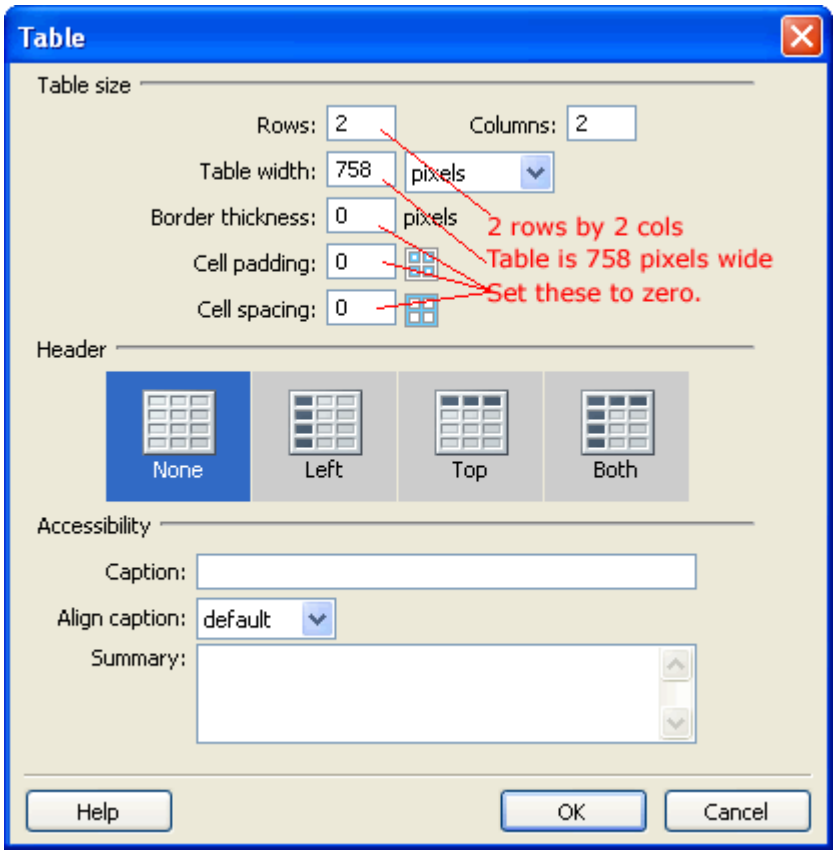


So far we have an empty **index.htm** page. Switch to the Design View and insert a table of 2 columns and 2 rows through these steps...



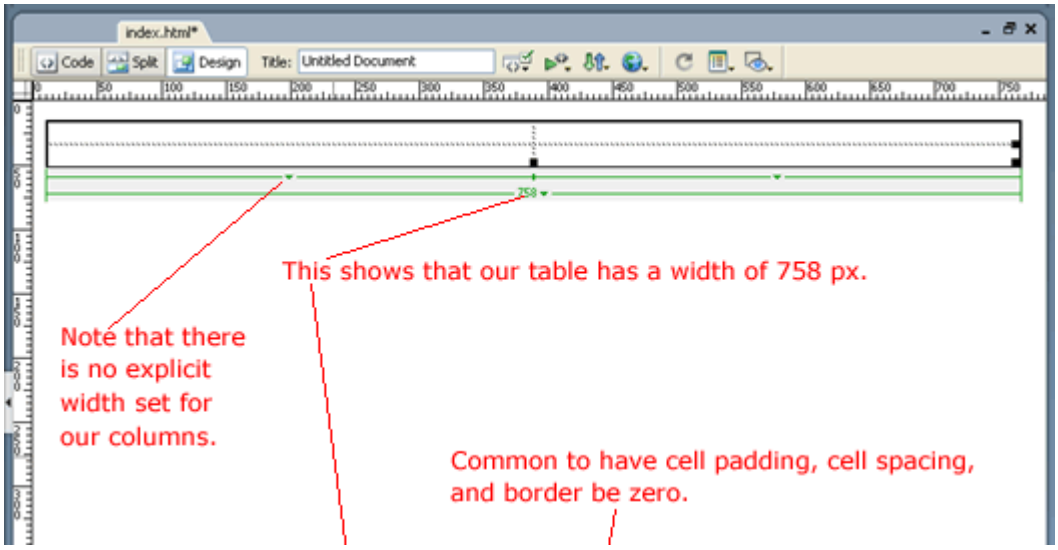


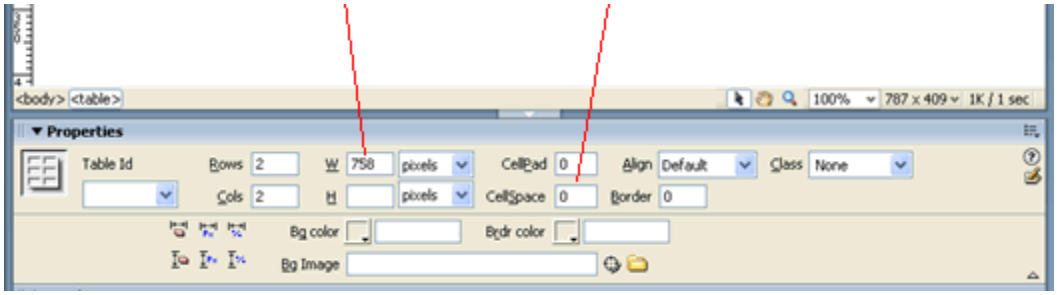
The **Table** dialog comes up...



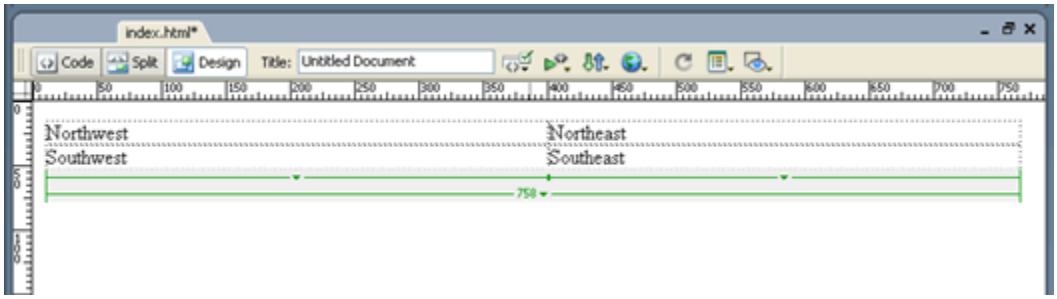
Since we want things (such as images) in our table to sit touching each other without space or gaps, it is very common to set the cell padding and cell spacing properties to zero. It is also common to set the border to zero so that you don't see the borders dividing our tables into cells.

After clicking **OK** with these settings, you will see the outline of the table in Design View as well as some of its properties in the **Properties** panel...





To learn how the HTML code looks for this setup, let us put in some filler text into each cell of our table as shown.



And switch to code view ...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-88
<title>Untitled Document</title>
</head>

<body>
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>Northwest</td>
    <td>Northeast</td>
  </tr>
  <tr>
    <td>Southwest</td>
    <td>Southeast</td>
  </tr>
</table>
</body>
</html>
```

Our table properties are set as attributes of the table tag.

<table> marks the start of our table

<tr> and </tr> marks the start and end of a table row.

Inside a table row are cell elements marked off by <td> and </td>

</table> marks the end of our table

The green HTML code represents our table which starts with the **<table>** tag and ends with the **</table>** tag. Anything in between these tags are inside our table.

Looking at the start **<table>** tag,

<table width="758" border="0" cellspacing="0" cellpadding="0">

we see that it has extra things known as attributes. For example, we see these four attributes in our table tag...

width="758"
border="0"
cellspacing="0"

cellpadding="0"

These attributes represents our table properties.

An attribute consists of the ***attribute name*** (here it is **width**), followed by an equal sign, and then followed by the ***attribute value*** in quotes (here it is **758**). According to the XHTML standard, tag names (such as **table**) and attribute names (such as **border**) should always be written in lower case letters. And attribute values should always be in quotes.

XHTML also states that one should always specify and attribute value along with an attribute name. In other words -- for those who are familiar with shorthand attributes -- XHTML says not to use them. For example, use ...

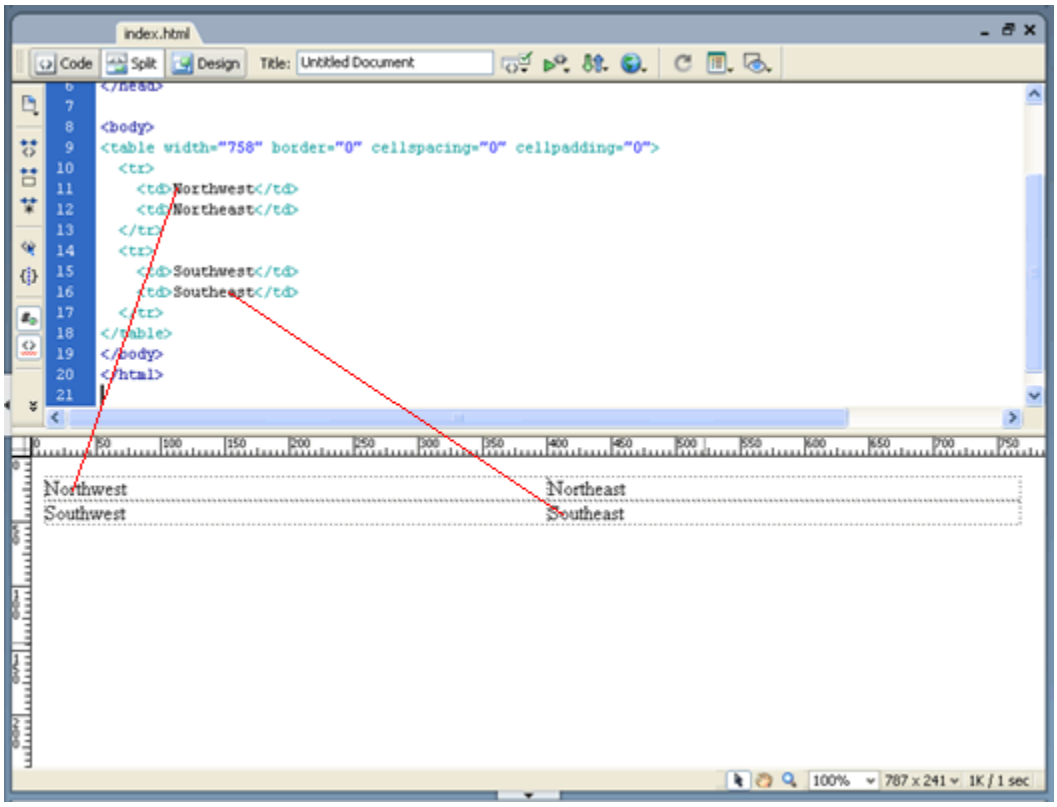
<input type="checkbox" checked="checked">

instead of

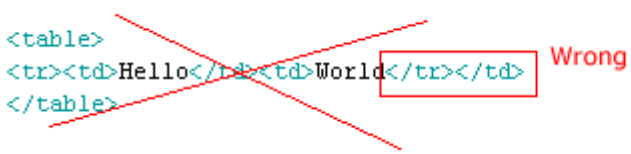
<input type="checkbox" checked>

Although the browsers will be able to render our page correctly even though we do not strictly obey the XHTML standards. You mind as well learn the correct habit of following the standards.

Inside our **<table>** and **</table>** tags, we have sets of **<tr>** and **</tr>** tags that represents table rows. Inside table rows are table cells (or table data) represented by **<td>** and **</td>**. And we see our filler text inside these table cells. By looking at the **Split View**, we can understand which **<td>**'s goes with which position in the table.

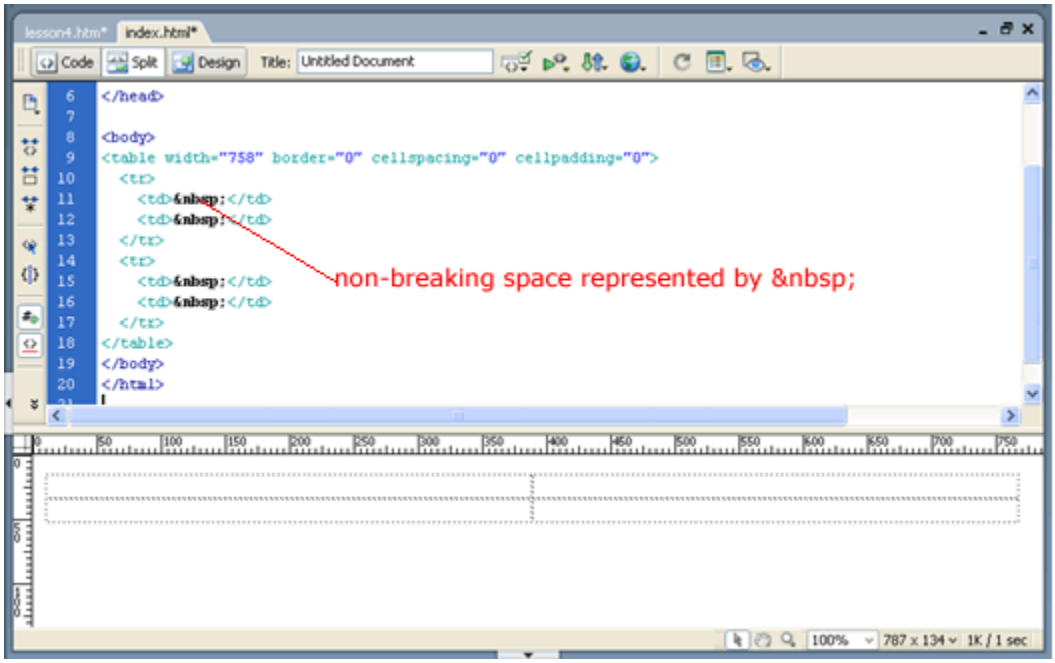


Make sure that you ***nest*** your table tags properly. As an example, for example ...



```
<table>
<tr><td>Hello</td><td>World</td></tr>
</table>
```
Correct

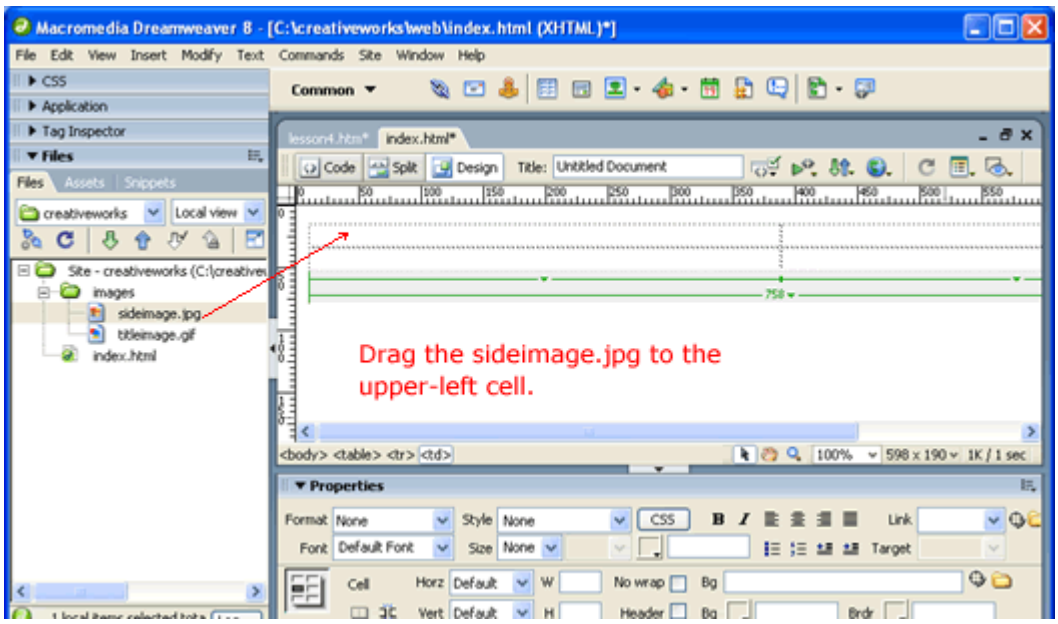
While in Split view, delete our filler text (**Northwest**, **Northeast**, **Southwest**, **Southeast**) from the bottom graphical view and see that Dreamweaver had put in ***non-breaking spaces*** (those things) in between the **<td>**'s.



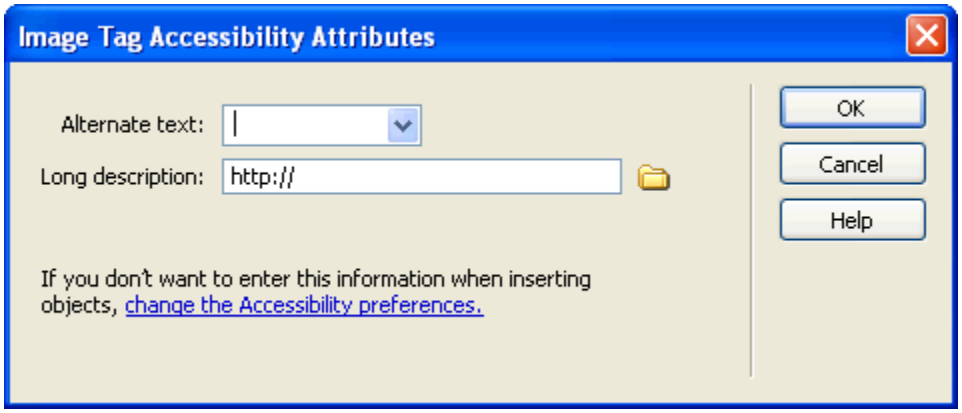
That is because browser sometimes will "***collapse***" empty **<td>**'s. Think of as just a space character to prop our table up.

Adding Our Image

Switch back to the Design view. To put our **sideimage.jpg** as the content of the northwest cell, simply drag it from the Files panel to the cell as shown.

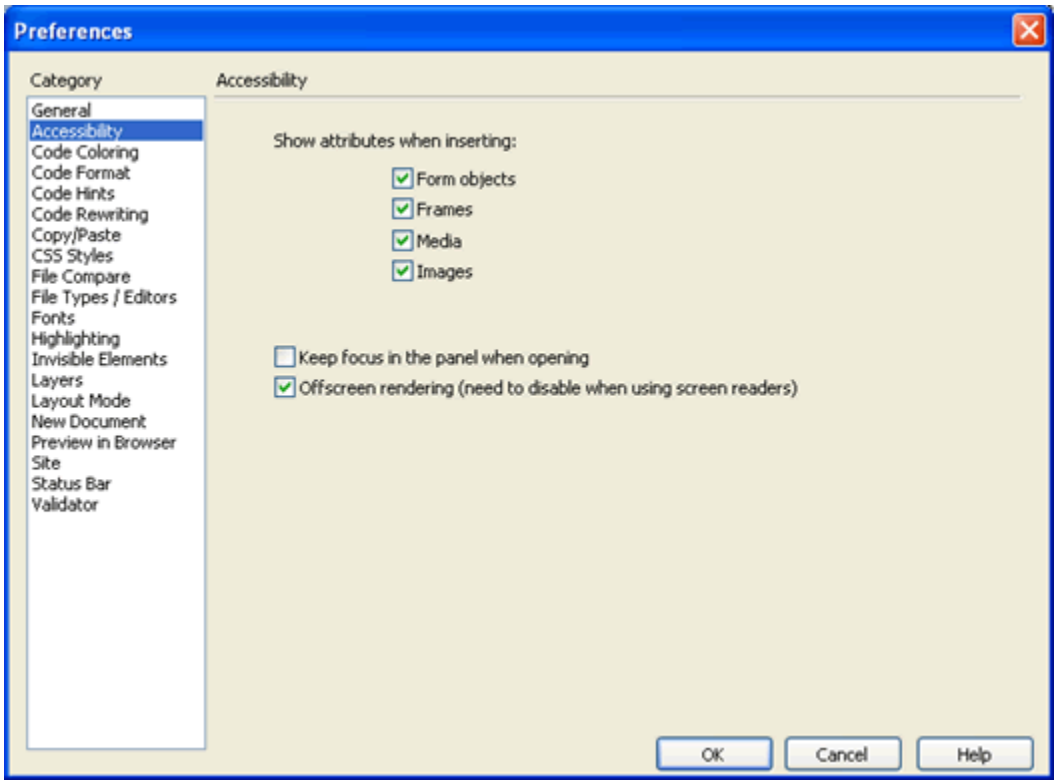


You might get the **Accessibility Attribute** dialog ...



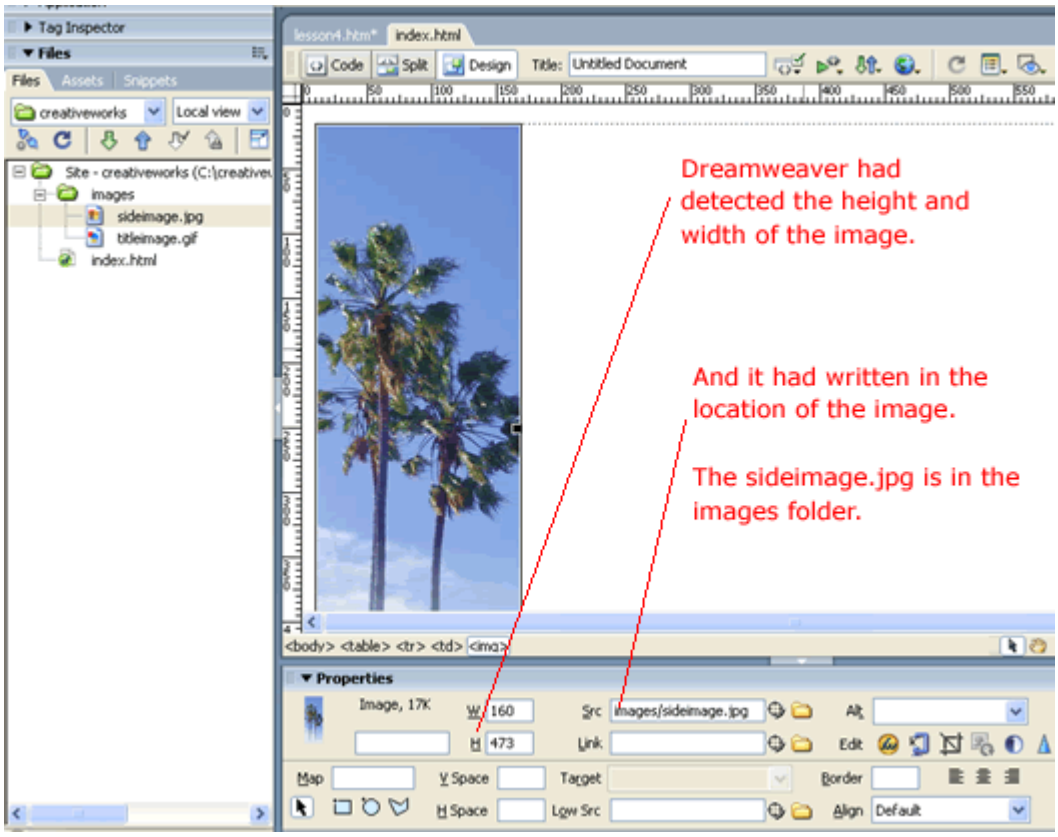
We won't worry about this now. So click **Cancel**.

I say that you might, because you will see this dialog whenever you place an image if your **Edit -> Preferences** are set as shown...

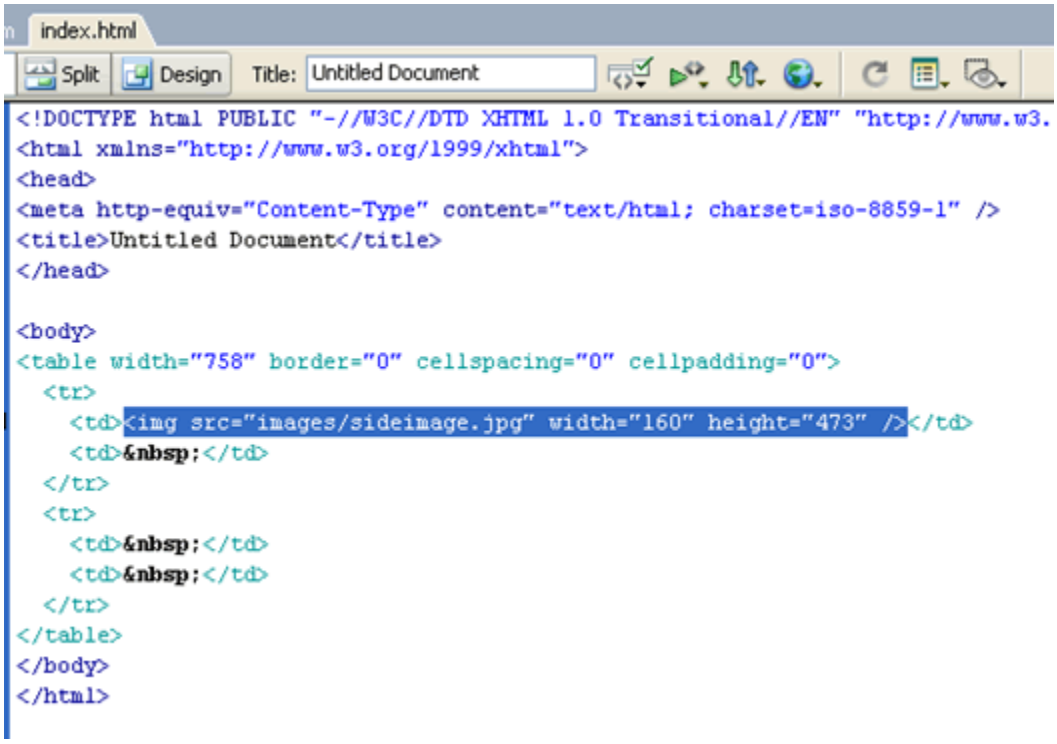


And in Dreamweaver 8 (but not in Dreamweaver MX), these are the default settings. Although we had dimissed **Accessibility Attribute** dialog this first time, we will be using it next time. So check your preference settings now and make sure that you have them set as shown above.

After dragging the image to the cell and selecting the image, you see the image's height and width already specified in the **Properties** window ...



While the image is still selected, switch to the code view. You will see that the code for the image is already highlighted by Dreamweaver ...



The image code (represented by the **<img>** tag) had replaced the non-breaking space in the **<td>** cell. Note that the image tag is said to ***"have no body"*** and that the start tag and end tag are merged into one tag as

<img />

Of course, the **<img>** tag has attributes. The **src** (stands for "source") attribute tells the browser where to get the image relative to the current html file. Since our image in within the images folder, the relative path of our image is **images/sideimage.jpg**

The **<img>** tag also specifies the **width** and **height** attribute. Although these attributes can be omitted since the browser can figure out the height and width of the image. It is good practice to always specify explicitly the height and width in the **<img>** tag. This way, the browser can easily know amount of space to allocate in page layout without having to wait for the image to download.

Specifying Column Width

We want the width of the first column to be exactly as wide as the side image (which is 160px). So we specify the width of the first column by adding the **width** attribute to the first **<td>** cell in the first row as shown.

```
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="160"><img src="images/sideimage.jpg" width="160" height="473" /></td>
    <td width="598"> </td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
</table>
```

width of the first column

width of the second column

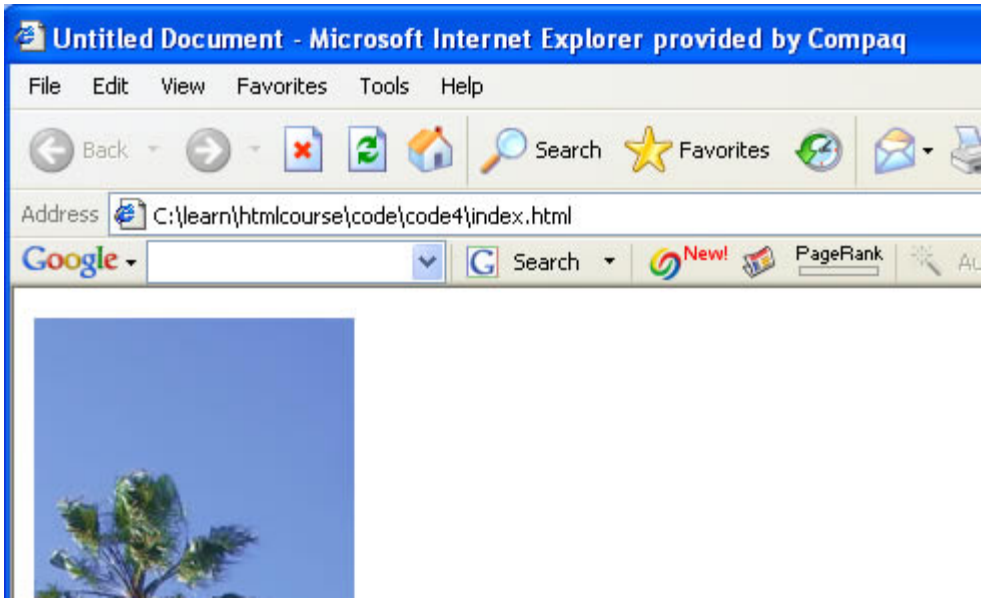
Similarly, we specify the width of the second column in the second **<td>**.

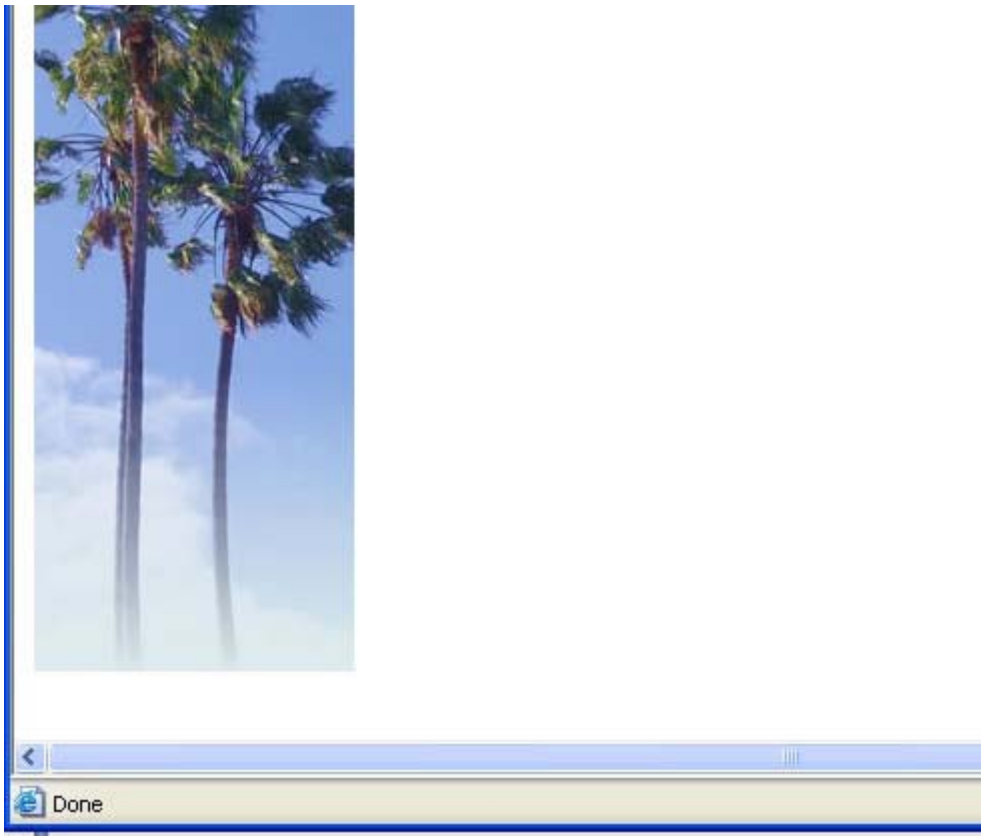
View Live Code

Download: **code4.zip**
Click link and save file to local disk.

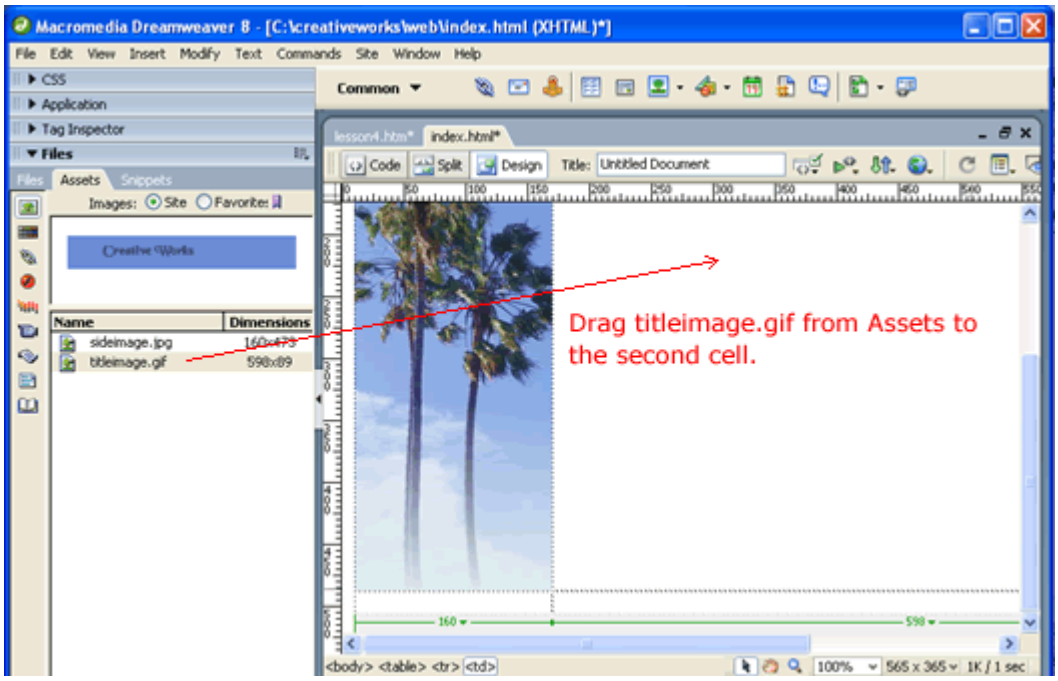
Adding Title Image

So far, we had placed one image...

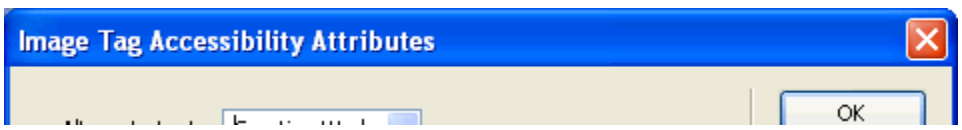


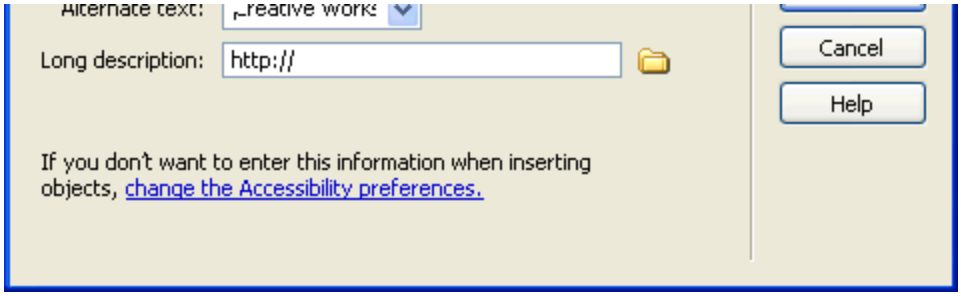


In the design view of **index.html**, we will add our second image, the **titleimage.gif**. This time, we will drag the **titleimage.gif** from the **Assets** panel as shown.

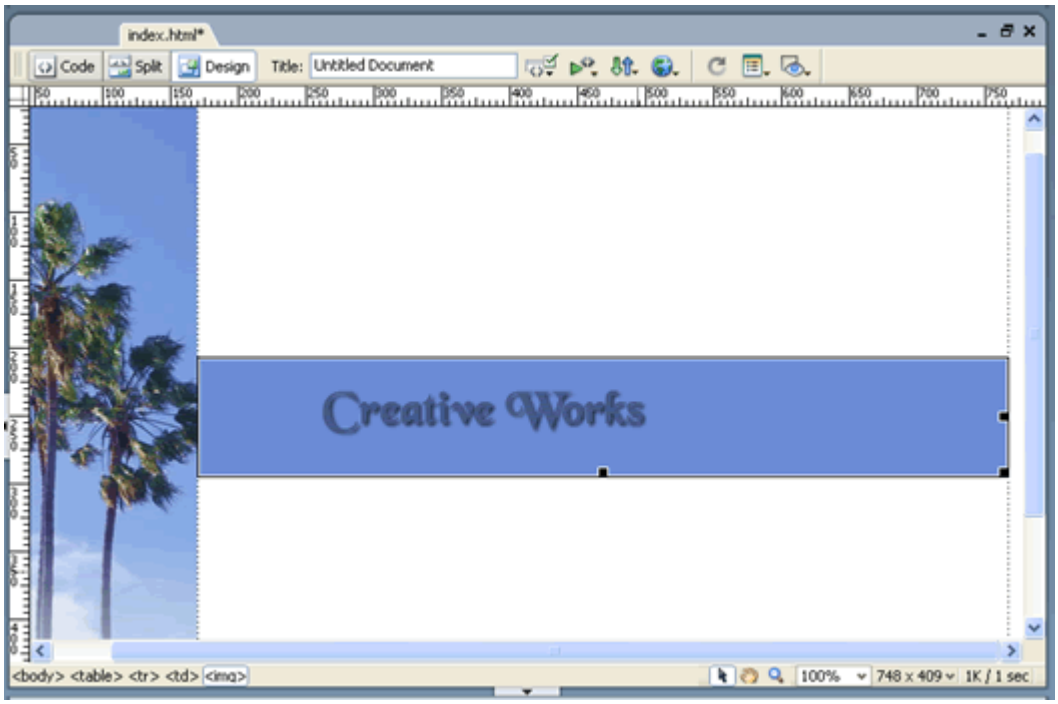


When the **Accessibility Attributes** dialog shows up, enter **Creative Works** for the **Alternate text**.





Click **Okay** and you get ...



Oops! By default, the image is centered vertically within the table cell. We'll fix that shortly. But first get to the code view to see what difference adding the **Alternate text** made.

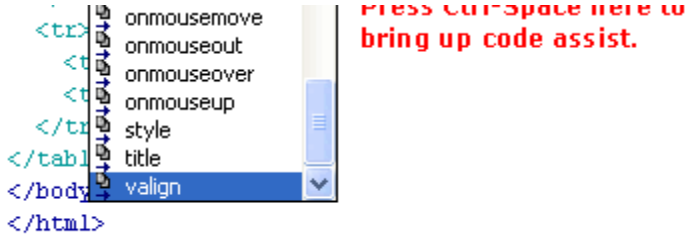
```
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="160"><img src="images/sideimage.jpg" width="160" height="473" /></td>
    <td width="598"><img src="images/titleimage.gif" alt="Creative Works" width="598" height="89" /></td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
</table>
```

The alt attribute was added.

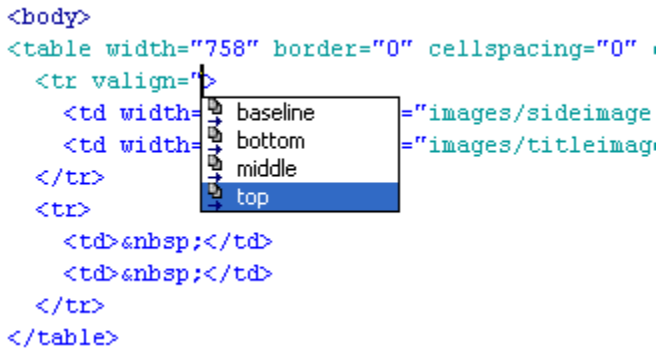
The **alt** attribute provides alternate text for the images in cases where the user had turned off images, is using a text-based browser, is using a screen reader due to visual impairement, or for search engines to read.

We want everything on the table row to be top-aligned. So we add an attribute to the **<tr>** tag. Type a space after "**tr**" as if you were about to type another attribute. The press **Ctrl-Space**

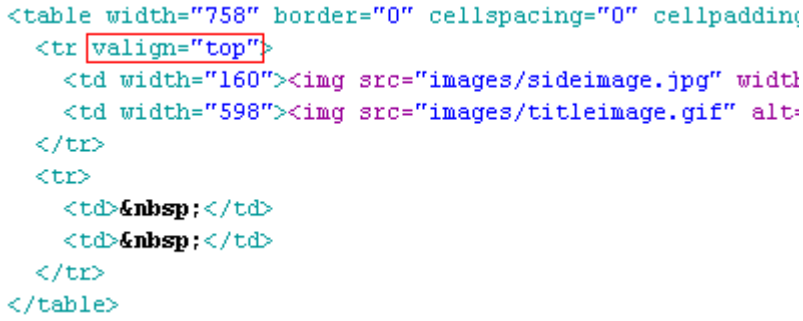
```
<body>
<table width="758" border="0" cellspacing="0" cellpaddi
  <tr >
    <t  onkeypress                src="images/sideimage.jpg" wid
    <t  onkeyup                   src="images/titleimage.gif" al
  </tr
                onmousedown
```



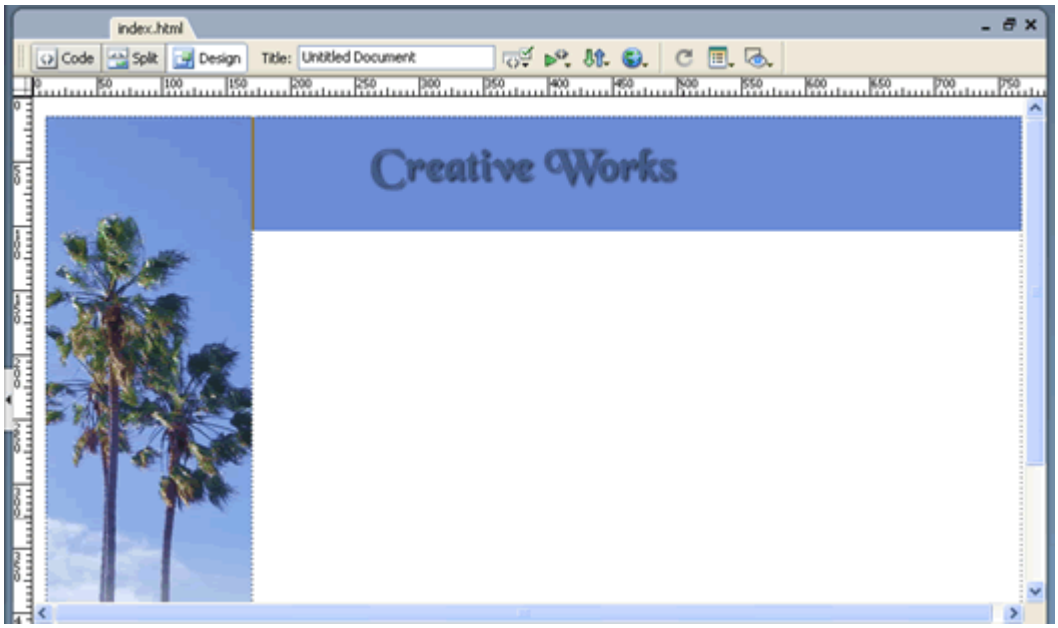
and select **valign** (for vertical alignment)...



Wow. Dreamweaver provides us with all the possible values for the **valign** attribute. Choice is now obvious, we want **top**.



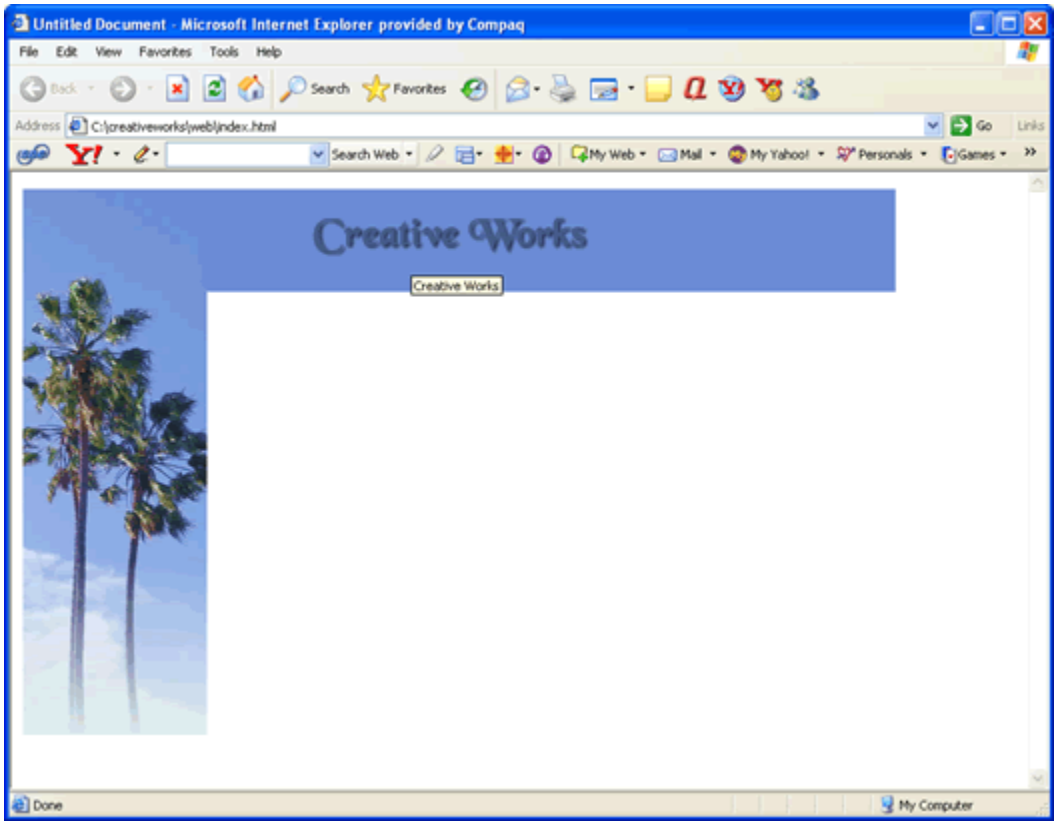
Switching back to design view, we see that we got what we want...



Save file.

Testing

Now that we had added two images, preview the page in IE and see that the tooltip appears as you hover your mouse over the title graphics. The text of that tooltip came from the **alt** attribute. As you will notice, we do not get the tooltip on the side image since we did not put in an alt attribute.



This tooltip effect is an IE specific feature. If you try this in Firefox, you will not get the tooltip. However, if you add in the **title** attribute to our **<img>** tag....

```
<body>
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160"><img src="images/sideimage.jpg" width="160" height="423" /></td>
    <td width="598"><img src="images/titleimage.gif" alt="Creative Works" title="Creative Works" width
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
```

You will get the tooltip in Firefox as well.

Adding the Title

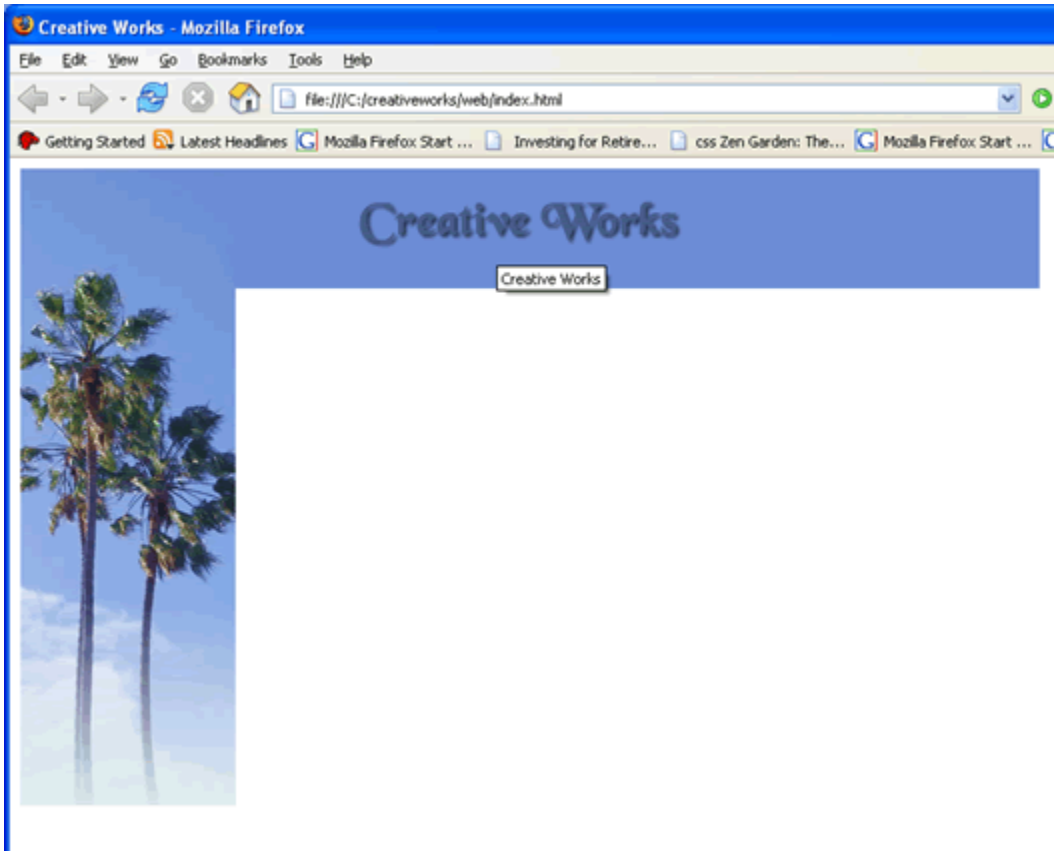
Another thing we noticed when we preview our page in the browser is that the browser title bar still says "Untitled Document".

That is because we had not given our page a title. Add in the title now...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
```

```
<html xmlns= http://www.w3.org/1999/xhtml >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
</head>
```

And we are all set.



View Live Code

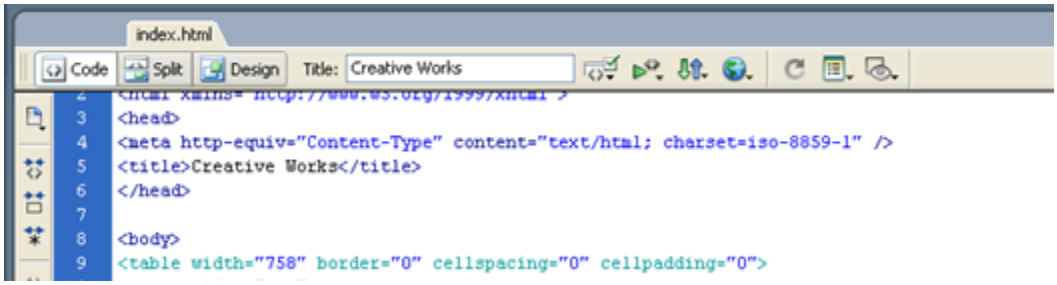
Download: **code5.zip**

Code Collapse and Word Wrap

As you might have noticed, the line containing the **<img>** tag for our title image is quite long, especially with the **alt** and **title** attributes added to it. This is a good time to learn about how to use Dreamweaver's 8 **Code Collapse** and **Word Wrap** features to deal with long lines. See tutorial **here**.

Adding Content

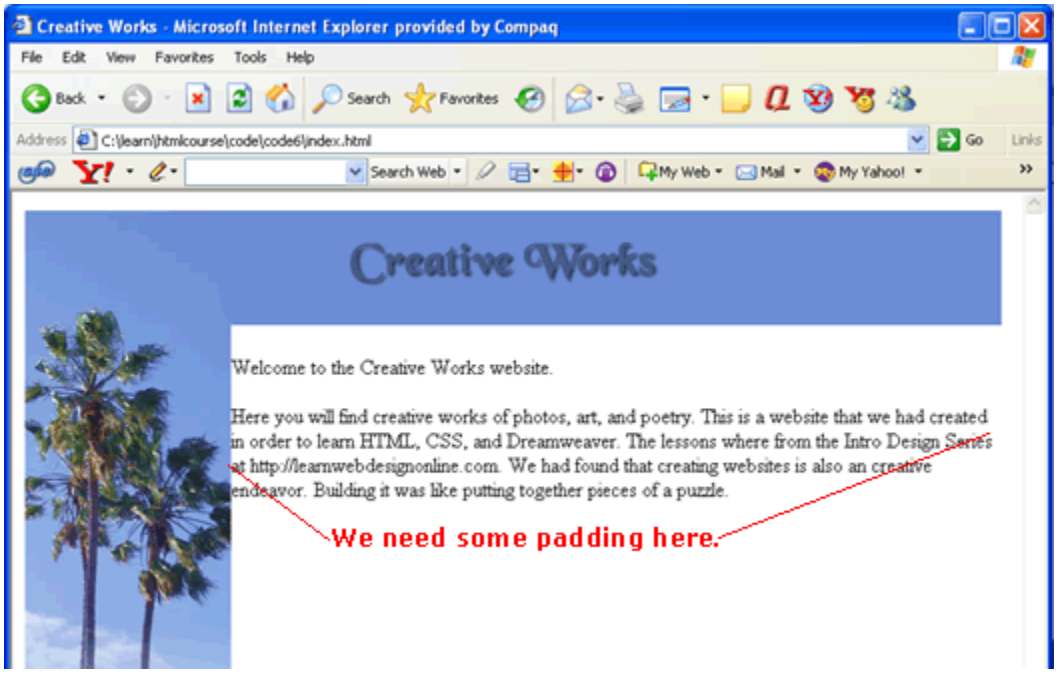
Open **index.html** in code view. In the same table cell as our title image, add in two paragraphs of text as shown...



```
<tr valign="top">
  <td width="160"><img src="images/sideimage.jpg" width="160" height="473" /></td>
  <td width="598"><img src="im... />

  <p>Welcome to the Creative Works website.</p>

  <p>
  Here you will find creative works of photos, art, and poetry.
  This is a website that we had created in order to learn HTML, CSS, and Dreamweaver.
  The lessons were from the Intro Design Series at http://learnwebdesignonline.com.
  We had found that creating websites is also an creative endeavor.
  Building it was like putting together pieces of a puzzle.
  </p>

  </td>
</tr>
<tr>
  <td> </td>
```

Add two paragraphs of text in the same <td> as our second image.

Previewing in the browser we see that we need some padding...



So we wrap our content inside a div and give it a class attribute of **pagecontent**...

```
<body>
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160"><img src="images/sideimage.jpg" width="160" height="473" /></td>
    <td width="598"><img src="im... />

    <div class="pagecontent">
    <p>Welcome to the Creative Works website.</p>

    <p>
    Here you will find creative works of photos, art, and poetry.
    This is a website that we had created in order to learn HTML, CSS, and Dreamweaver.
    The lessons were from the Intro Design Series at http://learnwebdesignonline.com.
    We had found that creating websites is also an creative endeavor.
    Building it was like putting together pieces of a puzzle.
    </p>
    </div>
```

Wrap our content inside a div.

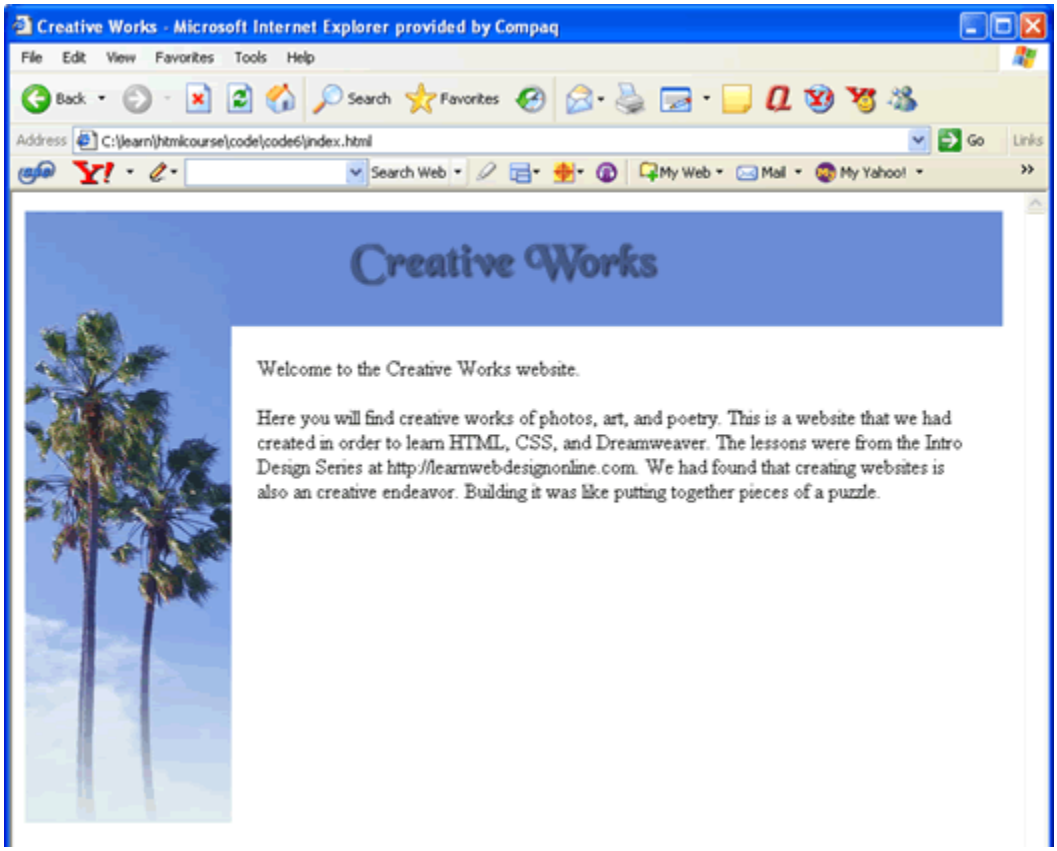
Give it class name of "pagecontent" so that we can attach styles to it.

Now we can attach styles to that div by writing the following rule...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
<style type="text/css">
.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}
</style>
</head>

<body>
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160"><img src="images/sideimage.jpg" width="160" height="4
    <td width="598"><img src="im..." />

    <div class="pagecontent">
    <p>Welcome to the Creative Works website.</p>
```

We add an internal style sheet with a CSS rule that will stylize this div.

The rule essentially says to give the div 20 pixel of left and right padding. And the page looks much better...



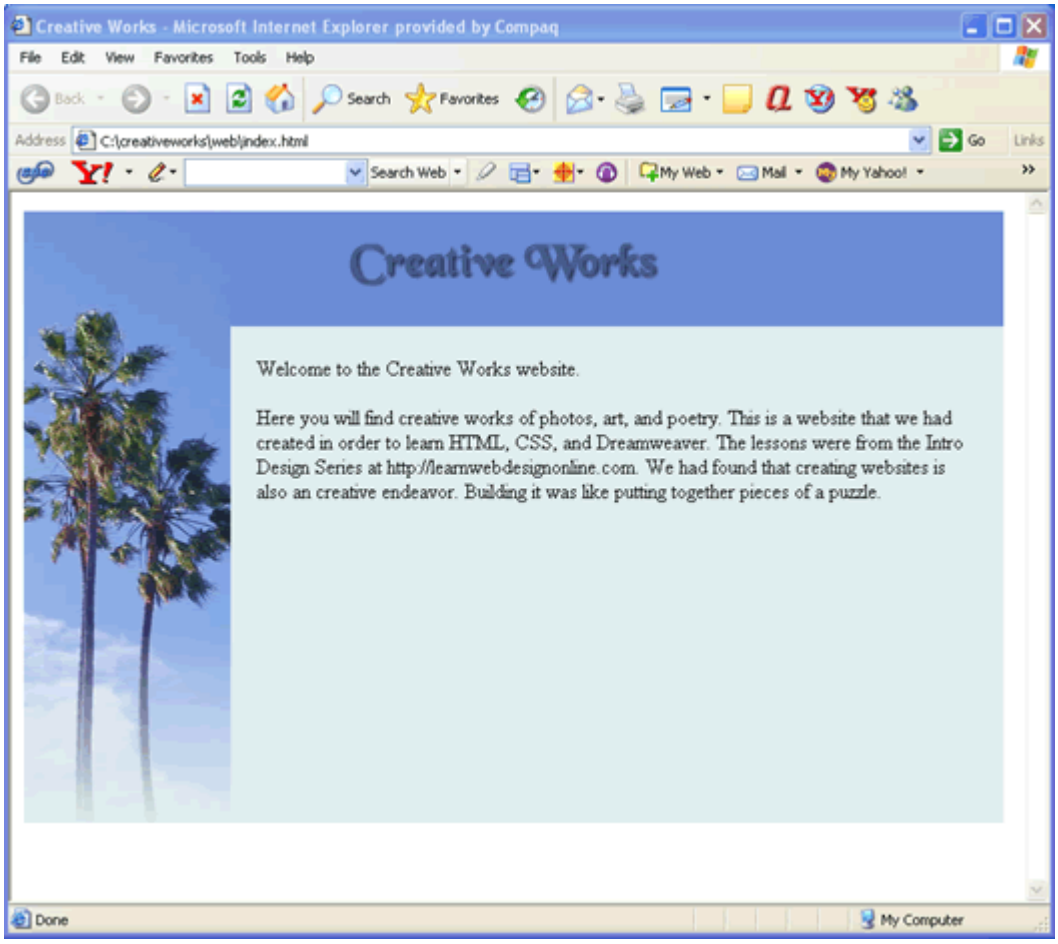
Except we need the table cell to have a light blue background as in the design comp. So add the **bgcolor** (background color) attribute to the **<td>**. Add it to both cells.

```
<style type="text/css">
```

```
<style type="text/css">
.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}
</style>
</head>

<body>
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160" bgcolor="#DFEDEE"><img src="images/sideimage.jp(
    <td width="598" bgcolor="#DFEDEE"><img src="im..." />

    <div class="pagecontent">
    <p>Welcome to the Creative Works website.</p>
```

To make the two cells have a light-blue background

Now you have a basic web page ...



View live code

Download: **code6.zip**

What's Next?

As an exercise to review what you have learned in the previous lessons, try setting the background color of the page and changing the font.

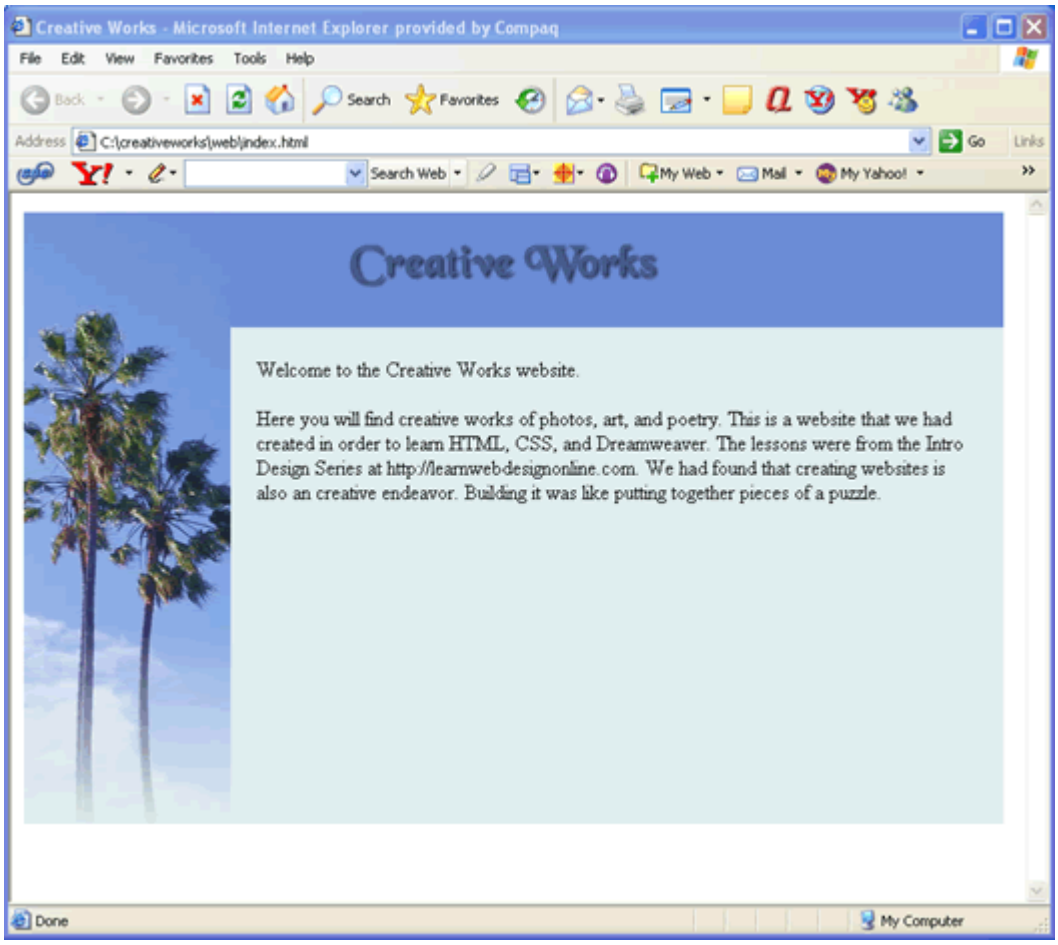
In the next set of lessons titled "**Introductory CSS**", we will continue to build upon this page to add in a navigation bar and a footer and will eventually build out a four page site.

Introductory CSS Lessons

For these series of lessons, we will learn the basic concepts of CSS (Cascading Style Sheets) as we complete our website project. We will be using Dreamweaver 8 which has very good CSS support.

Introduction to CSS

In the **HTML/Dreamweaver lessons**, we had completed a basic web page that looks like ...

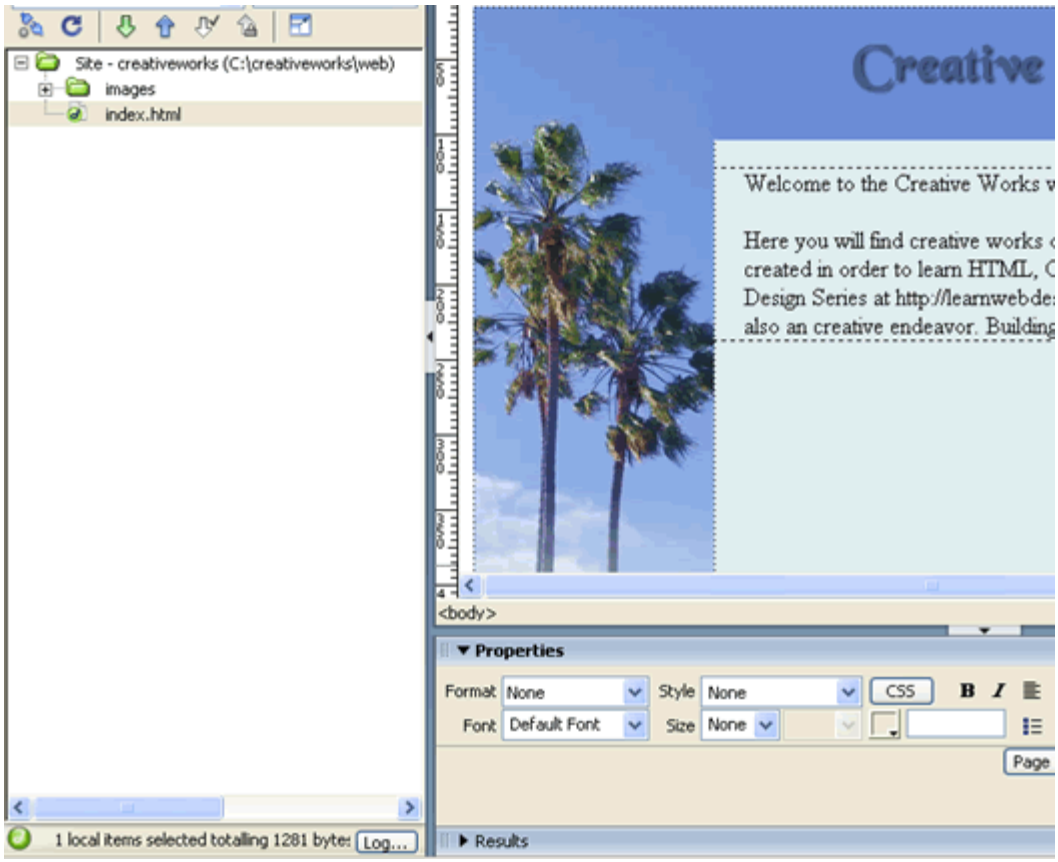


View live code

Download: **code1.zip**

To download the source files that we have so far, click on link to open up in your browser and save file to local disk. Then extract its contents into the folder **C:\creativeworks\web**. Then setup a Dreamweaver site called **creativeworks** that points to that directory we had done in the **HTML/Dreamweaver lessons**.

Open up the **index.html** file in Dreamweaver's Design View and you are ready to go. It should look something like this...



Browsers

In this course, you will need at least two browsers. The first browser you need is Internet Explorer (IE) because it does have the highest market share and is what most users have. So we need to make our pages look good on it. Remember that not all browser will display an HTML page the same way. Because IE sometime fails to follow the CSS Specifications, CSS designed pages will often work in one browser such as Firefox, but not in IE. CSS stands for Cascading Style Sheet and you will be learning a little of it shortly.

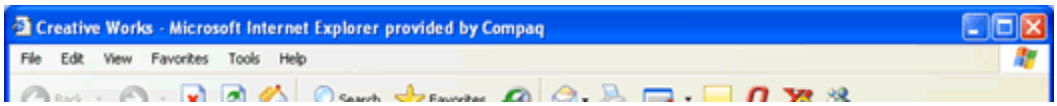
The second browser that you need for this course is Firefox, because it is one of the browsers that does follow the CSS specification quite closely and is rapidly gaining market share. If you do not have Firefox, it is a **free download**.

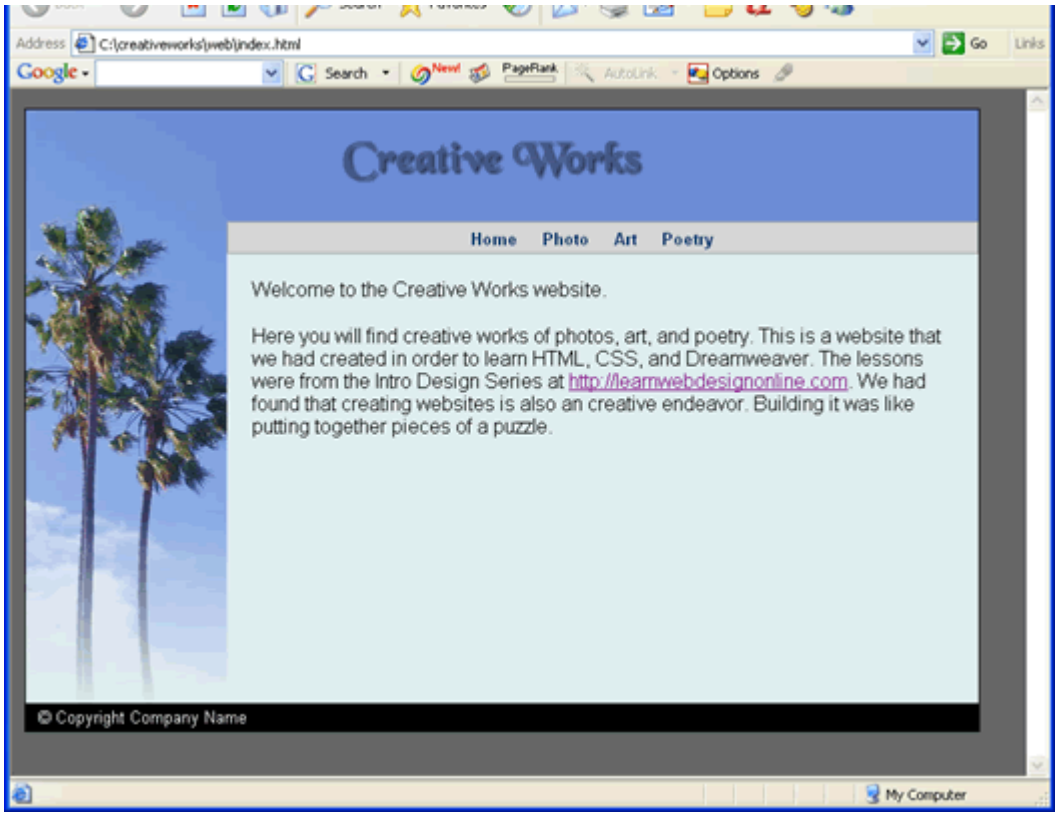
With that said, a technique from going crazy with browser bugs is to test often in both browsers. Write a little code and then test in both. Then write some more code and then test in both. Continue as such. Do not write a whole page of code in between testing. Because surely you will eventually come across a bug where it works in one browser but not the other. And you will not know which piece of code is causing the problem. If you test after writing a small amount of code, you will know exactly where the problem code is. I would test in Firefox first and then test in Internet Explorer (IE).

Next Steps

Our next steps are to build upon this page to add in a navigation bar and a footer and will eventually build out a four page site. Along the way, you will learn the basic CSS concepts as we stylize and adust the font, positioning of elements, and implent link hover changes. Lastly, we will convert this table-based design into a CSS two-column layout design that is horizontally centered across the browser.

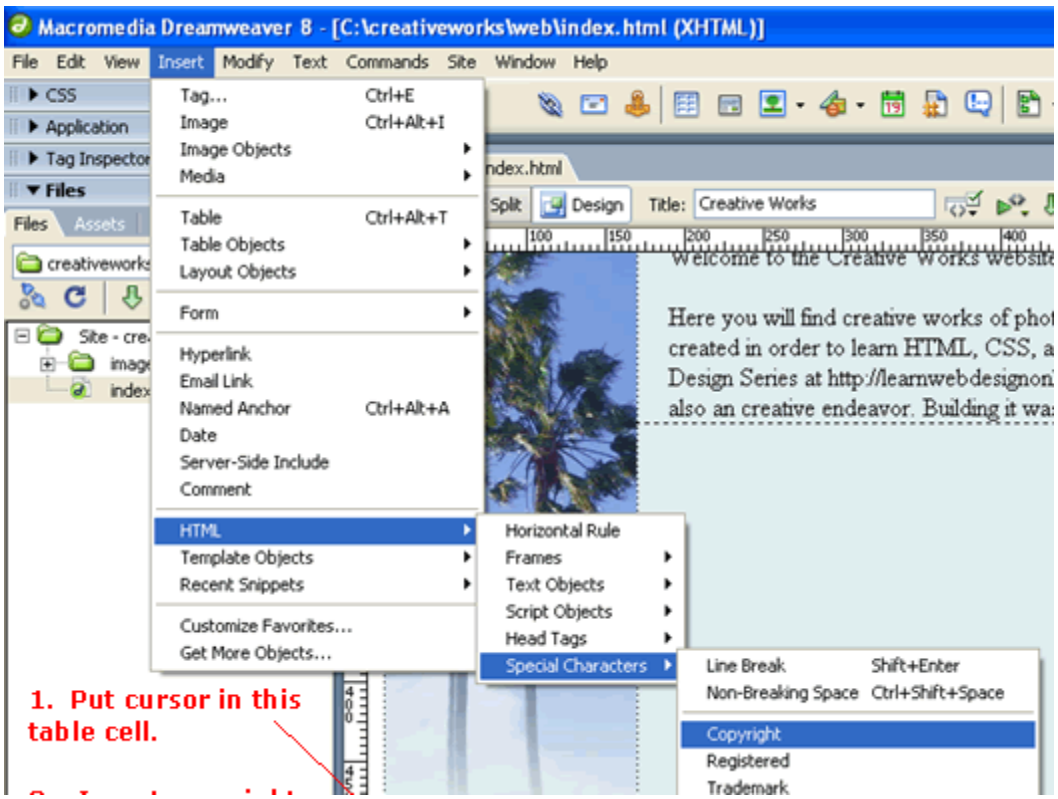
Here is the picture of the final product that we will be creating...

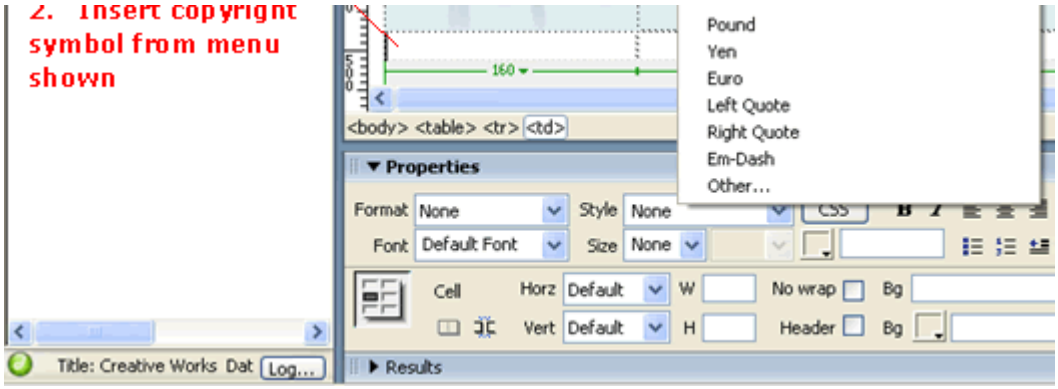


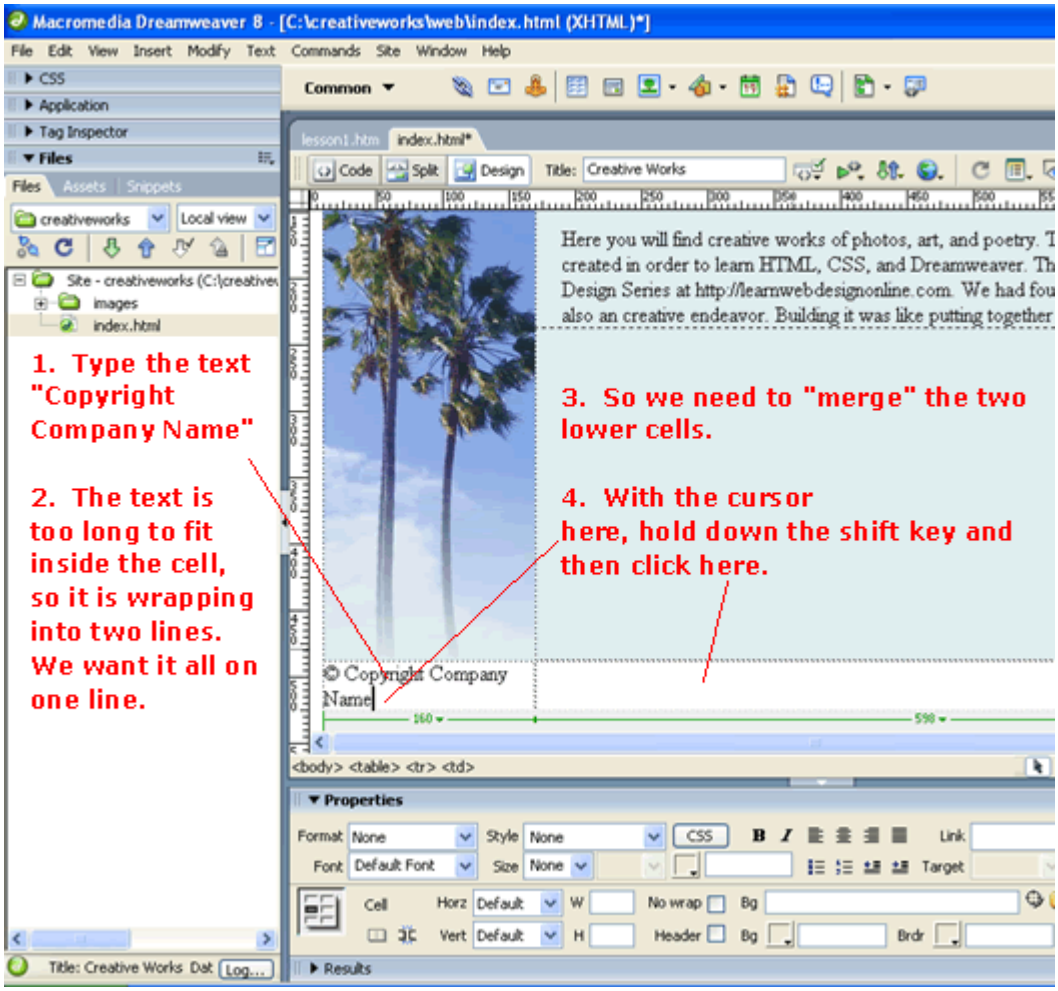


Adding the Footer

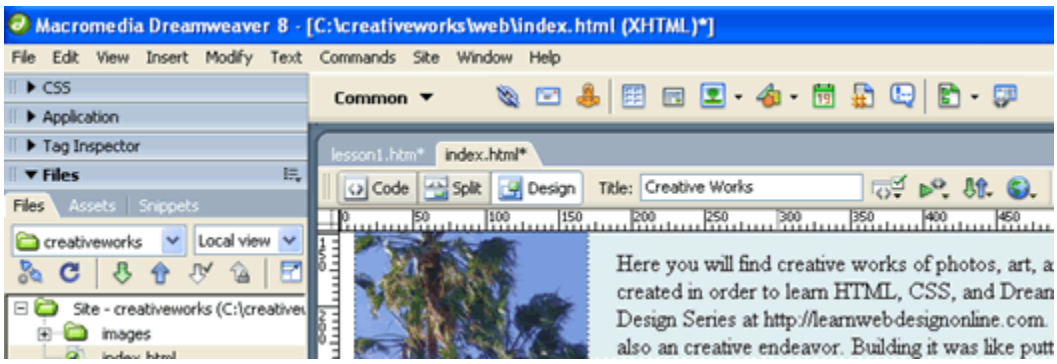
The next thing that we'll do is to add a the black footer seen above. Open up index.html in the Design View. In the lower left hand table cell, we want to insert the copyright symbol...

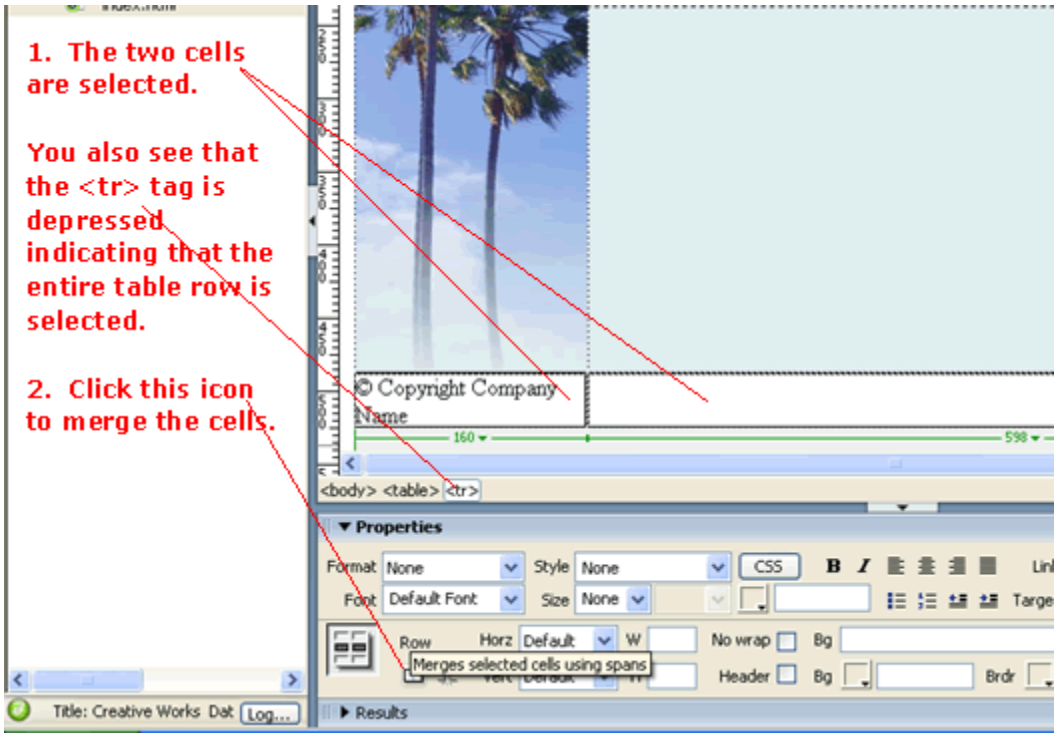




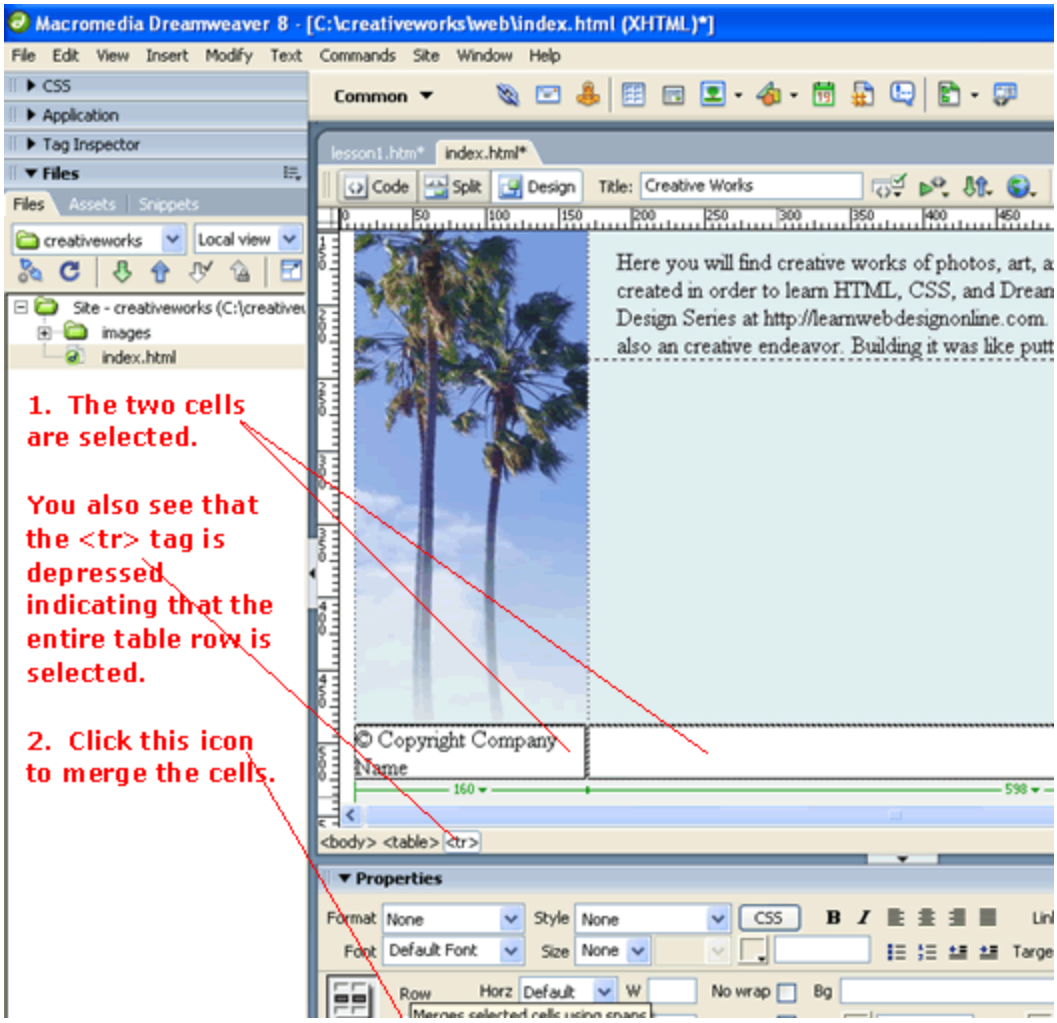


So now the two lower cells are selected as shown below.

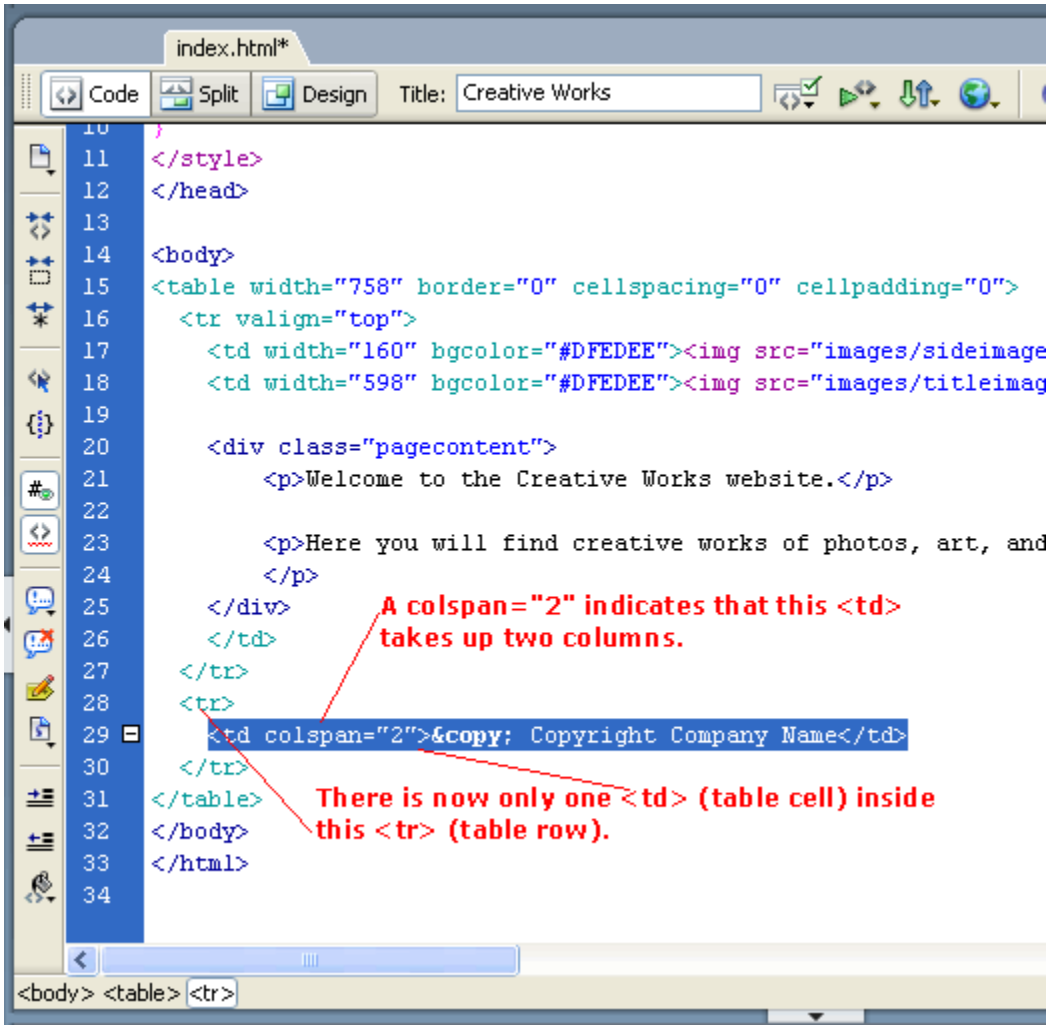




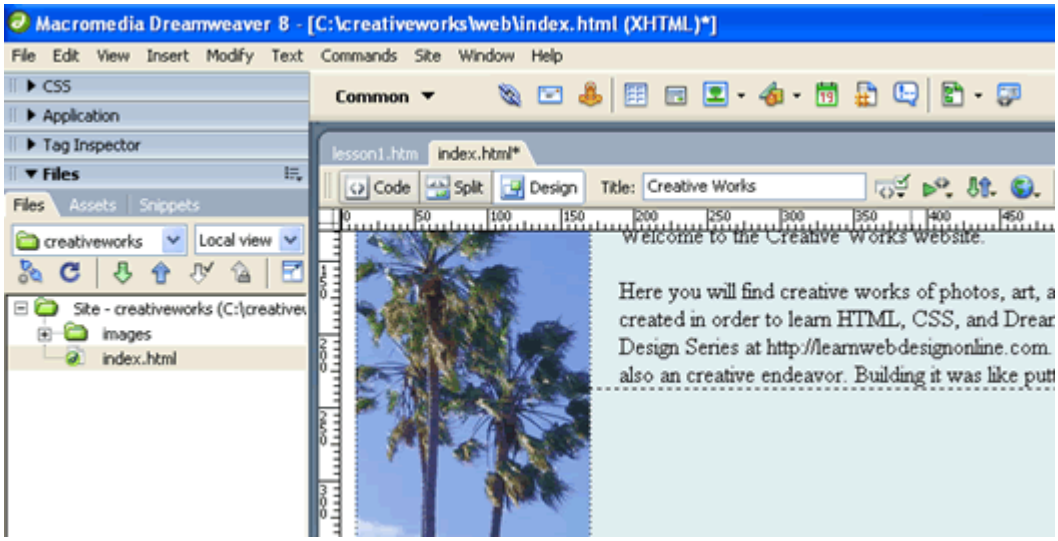
Click the "merge cell" button and the two cells become one as shown below...

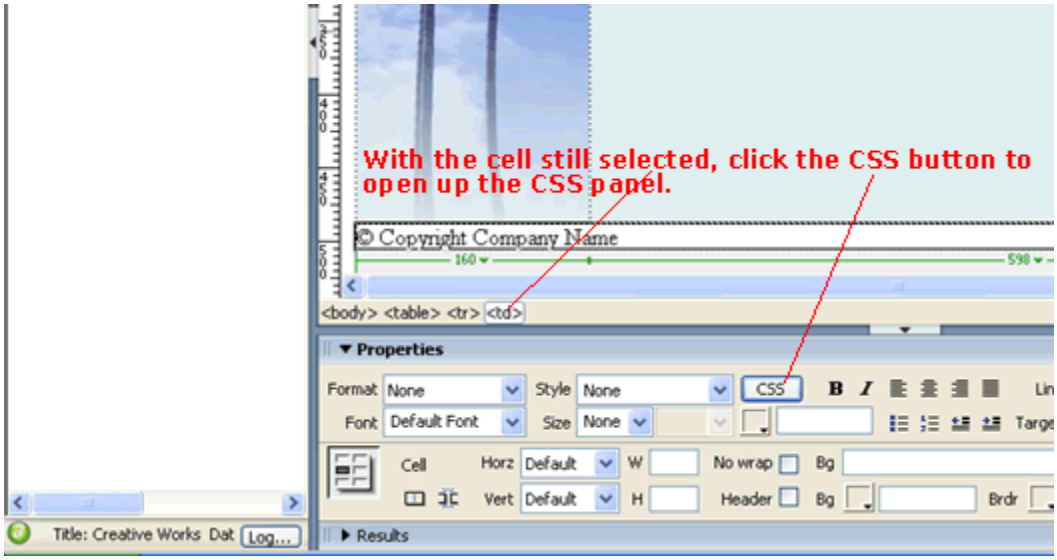


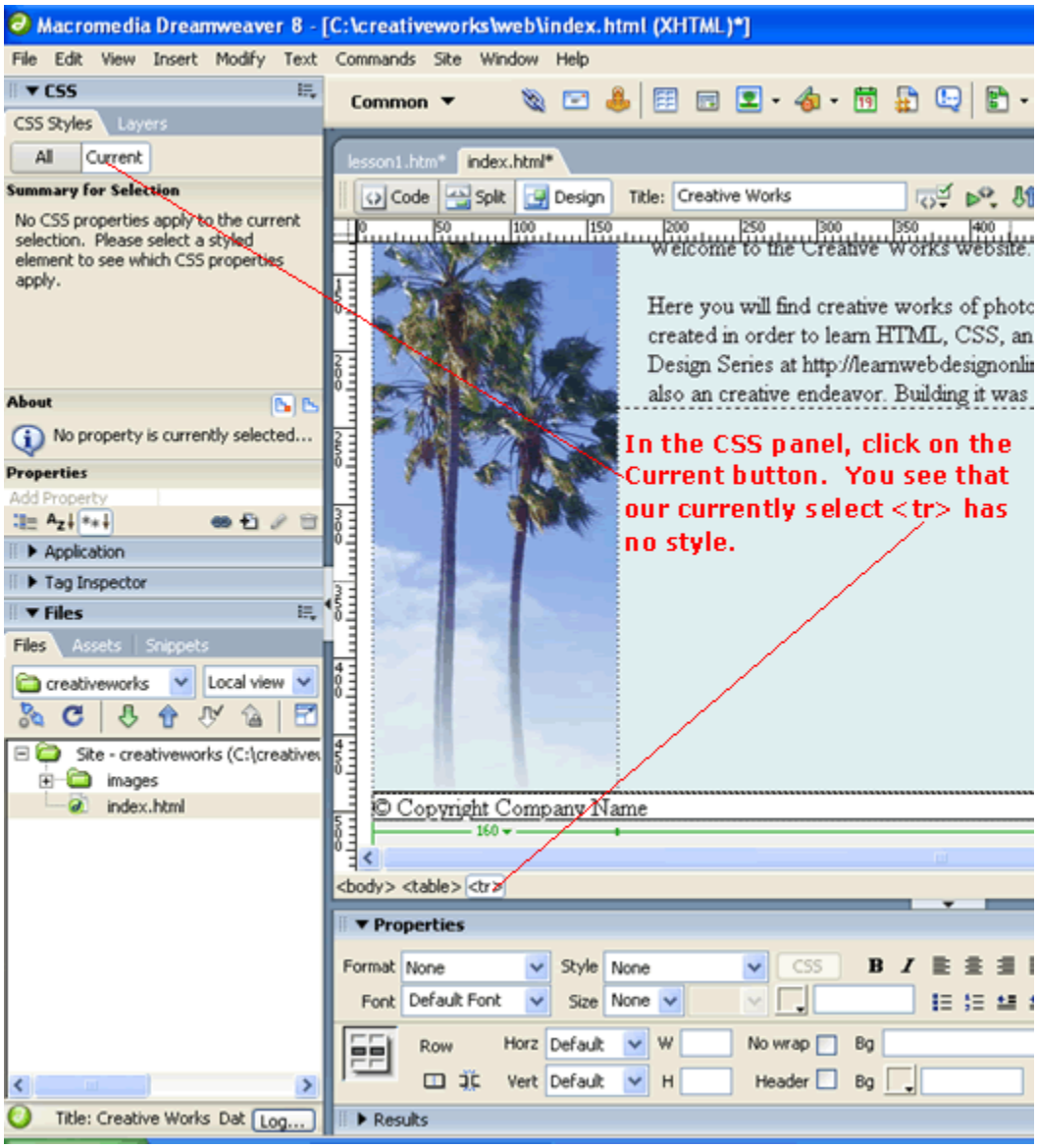
Switching to code view, we can study the HTML code that made this happen...



We want to have the footer be white text on a dark background. Switch back to the design view so that we can stylize the footer...

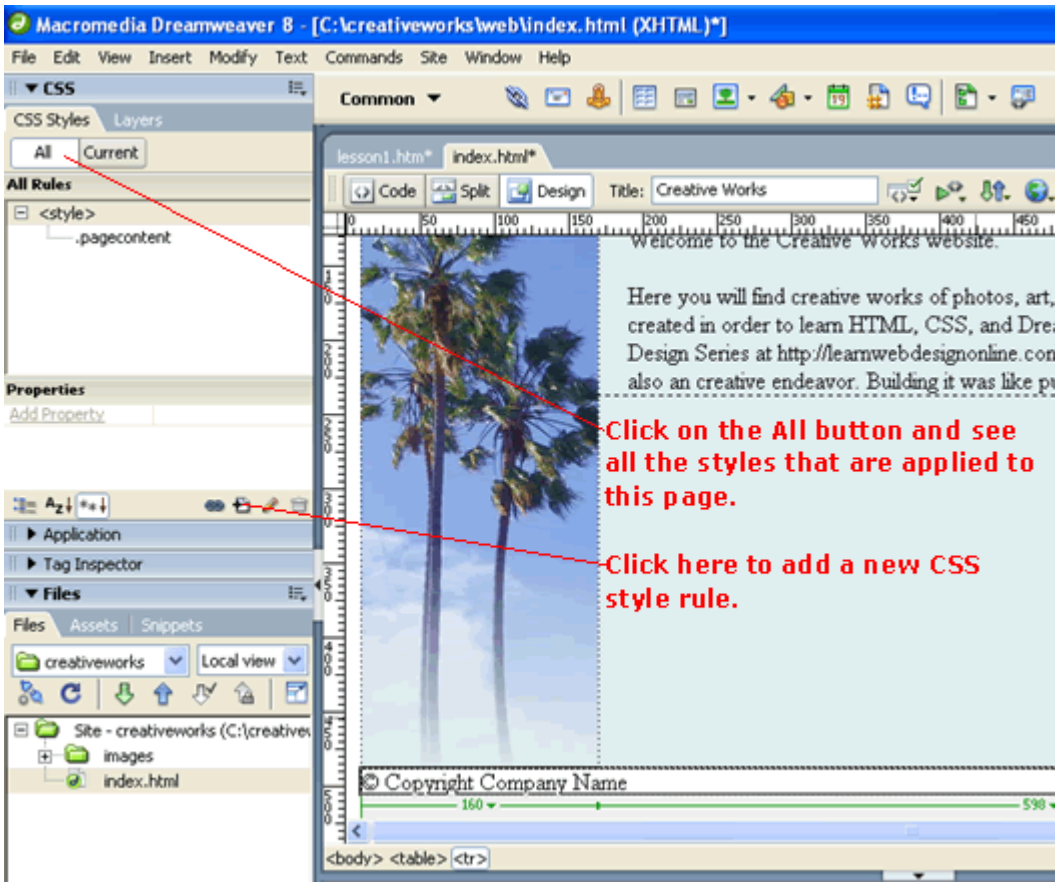




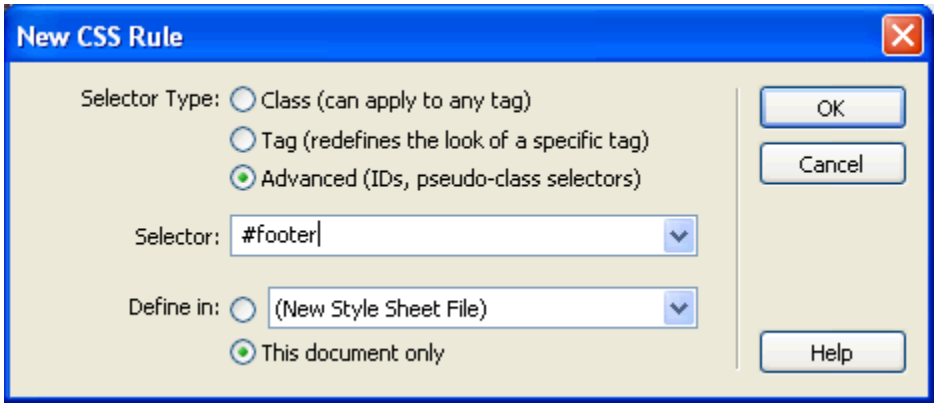


When you click the "All" button of the CSS panel, you will see all the CSS rules for the page. See below picture. If you

recall, the we had created the **pagecontent** class in the HTML/Dreamweaver course. The dot in front indicates that this is a ***class selector*** rule.

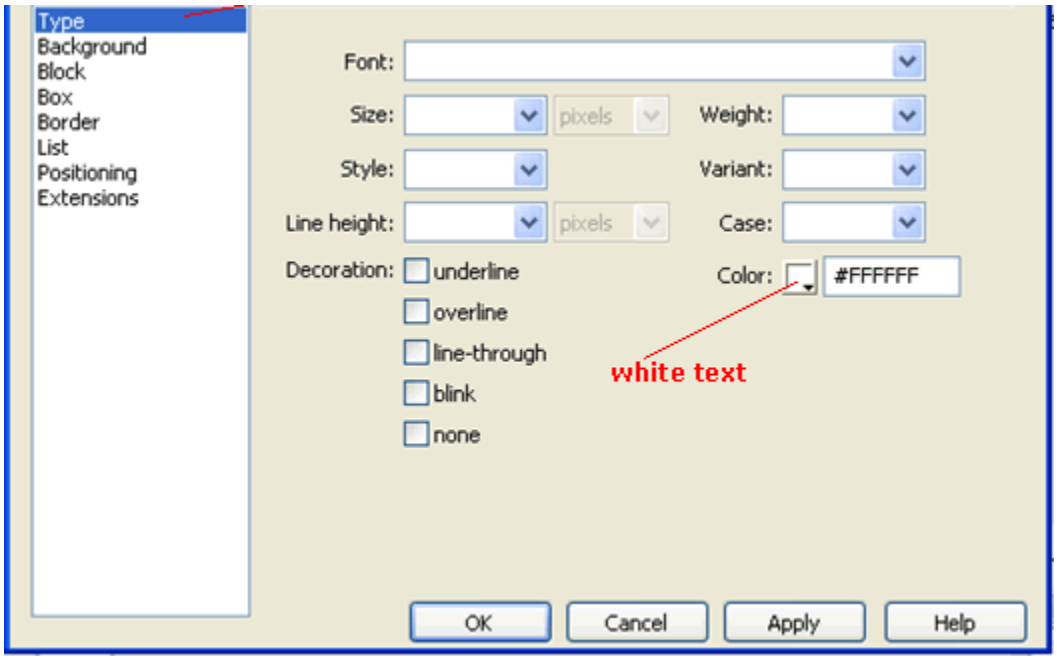


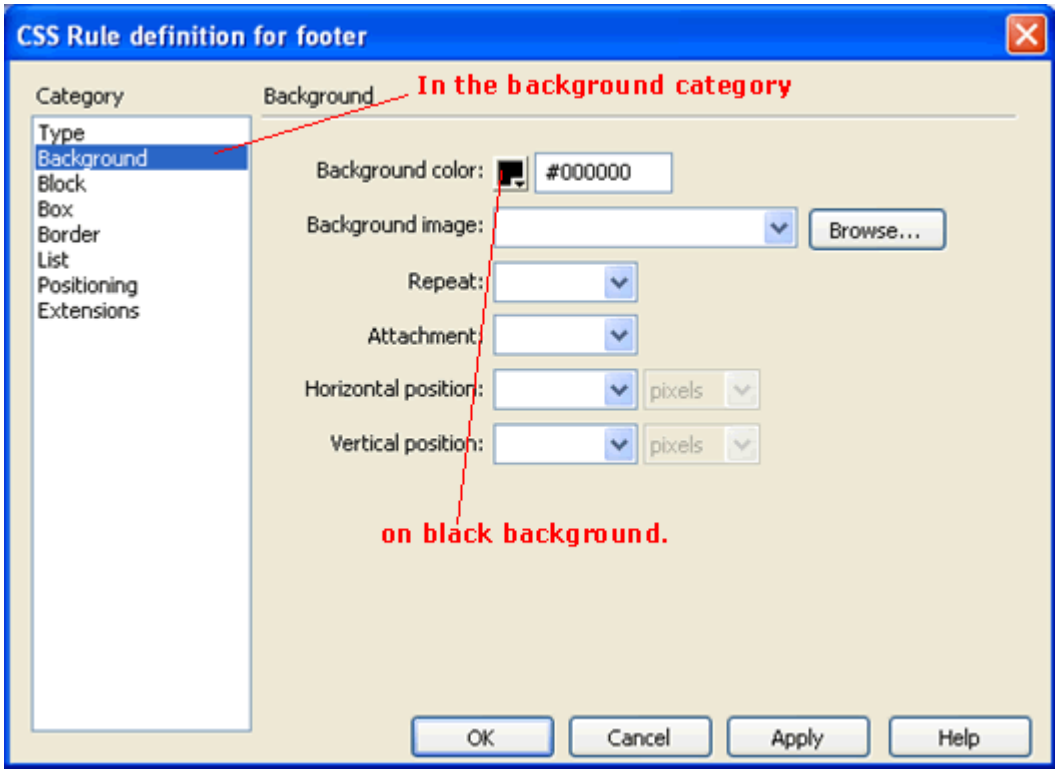
After clicking the "Add CSS Rule" button, you get the dialog shown below. This time, we will learn about the ***id selector*** rule, so select the "Advanced" selector type and type **#footer** for the selector. The # symbol in front indicate that this is to be an ***id selector***.



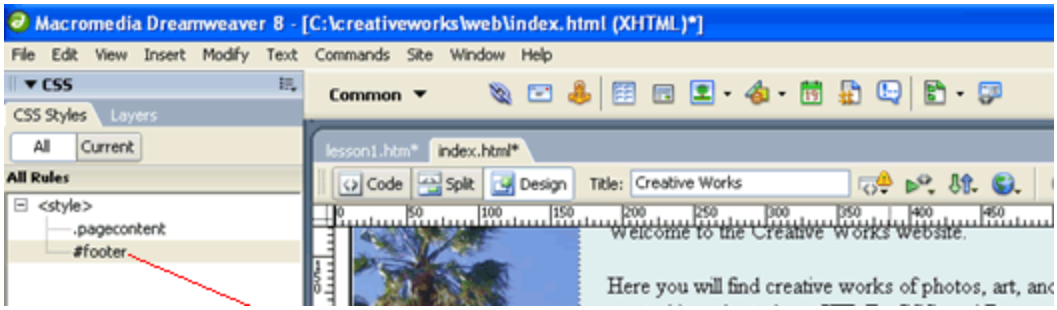
Since the last class, we were using an internal style sheet, we will continue to use this internal style sheet for this rule. Therefore, select "**This document only**" and then click **OK**.

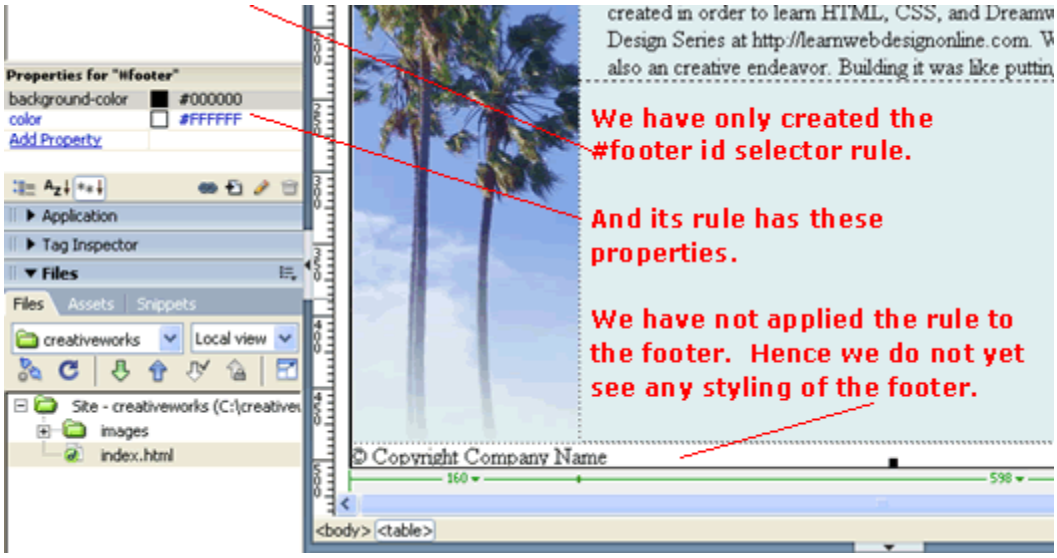
Then select white on black for the CSS rule definition as shown below...


Type
Background
Block
Box
Border
List
Positioning
Extensions
Font:
Size:
pixels
Weight:
Style:
Variant:
Line height:
pixels
Case:
Decoration:
underline
overline
line-through
blink
none
Color:
#FFFFFF
white text
OK
Cancel
Apply
Help

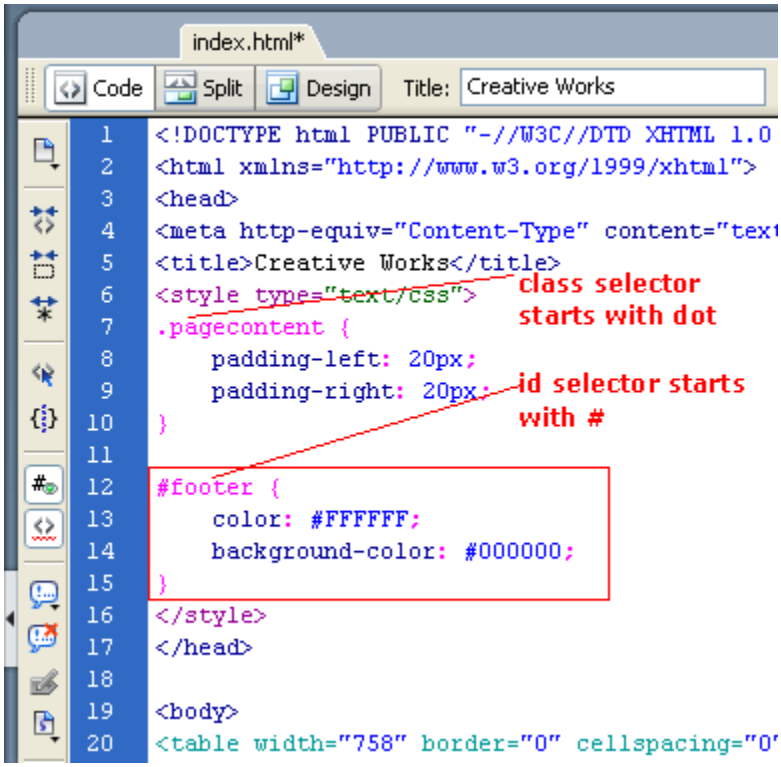

CSS Rule definition for footer
Category
Type
Background
Block
Box
Border
List
Positioning
Extensions
Background
In the background category
Background color:
#000000
Background image:
Browse...
Repeat:
Attachment:
Horizontal position:
pixels
Vertical position:
pixels
on black background.
OK
Cancel
Apply
Help

And click **OK**.

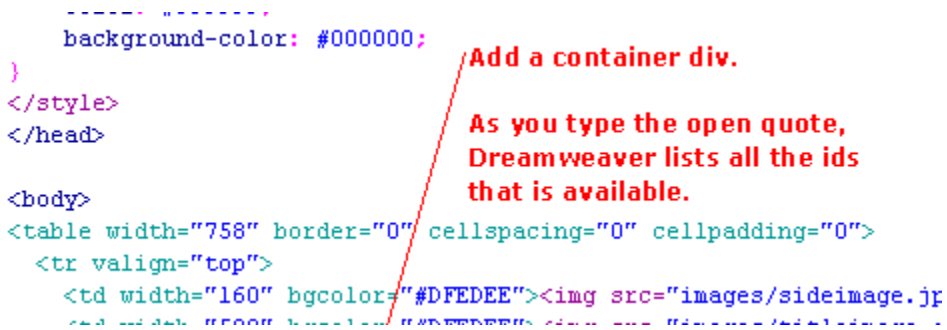

Macromedia Dreamweaver 8 - [C:\creativeworks\web\index.html (XHTML)*]
File Edit View Insert Modify Text Commands Site Window Help
CSS
CSS Styles
Layers
All
Current
All Rules
<style>
.pagecontent
#footer
Common
lesson1.htm*
index.html*
Code
Split
Design
Title:
Creative Works
Here you will find creative works of photos, art, and



Now switch back to the code view, and see our id **selector rule** in the internal CSS styles section.

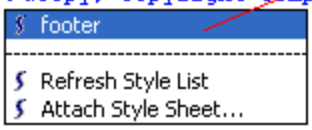


Now we want to attach the **#footer** style to our footer text. Start adding a ** ** tag around our footer text and give it an **id** attribute as shown below.



```
    <td width="596" bgcolor="#DFEDEE"><img src="images/circleImage.g

    <div class="pagecontent">
        <p>Welcome to the Creative Works website.</p>

        <p>Here you will find creative works of photos, art, and pc
    </div>  </td>
  </tr>
  <tr>
    <td colspan="2"><div id="">&copy; Copyright Company Name</td>
  </tr>
</table>
</body>
</html>
```

We have only one choice, which is the footer id selector. So pick that one.



After selecting the **footer** id selector and closing the **div** tag, you should have something like this...

```
    <div class="pagecontent">
        <p>Welcome to the Creative Works website.</p>

        <p>Here you will find creative works of photos, art, and poetry. This i
    </div>  </td>
  </tr>
  <tr>
    <td colspan="2"><div id="footer">&copy; Copyright Company Name</div></td>
  </tr>
</table>
</body>
```

Added a wrapper div

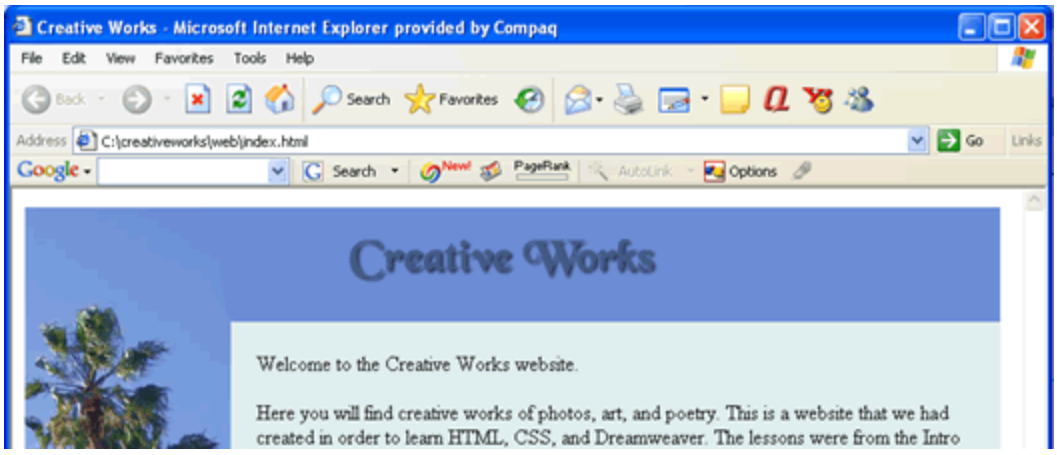
© is the copyright symbol

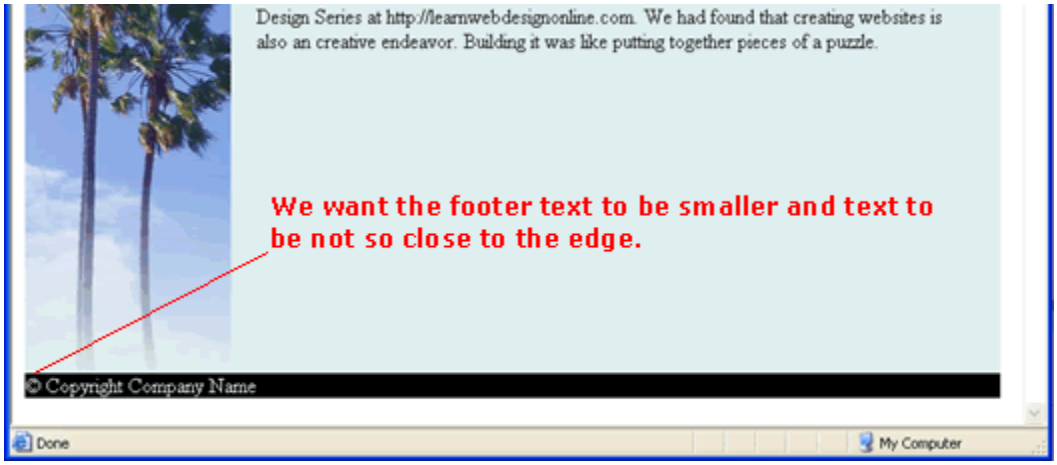
Now you may wonder, when do you use a class selector and when do you use an **id** selector. The main difference is that an **id** value can only be used once in the entire page; whereas the **class** value can be used many time for any element.

So if the div tag above has an **id** of footer, then no other element on that page can have an **id="footer"** attribute. Class attributes are different. Even though you already have a ** **, you can also add other ** ** anywhere else. And in fact you can add the **class="pagecontent"** attribute to any element, not just div's. So you can have **<td class="pagecontent">** for example.

So to summarize, use the id selector for unique elements such as footer. You know you will only have one footer in the page. Any other time, use the class selector.

Testing in your browser, you see that we have a stylized footer...





View live code

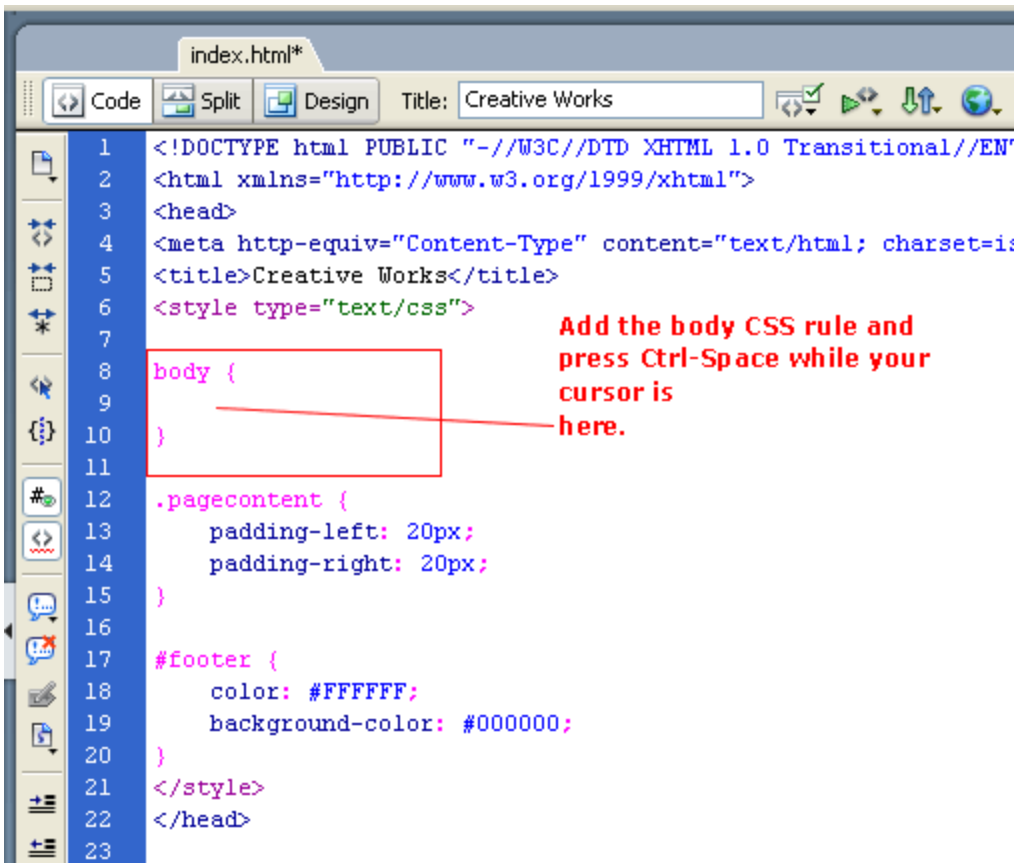
Download: **code2.zip**

Style Text

Now we want to stylize the text. Since we want a consistent typeface throughout the entire site, we will apply the text style to the whole page. In order words, we want to apply the styles to the <body> tag.

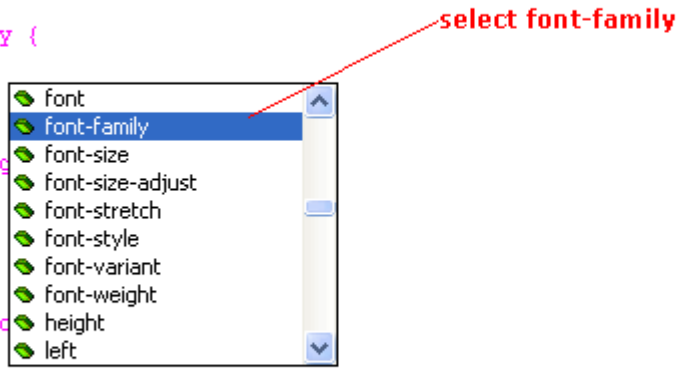
Since we learned how to use the Page Properties dialog to stylize the font to Verdana in the HTML/Dreamweaver lessons, you will learn how to hand code the font style to Geneva here. Geneva is another san-serif font type.

Open **index.html** in code view and start adding the body CSS style rule as shown...



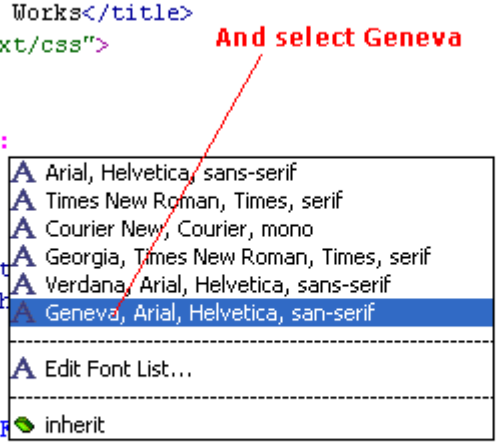
And you will see Dreamweaver's ***code assist*** comes up giving you all the possible CSS properties...

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
<style type="text/css">

body {

}

.pag

}

#foc
    background-color: #000000;
}
</style>
</head>
```



Now Dreamweaver is giving you some possible values for the **font-family** property.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
<style type="text/css">

body {
    font-family:
}

.pagecontent {
    padding-left
    padding-righ
}

#footer {
    color: #FFFF
    background-color: #000000;
}
</style>
</head>
```



You should now have the below CSS rule. Don't forget to add the semi-colon...

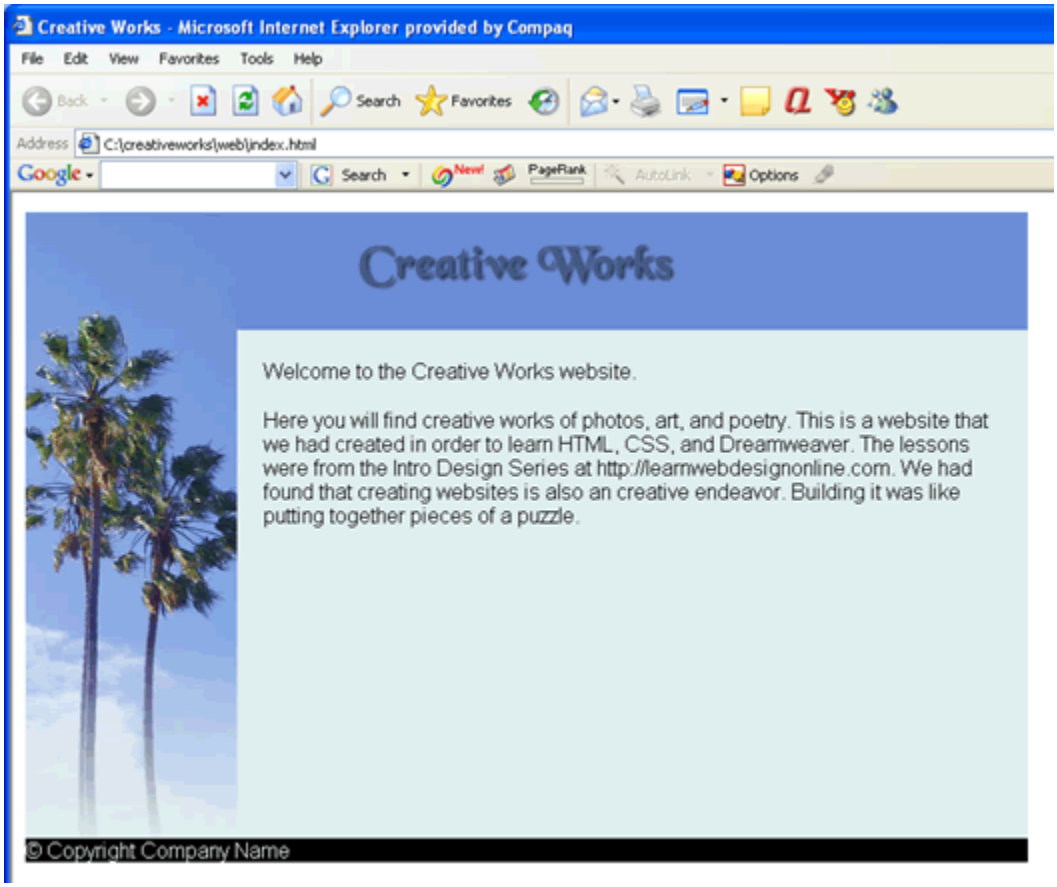
```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
<style type="text/css">

body {
    font-family:Geneva, Arial, Helvetica, san-serif;
}
```

```
.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}

#footer {
    color: #FFFFFF;
    background-color: #000000;
}
</style>
</head>
```

Add a semi-colon.

For the value of our **font-family** property, we have serveral fonts listed: **Geneva, Arial, Helvetica, san-serif**. This means that if the user's computer has the Verdana font installed, the browser will render the text in that font. If not, the browser will try Arial font, and then try Helvetica. If all else fails, the browser will render it in a generic sans-serif font.

Testing in browser, we see that our font had changed to Geneva...



Using em to Size Fonts

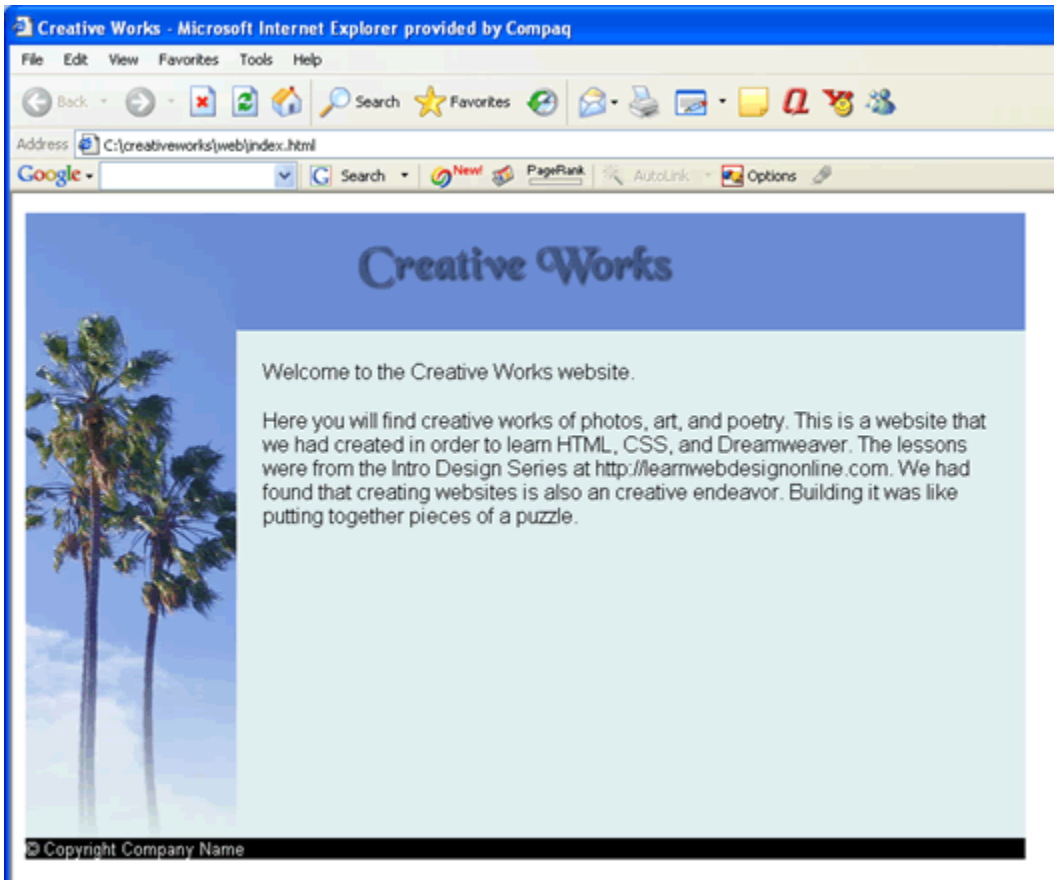
We want to make the footer font 80% smaller than normal. So add the following property to the **#footer** id selector...

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
<style type="text/css">

body {
```

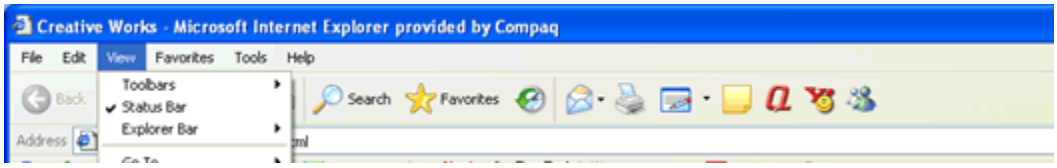
```
    font-family:Geneva, Arial, Helvetica, san-serif;
}

.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}

#footer {
    color: #FFFFFF;
    background-color: #000000;
    font-size: 0.8em;
}
</style>
</head>
```

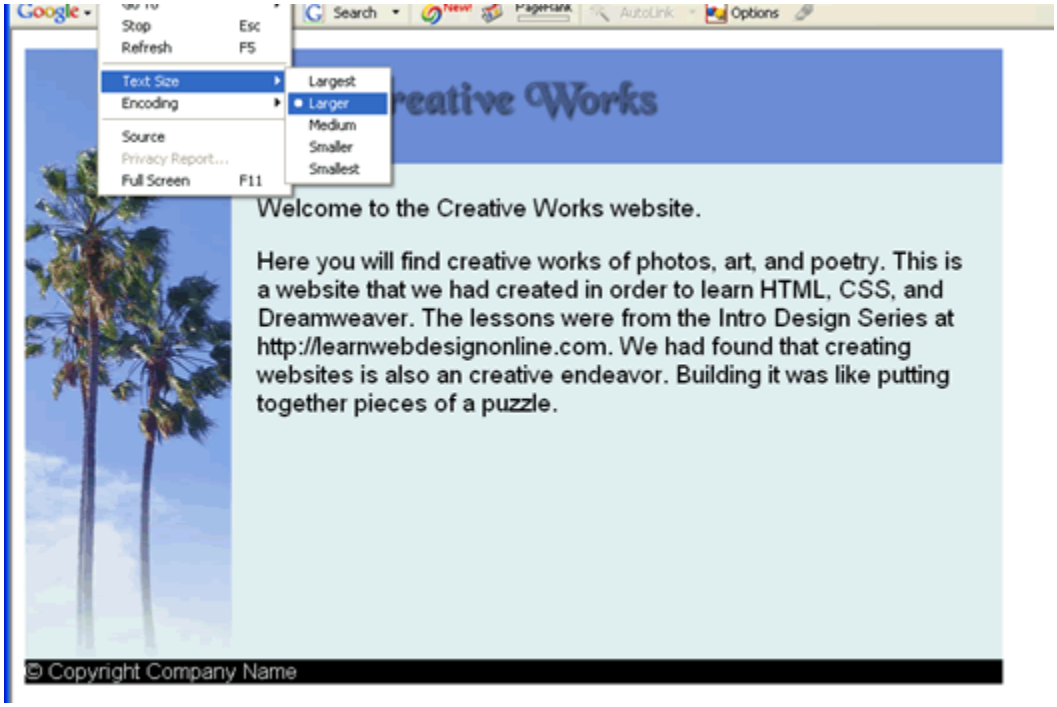
em is a relative font size unit. By saying **0.8em**, we are telling the browser to render that font size to be 80% smaller than what you would normally have rendered it.

And in the browser, you see that indeed that is the case...



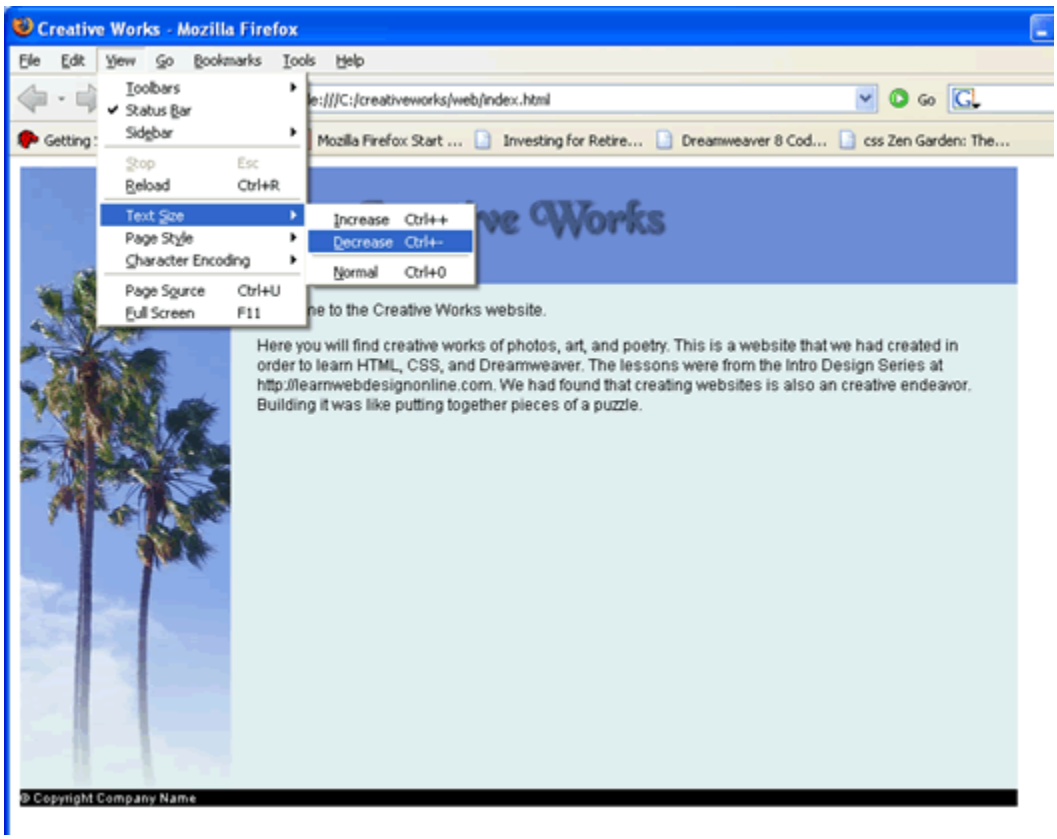
The **em** unit is known as a ***relative font size*** unit because it is not fixed at a specific pixel or point size. For example, the user is free to increase the font size in his or her browser ...





And the font size will become bigger. Yet the footer text always remains 80% the size of the main text.

Similarly if the user resizes the font smaller. This time, I use Firefox...



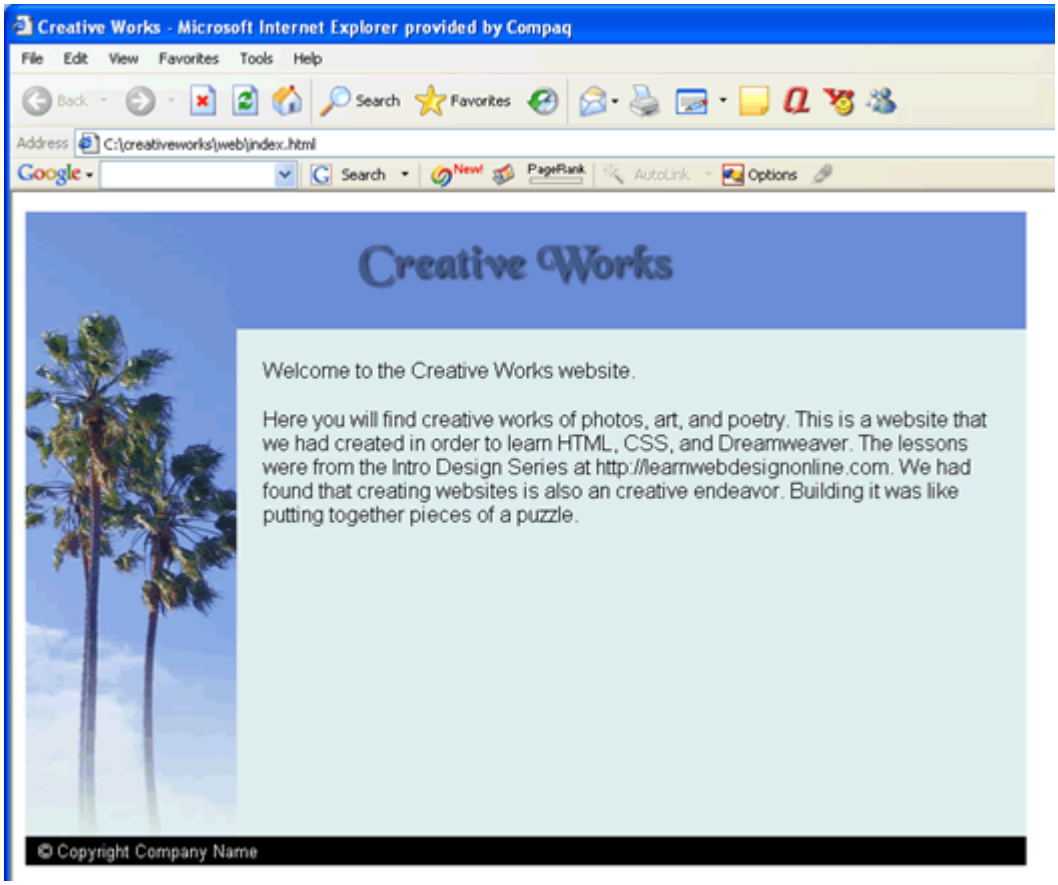
Now make sure you set both browser back to normal text size.

Adding Padding to Footer

Add the following padding to the footer style...

```
<style type="text/css">

body {
    font-family:Geneva, Arial, Helvetica, san-serif;
}

.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}

#footer {
    color: #FFFFFF;
    background-color: #000000;
    font-size: 0.8em;
    padding-top: 3px;
    padding-bottom: 3px;
    padding-left: 10px;
    padding-right: 10px;
}
</style>
```

And now it looks better...



CSS Shortcut Notation

The above four CSS padding properties can be combined into one property as in ...

```
#footer {
    color: #FFFFFF;
    background-color: #000000;
    font-size: 0.8em;
    padding: 3px 10px;
}
```

This is the top and bottom padding.

This is the left and right padding.

You can only do this if the top and bottom padding are the same, and if the left and right padding are the same. And if you test it in your browser, it looks exactly the same as before.

This ***shortcut notation*** will cut down on the typing. And more importantly, it will reduce the size (slightly) of the CSS file making it a faster download.

The shortcut notation goes even further. If your top, bottom, left, and right padding are all the same, then you can write ...

```
#footer {
    color: #FFFFFF;
    background-color: #000000;
    font-size: 0.8em;
    padding: 3px;
}
```

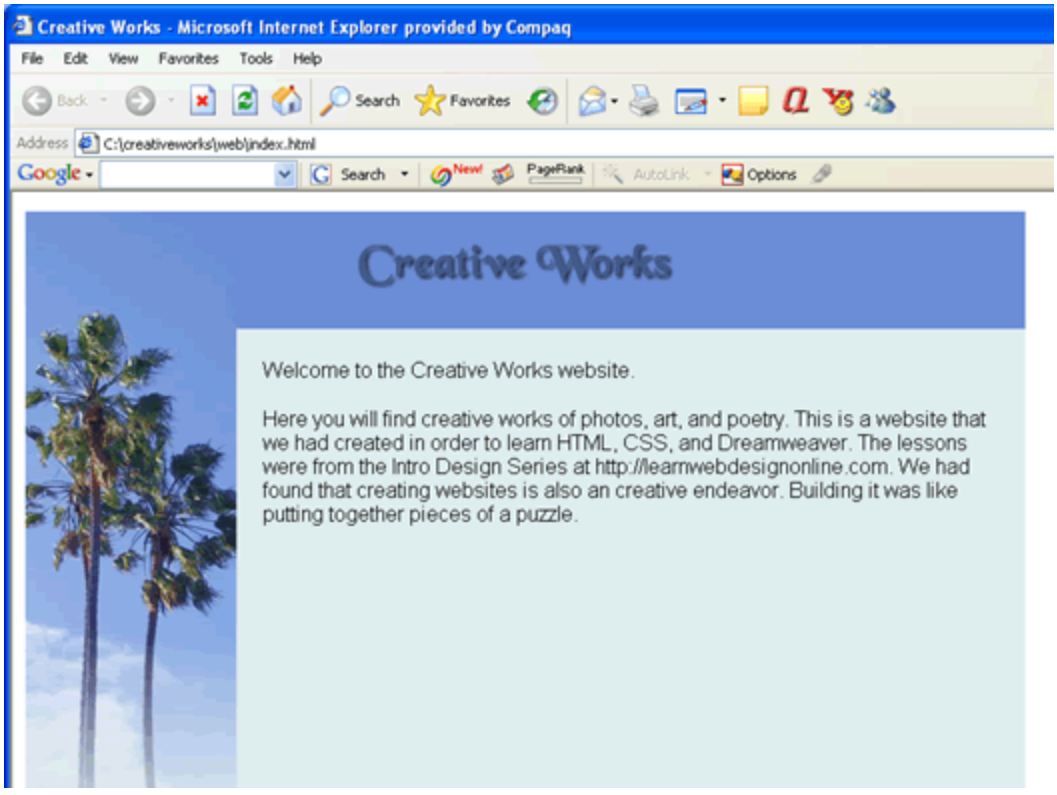
Top, bottom, left, and right padding all set to 3px.

View live code

Download: **code3.zip**

Navigation Bar

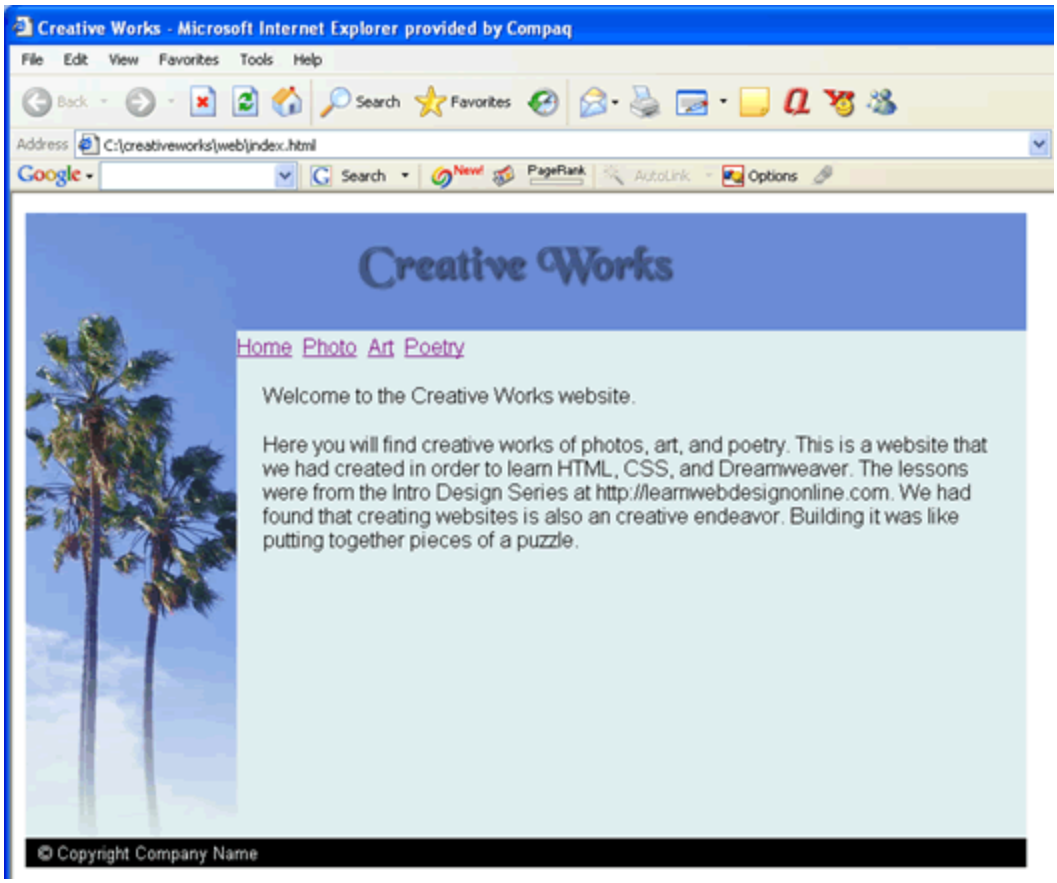
Now we will add in the navigation bar right under the title image and above the page content.




```
<body>
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160" bgcolor="#DFEDEE"><img src="images/sideimag
    <td width="598" bgcolor="#DFEDEE"><img src="images/titleima

    <div id="navbar">
        <a href="index.html">Home</a> 
        <a href="photo.html">Photo</a> 
        <a href="art.html">Art</a> 
        <a href="poetry.html">Poetry</a> 
    </div>

    <div class="pagecontent">
        <p>Welcome to the Creative Works website.</p>

        <p>Here you will find creative works of photos, art, ar
    </div>  </td>
```

We use a div as a containing element for the navigation bar. And we gave this div an id called **navbar**. It would have worked if we used a class. But since we are going to have at most one navbar per page, I choose to use and id selector rule to stylize the navbar.

Inside this navbar, we put in four links as shown.



We need to attach a style to the navbar id. We give it a gray background ...

```
body {
    font-family:Geneva, Arial, Helvetica, san-serif;
}

.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}

#footer {
    color: #FFFFFF;
    background-color: #000000;
    font-size: 0.8em;
    padding: 3px 10px;
}

#navbar {
    background-color: #D6D6D6;
}

</style>
```

Now we give it a one pixel dark gray border all around...

```
#navbar {
    background-color: #D6D6D6;
    border-color: #A8A8A8;
    border-style: solid;
    border-left-width: 1px;
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-top-width: 1px;
}
```

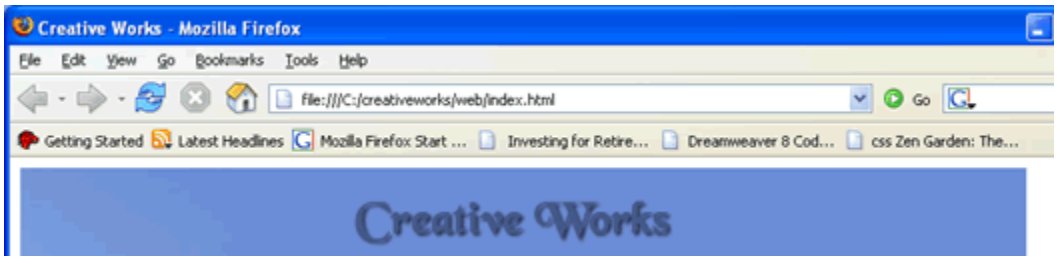
This can be be written in short-hand form by ...

```
#navbar {
    background-color: #D6D6D6;
    border: 1px solid #A8A8A8;
}
```

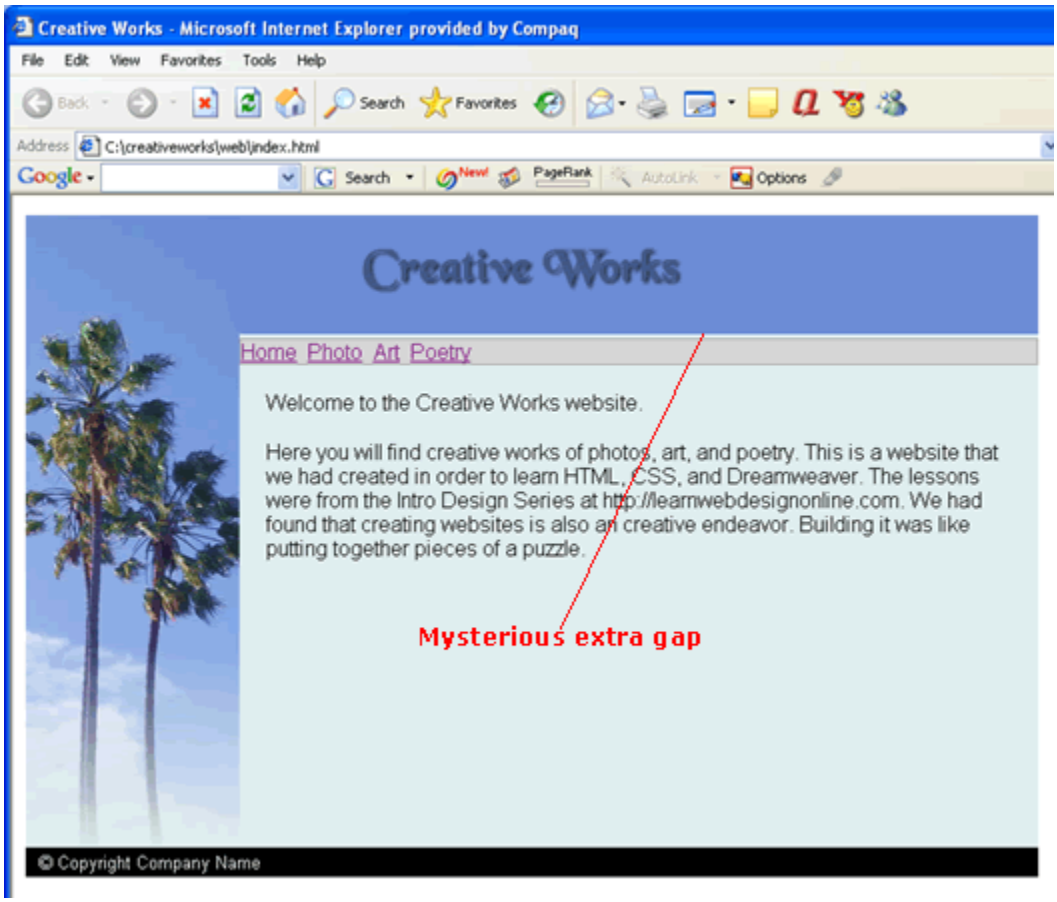
width style color

We just lump all the values together after the colon. The order of the values does not really matter, but I always like to put it in this form. Since the property name is **border**, the values are applied to all four borders.

And if you look at it using Firefox, it should look like ...



However, when you look at it in Internet Explorer 6.0 for Windows, you get an mysterious space ...



Now you know that different browsers can sometimes render the same code in different ways. This cross-browser compatibilities issue becomes even greater with CSS layout and code. In the Intermediate CSS course, we look at some of these issues and their workarounds. That is why we should always test in both browsers.

The gap in Internet Explorer was caused by the line break between the **<img>** tag and the ** ** tag.

```
<body>
```

```
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160" bgcolor="#DFEDEE"><img src="images/sideimag
    <td width="598" bgcolor="#DFEDEE"><img src="images/titleima

    <div id="navbar">
        <a href="index.html">Home</a> 
        <a href="photo.html">Photo</a> 
        <a href="art.html">Art</a> 
        <a href="poetry.html">Poetry</a> 
    </div>

    <div class="pagecontent">
        <p>Welcome to the Creative Works website.</p>

        <p>Here you will find creative works of photos, art, ar
    </div>  </td>
```

Mysterious gap caused by extra spacing and line break here.

Even if we remove the blank line ...

```
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160" bgcolor="#DFEDEE"><img src="images/sideimage.jpg" widt
    <td width="598" bgcolor="#DFEDEE">
    <img src="im... />
    <div id="navbar">
        <a href="index.html">Home</a> 
        <a href="photo.html">Photo</a> 
        <a href="art.html">Art</a> 
        <a href="poetry.html">Poetry</a> 
    </div>
```

collapsed part of the img tag

The mysterious gap caused by the line-break.

the mysterious gap does not disappear. That is because it is the line-break between the <img> and the tag. Note that I have code-collapsed the long list of attributes of the img tag. See **here** for more on Dreamweaver 8's new code collapse feature.

When we have no space between the two tags...

```
<table width="758" border="0" cellspacing="0" cellpadding="0">
  <tr valign="top">
    <td width="160" bgcolor="#DFEDEE"><img src="images/sideimage.jpg" widt
    <td width="598" bgcolor="#DFEDEE">
    <img src="im... /><div id="navbar">
        <a href="index.html">Home</a> 
        <a href="photo.html">Photo</a> 
        <a href="art.html">Art</a> 
        <a href="poetry.html">Poetry</a> 
    </div>
```

Mysterious gap disappears when there is no space between these two tags.

then the space disappears.

However, I don't like to have the <img> and the on the same line. It looks funny. So an alternative is to put a wrapper div around the <img> tag. As in ...

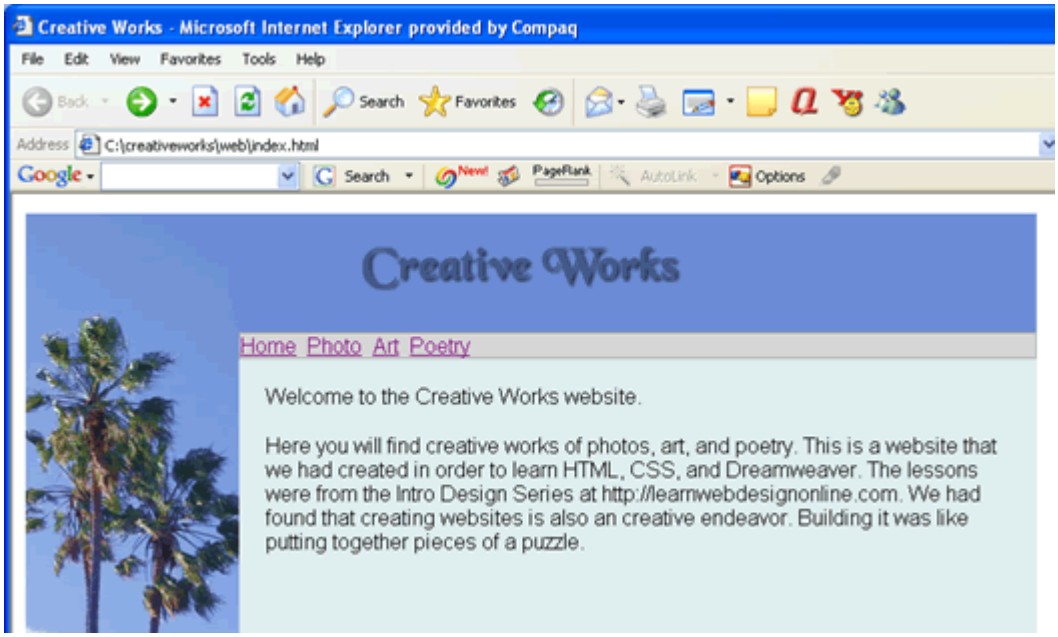
```
<div><img src="im... /></div>
```

I wrap the img

```
<div id="navbar">
    <a href="index.html">Home</a> 
    <a href="photo.html">Photo</a> 
    <a href="art.html">Art</a> 
    <a href="poetry.html">Poetry</a> 
</div>
```

tag with its own div, making sure there is no space here.

And this looks fine without the extra gap.



If you still see the gap, expand the collapsed code and make sure there is not space between the end of the <img> tag and the tag. I have seen some problems with the code collapse feature giving extra spaces at the end of the img tag.

The nice thing about using the extra div is now I can organize my code the way I want it and can even add in HTML comments as well...

```
<!-- title image -->
<div><img src="im..." /></div>

<!-- navigation bar -->
<div id="navbar">
    <a href="index.html">Home</a> 
    <a href="photo.html">Photo</a> 
    <a href="art.html">Art</a> 
    <a href="poetry.html">Poetry</a> 
</div>
```

HTML comments are for our own notes.

Centering the links

And lastly, we want the links to be centered along the horizontal navigation bar. We can either set an attribute in the tag itself...

```
<!-- title image -->
```

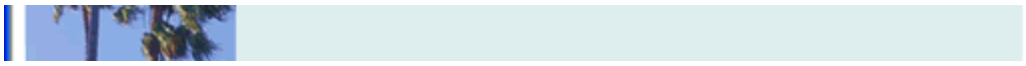
align="center" on the div tag

```
<div><img src="im..." /></div>

<!-- navigation bar -->
<div id="navbar" align="center">
    <a href="index.html">Home</a> 
    <a href="photo.html">Photo</a> 
    <a href="art.html">Art</a> 
    <a href="poetry.html">Poetry</a> 
</div>
```

or add a style property in the CSS...

```
<style type="text/css">

body {
    font-family:Geneva, Arial, Helvetica, san-serif;
}

.pagecontent {
    padding-left: 20px;
    padding-right: 20px;
}

#footer {
    color: #FFFFFF;
    background-color: #000000;
    font-size: 0.8em;
    padding: 3px 10px;
}

#navbar {
    background-color: #D6D6D6;
    border: 1px solid #A8A8A8;
    text-align: center;
}

</style>
```

Both will give the same results...



However, let choose to do it using the latter method of the CSS property because that provides better decoupling of the styling from the structural markup of the HTML.

Note the differences between the attribute in the HMTL ...

```
<div align="center">
```

and a CSS property...

```
text-align: center;
```

First of all the names are not the same. The former is align, the latter is text-align. The attribute uses an equal sign. The property using a colon. The attribute value is within quotes. The property is not.

Adding Padding

Looks to me that I want to have the nav bar higher so that the text have more air to beathe on the top and bottom. So lets add some top and bottom padding...

```
#navbar {
    background-color: #D6D6D6;
    border: lpx solid #A8A8A8;
    text-align: center;
    padding-top: 3px;
    padding-bottom: 3px;
}
```

which we can write in short-from as ...

```
#navbar {
    background-color: #D6D6D6;
    border: lpx solid #A8A8A8;
    text-align: center;
    padding: 3px 0;
}
```

top and bottom padding

left and right padding

Note that when we have 0px, we can write it as a bare 0. Since zero is zero regardless of the units.

Okay, looks much better. However, I would like more spaces between the links ...

```
<!-- title image -->
<div><img src="im..." /></div>

<!-- navigation bar -->
<div id="navbar">
    <a href="index.html">Home</a>    
    <a href="photo.html">Photo</a>    
    <a href="art.html">Art</a>    
    <a href="poetry.html">Poetry</a>    
</div>
```

Add more non-breaking spaces.

Okay good enough. Remember to check in both Firefox and Internet Explorer. The font on the links is still a bit too big. But we will style that in the next lesson when we style the links and implement their hover over effect.

View live code

Download: **code4.zip**

Style Links

Let make the navigation links smaller and bold it. Add the following CSS rule ...

```
#navbar a {
    font-size: 0.8em;
    font-weight: bold;
}
```

This selector is a bit different in that it has two elements: **#navbar a**. This means that the selector will select all **<a>** tags that are nested within the element with id **navbar**. Hence we are styling all links within the navbar div.

Now we want to add an hover effect on the link. We first stylize the default state of the links to be a deep blue #003366 color without underline...

```
#navbar a {
    font-size: 0.8em;
    font-weight: bold;
    color: #003366;
    text-decoration: none;
}
```

For the hover state, we want to brighten up the blue to color #0000FF and we want to add in the underline. For the hover state, we need a whole new CSS rule that employes the **:hover** psuedo-class as shown...

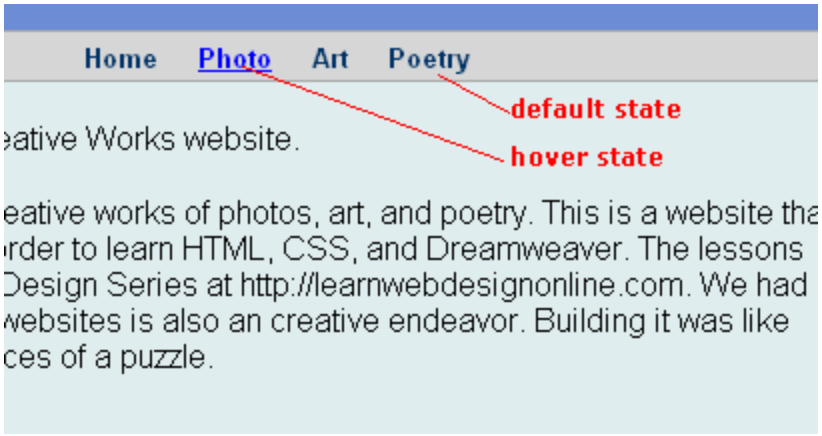
```
#navbar a {
    font-size: 0.8em;
    font-weight: bold;
    color: #003366;
    text-decoration: none;
}
```

The default state

```
#navbar a:hover {
    color: #0000FF;
    text-decoration: underline;
}
```

The hover state

This rule will apply its style when the mouse is hovered over the **<a>** tag that is within the **navbar** div. So the default states and hover states looks like ...



Short Cut Color Notation

Note that if you hexdecimal color code have their digits in pairs ...

```
#navbar a {
    font-size: 0.8em;
    font-weight: bold;
    color: #003366;
    text-decoration: none;
}

#navbar a:hover {
    color: #0000FF;
    text-decoration: underline;
}
```

#003366 can be written as #036

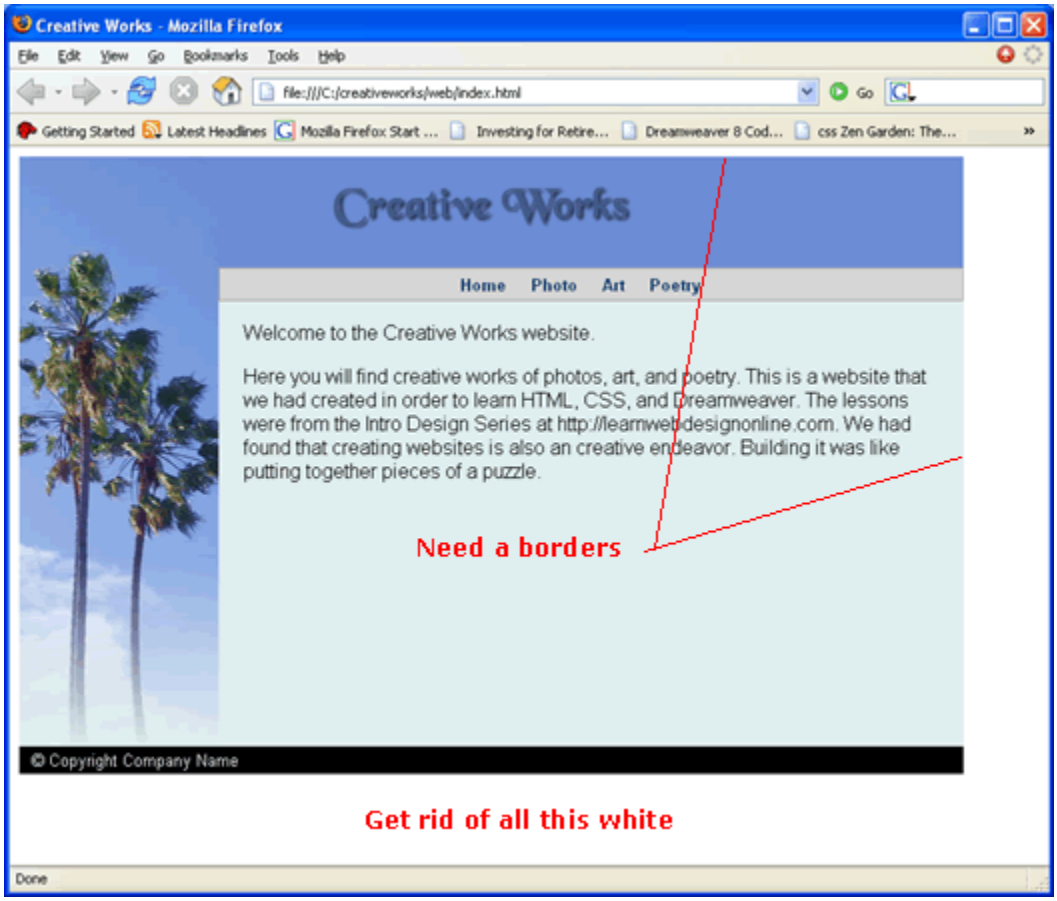
#036

This can be written as #00f
(capitalization in hexidecimal code
does not matter)

then you can use the shortcut notation as in the below...

```
#navbar a {
    font-size: 0.8em;
    font-weight: bold;
    color: #036;
    text-decoration: none;
}

#navbar a:hover {
    color: #00f;
    text-decoration: underline;
}
```

Our Page Now Looks Like



The Final Touches

What is still left is to put a dark gray (#666666) background on the **<body>** tag, so we don't see all this white. In the last class, we had set the background color though the page properties. This time we will just write the CSS code ...

```
body {
    font-family:Geneva, Arial, Helvetica, san-serif;
    background-color: #666666;
}
```

We want to put a border around our page (or table in this case).

```
#navbar a:hover {
    color: #00f;
    text-decoration: underline;
}

.maintable {
    border: 1px solid black;
}

</style>
</head>

<body>
<table width="758" border="0" cellspacing="0" cellpadding="0" class="maintable">
```

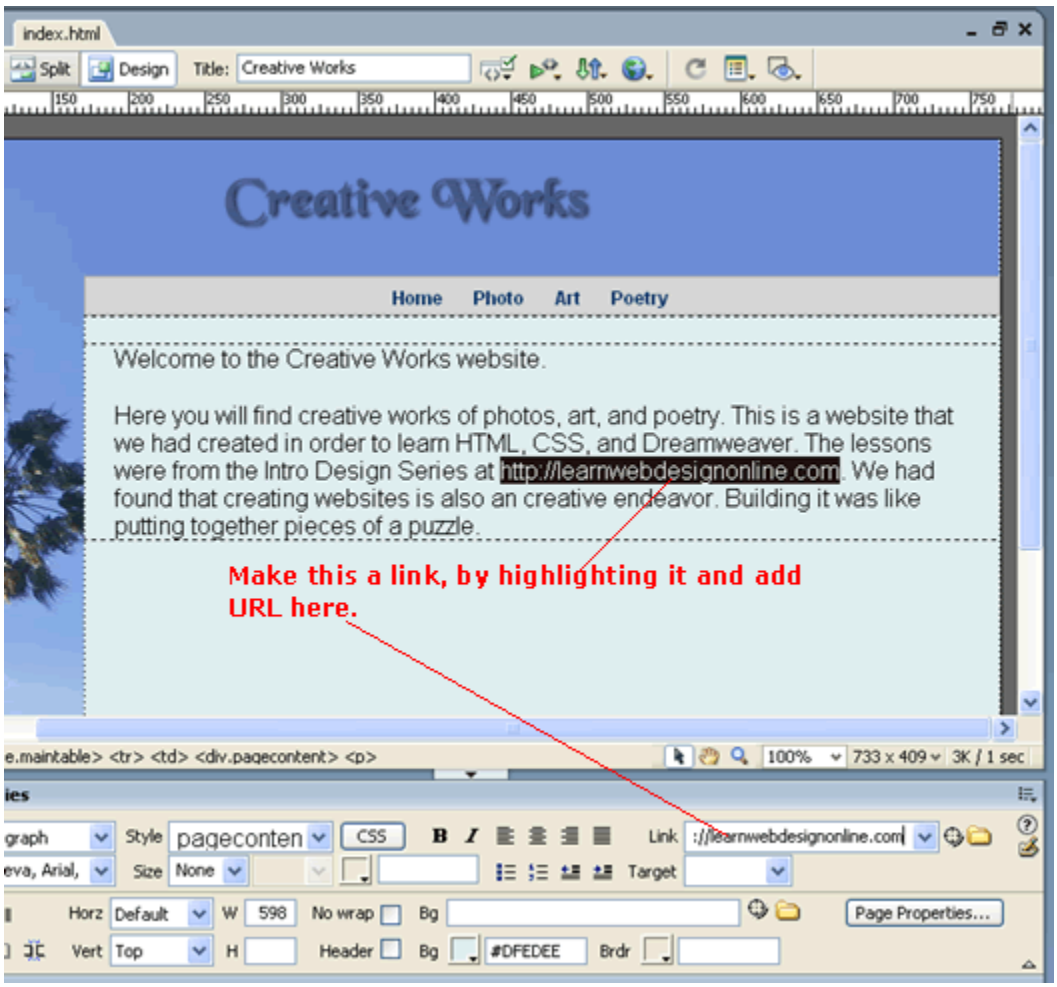
We attach a class to our table.

And gave styling for that class.

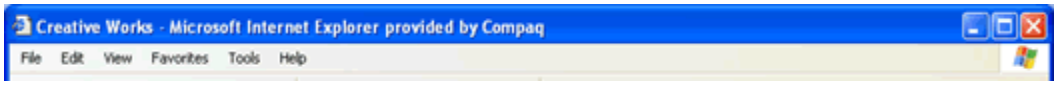
We used the CSS color code instead of the hexidecimal #000000 for black.

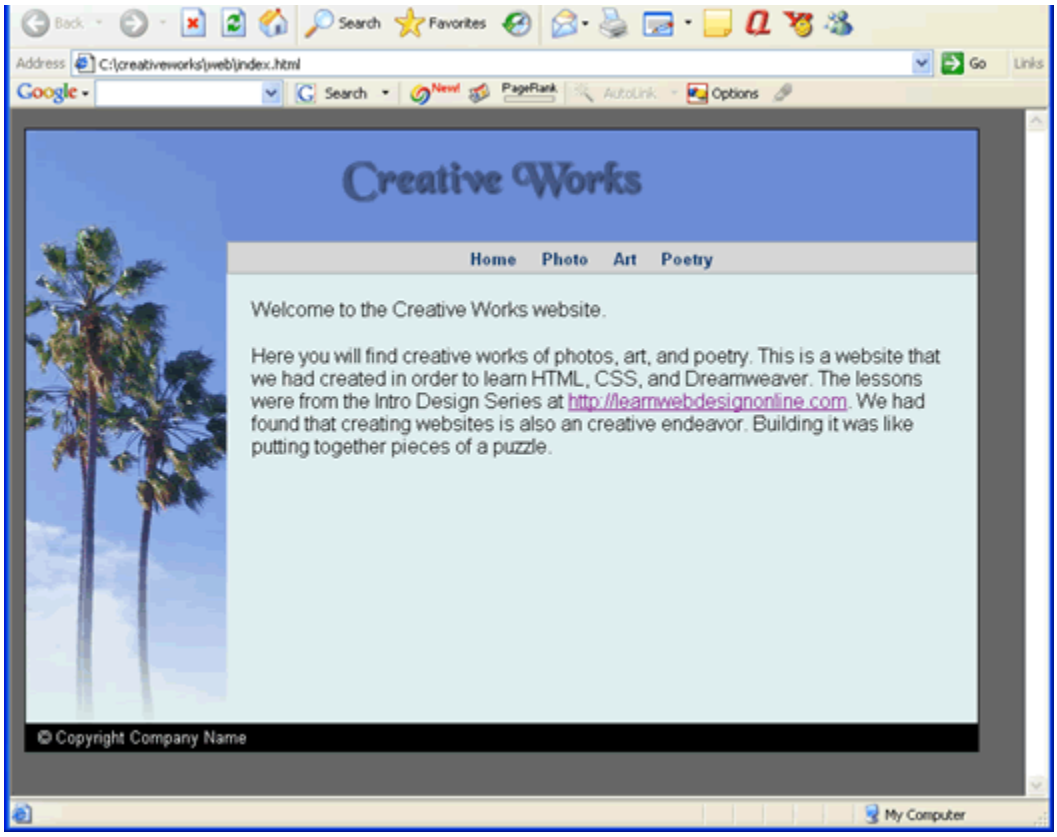
Note that we used the color name value for black this time instead of the hexidecimal code. You can do this provided that the color is one of the sixteen official color name values listed **here**.

Switching to design view, we see that in our text, we want to make one link...



Test in both Firefox and Internet Explorer.





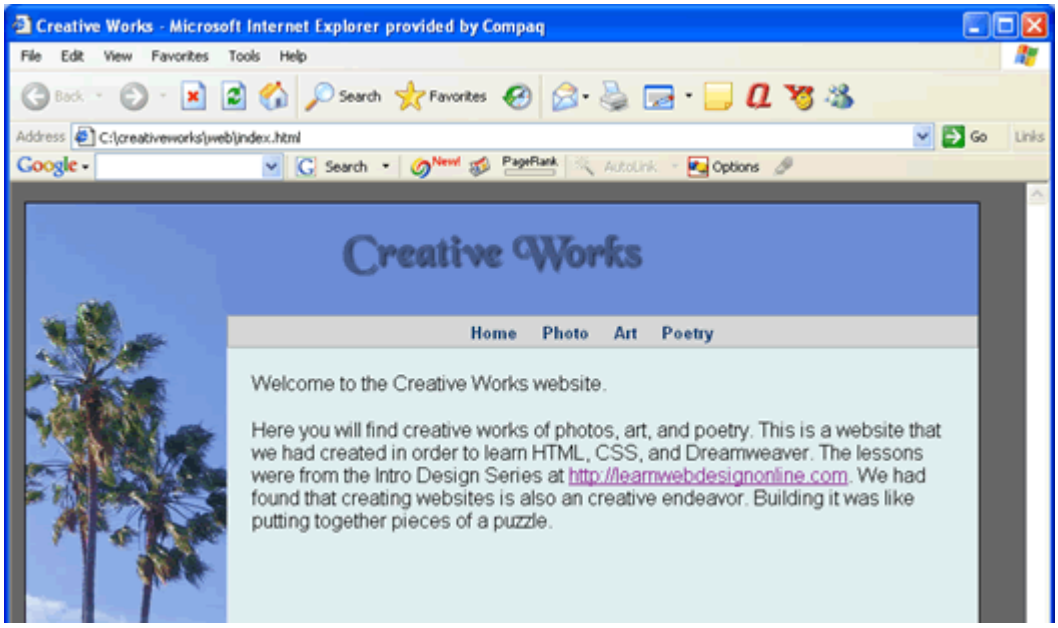
It looks much better now.

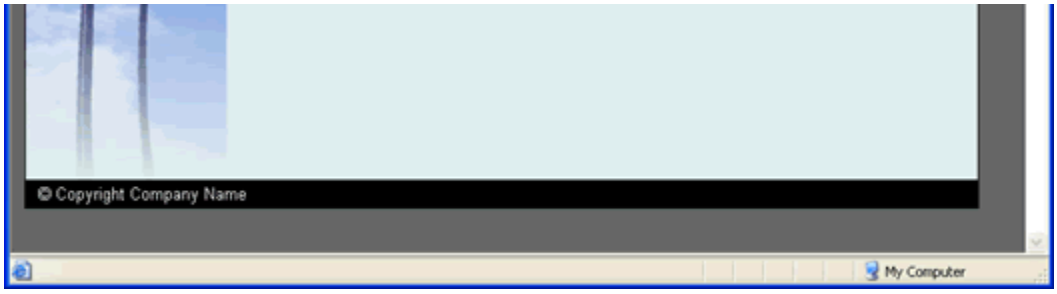
View live code

Download: **code5.zip**

Remaining Pages

We now have a completed webpage of the **index.html** file.

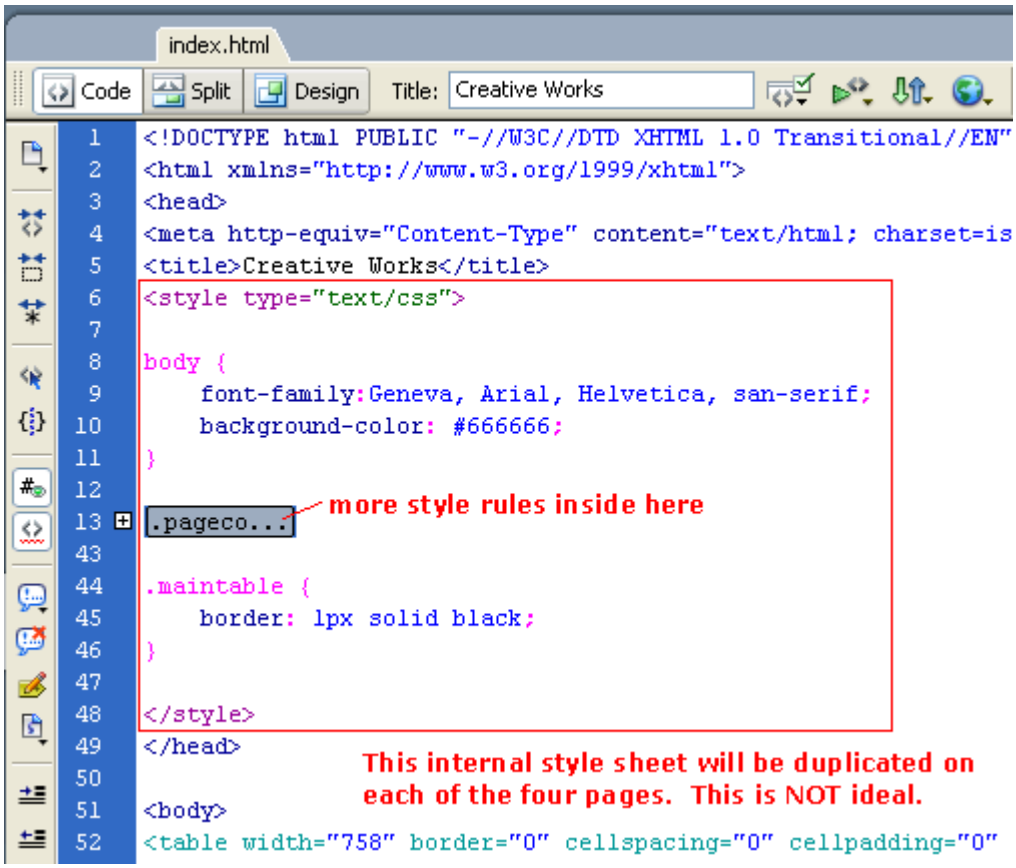




The next thing we want to do is to build the other three pages: **photo.html**, **art.html**, and **poetry.html**. These are the filenames that were linked to in the navbar...

```
<!-- navigation bar -->
<div id="navbar">
    <a href="index.html">Home</a>    
    <a href="photo.html">Photo</a>    
    <a href="art.html">Art</a>    
    <a href="poetry.html">Poetry</a>    
</div>
```

Actually, we don't really have to build those three pages. Since they are essentially the same as **index.html**, we simply will duplicate them from **index.html** and change its content. However, note that this means that the internal style sheet at the top of index.html will now be present in the other three pages as well. This is not good.



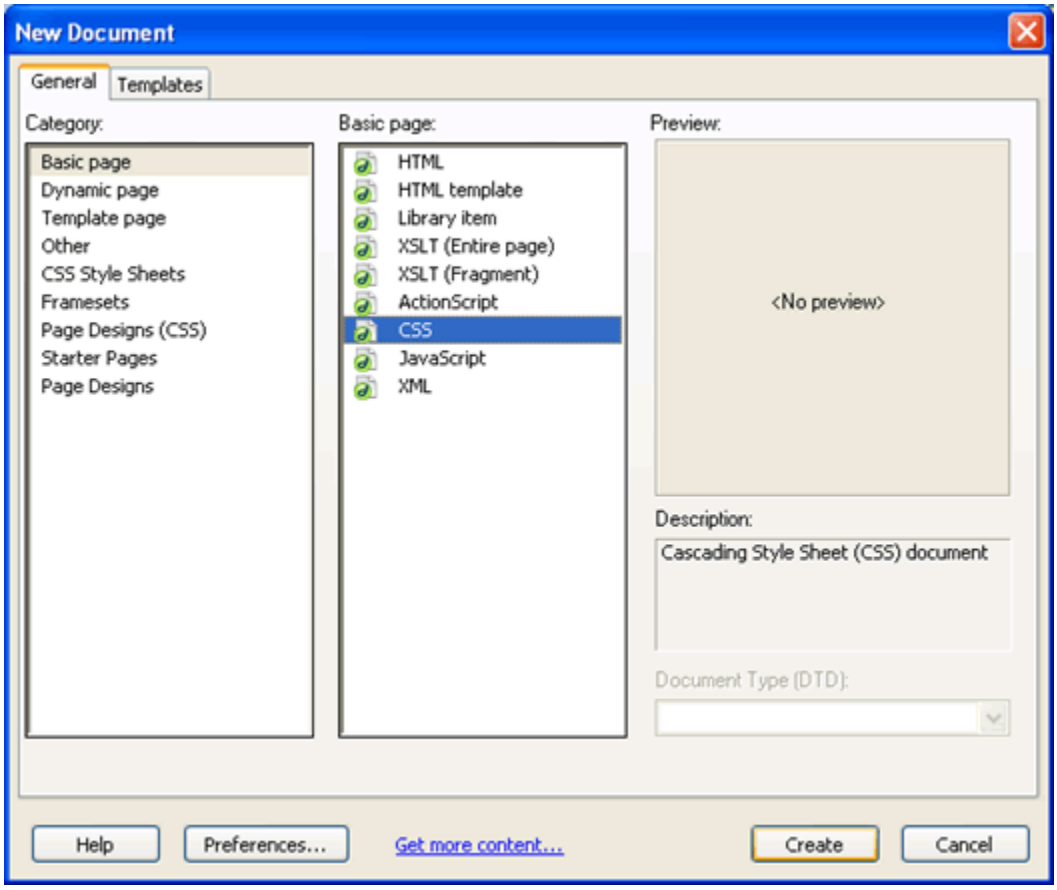
In the programming world, it is always good practice to avoid code duplication.

Imagine if you will... Let's say that you did duplicate the file with the styles in it. Now later on we decided to change on of

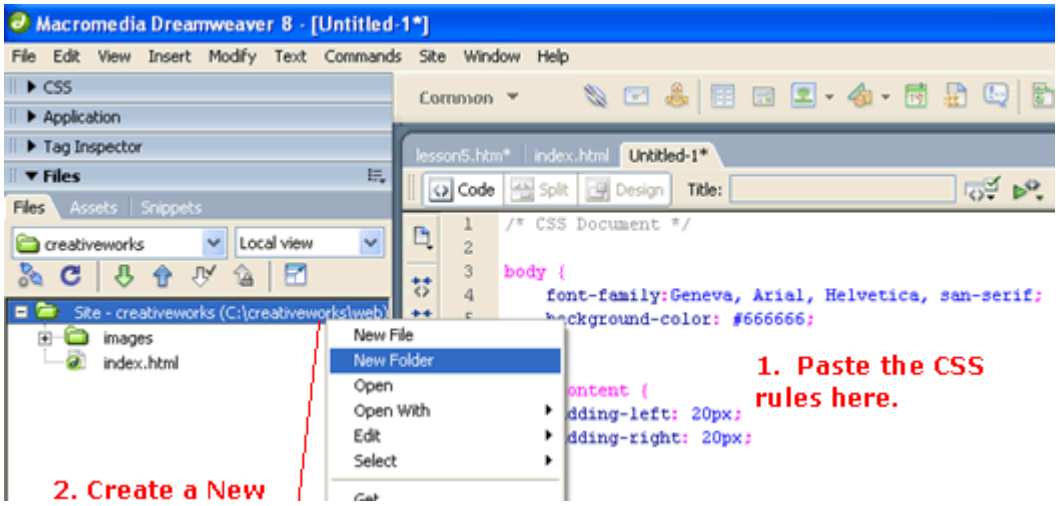
the styles (such as the **font-family** of the **body** tag to be **Georgia** instead of **Geneva**). You would then have to make the change to that rule at the top of all four files. What if this was a site with a hundred files. It would be a lot of work to make the change. Plus all that duplicated CSS code in each of the file will slow down the page downloads and increase bandwidth.

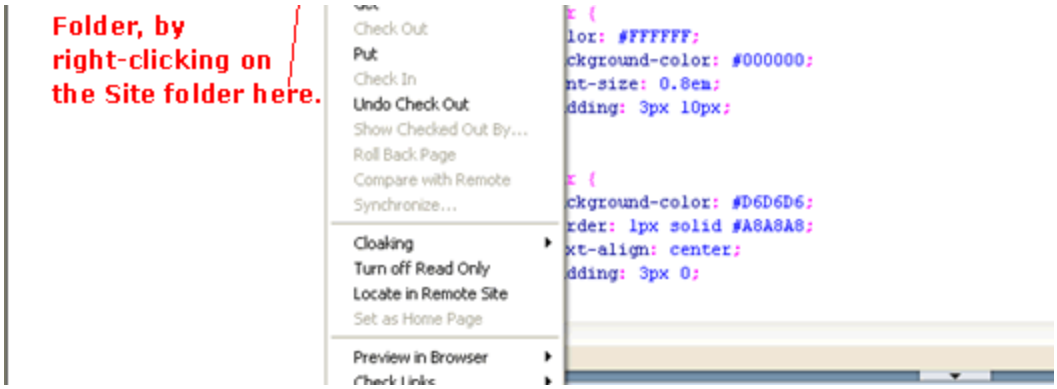
What we can do to avoid code duplication is to take all those style rules and separate it out from index.html file. We will put those style rules in an separate CSS file (also known as an external style sheet).

Do a **File -> New** and create a new CSS file as shown...

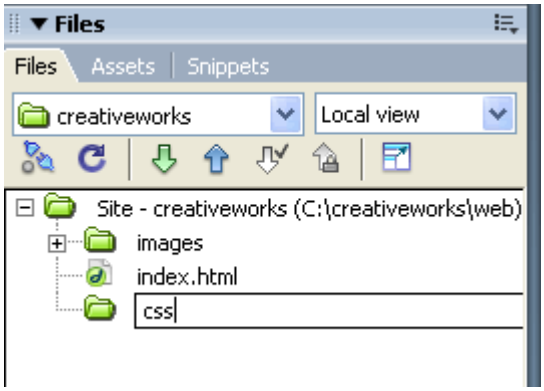


Copy and paste the CSS rules from the **index.html** file to the new untitled file. Make sure you copy only the rules and not the style tag.

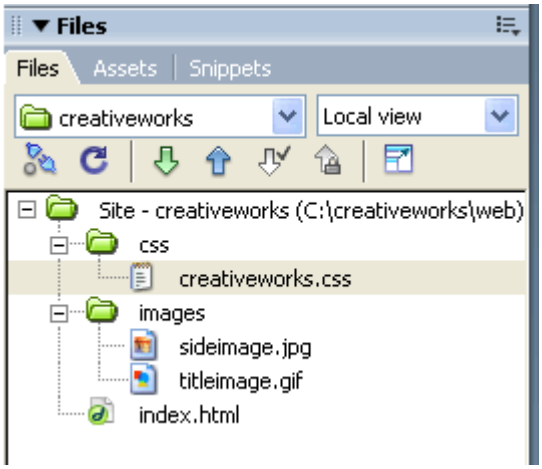




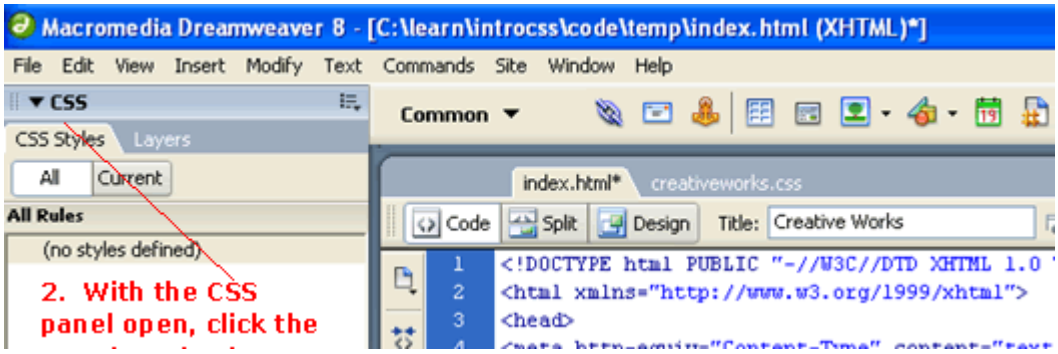
Create a new folder called **css**

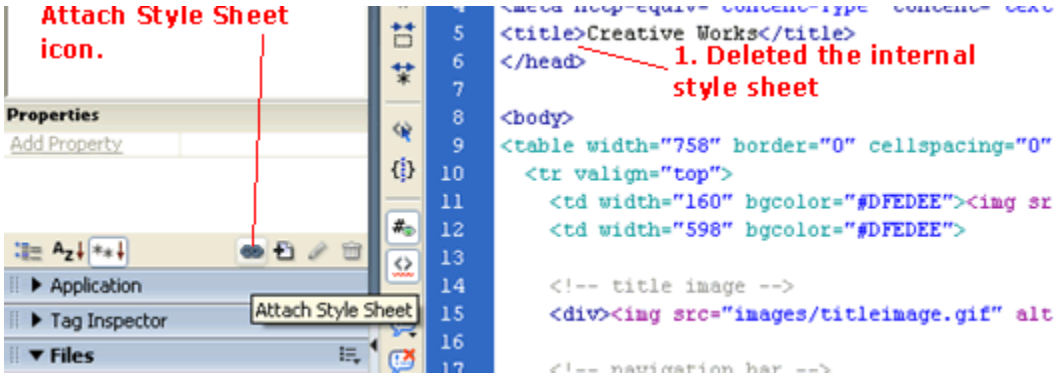


Save the new file as **creativeworks.css** into this **css** folder.

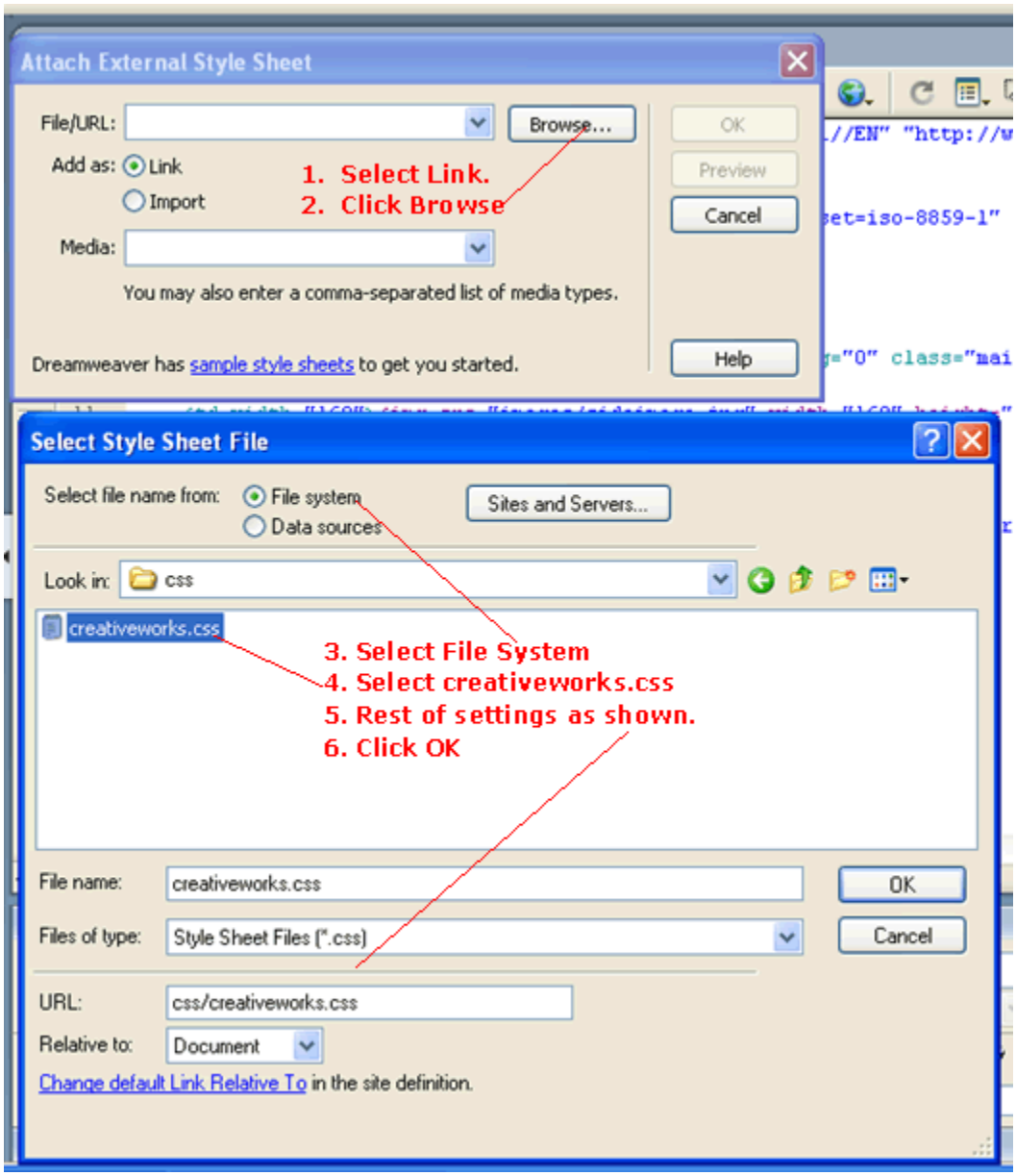


Now go back to the code of the **index.html** file. Delete the internal style sheet so that it looks like ..





We are now going to attach the new external style sheet to **index.html**. With **index.html** as the active document, click the **Attach Style Sheet** icon. When the **Attach External Style Sheet** dialog comes up, follow these steps...



And finally, click the OK on the **Attach External Style Sheet** dialog.

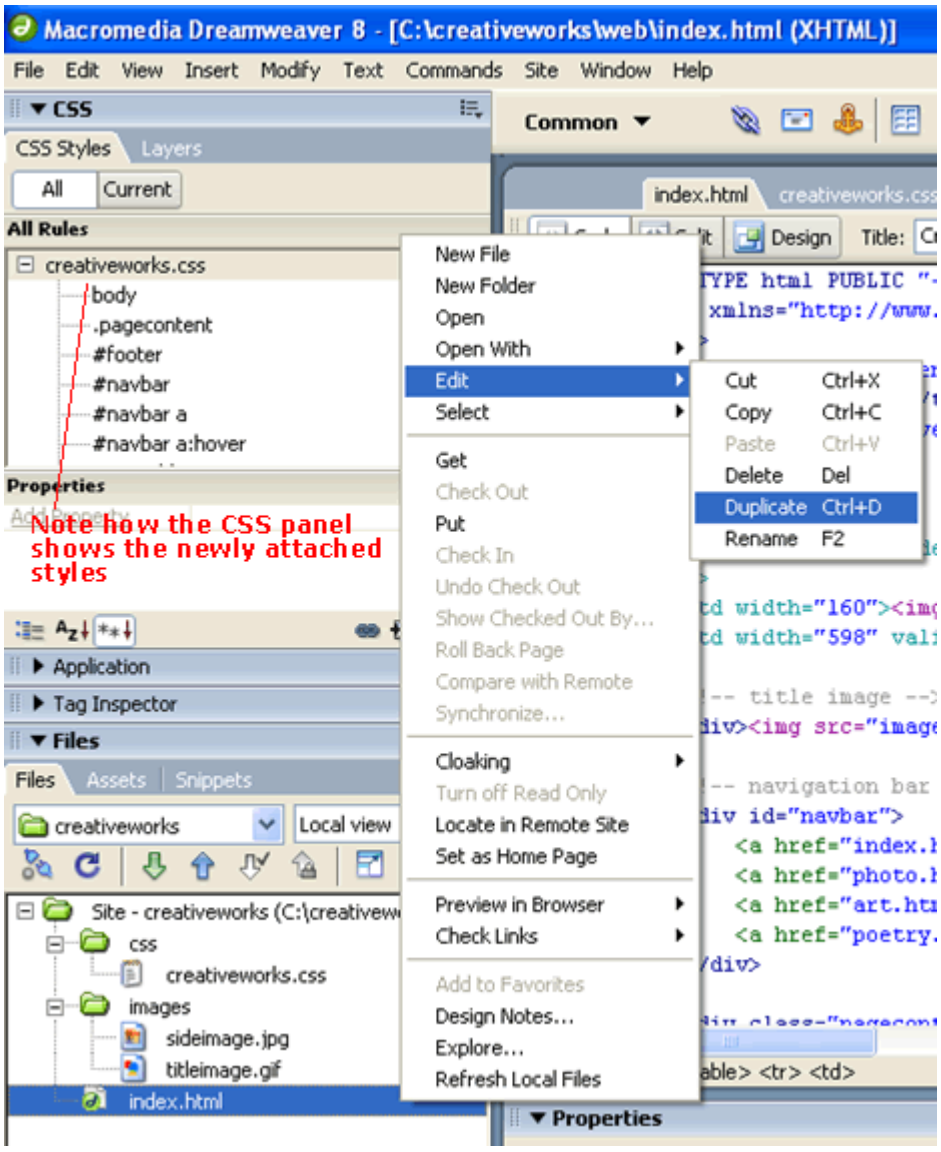
In the code view of **index.html**, see that Dreamweaver has added an **<link>** tag that references our external style sheet

creativeworks.css.

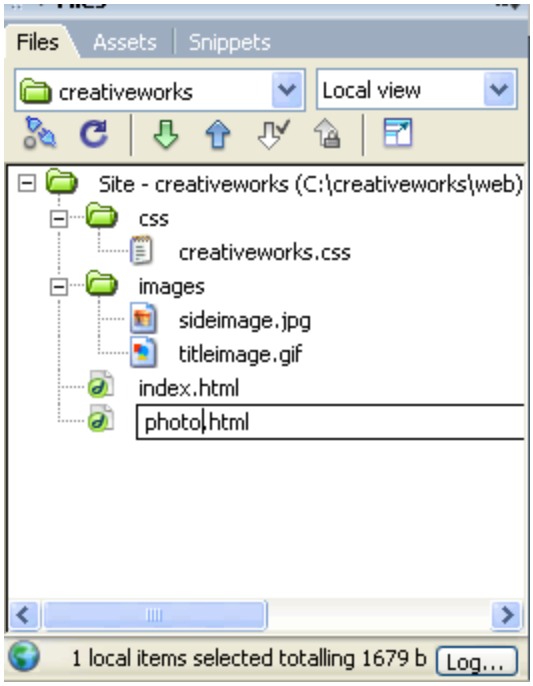
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Creative Works</title>
<link href="css/creativeworks.css" rel="stylesheet" type="text/css" />
</head>
```

Test and see that our page works just as before. Only now we have separated the presentation (the CSS styles) from the content (the HTML file) -- which is a good thing. Because now we can clone **index.html** into the other three pages.

Duplicating the Pages



Select the **index.html** file in the **Files** panel and right-click to select **Duplicate** as shown above. Then rename the duplicated file to be **photo.html**.



Do the same to obtain **art.html** and **poetry.html**.

Your Turn to Add Content

Now just have fun with it and and your own content to these three new pages.

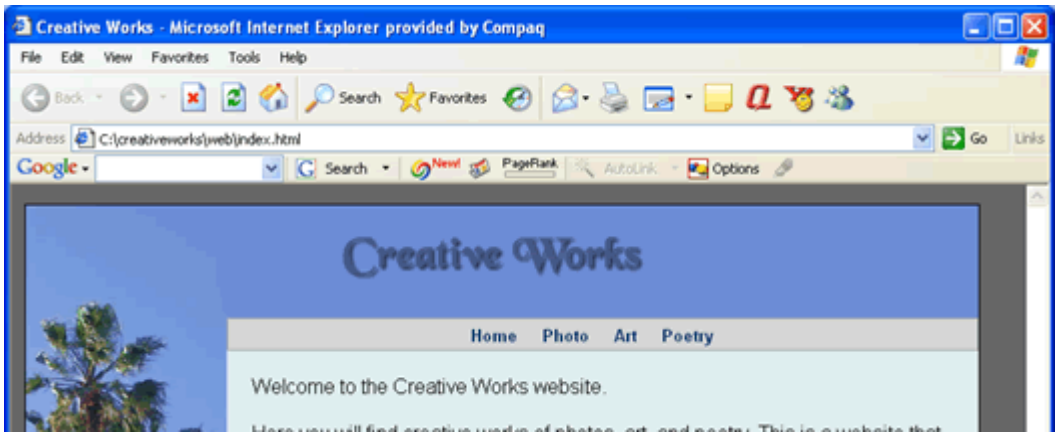
Here is what I have... **View Live Code**

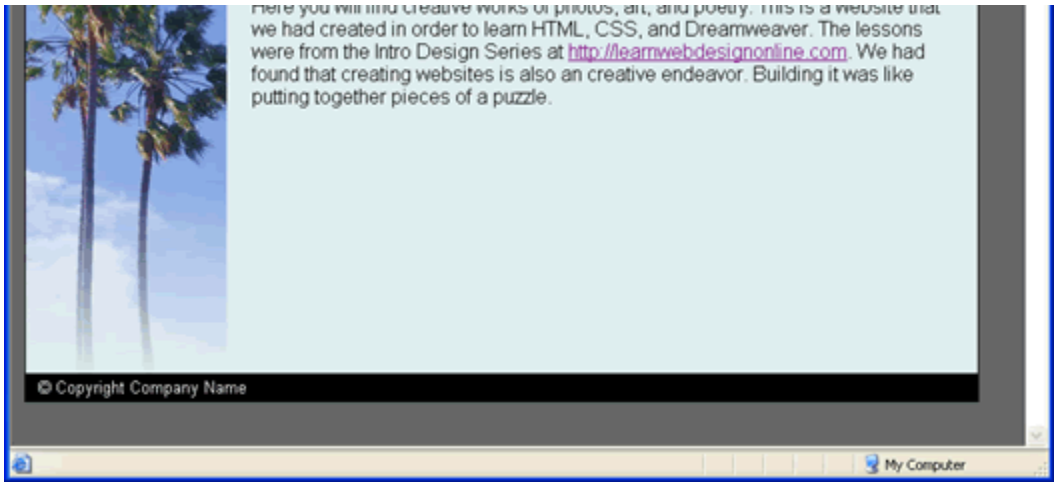
Download: **code6.zip**

Notice that the "Photo" page is a bit longer than the other pages. The layout of the automatically expands appropriately depending on the content of the page.

Centering Page Content

Currently our page is fixed to the left side of the browser window regardless of how wide the user expands his/her window. The final step is to make the page content be horizontally centered across the browser regardless of the user's browser width.

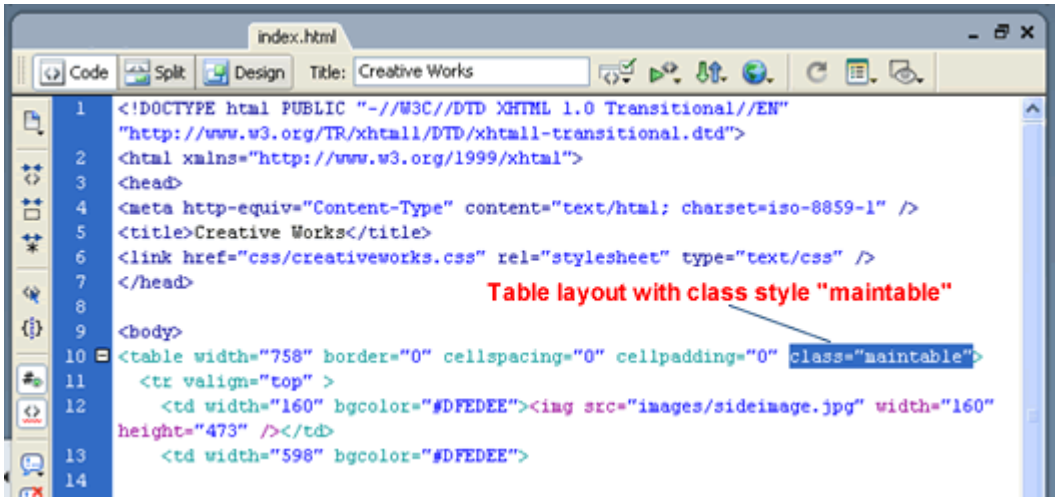




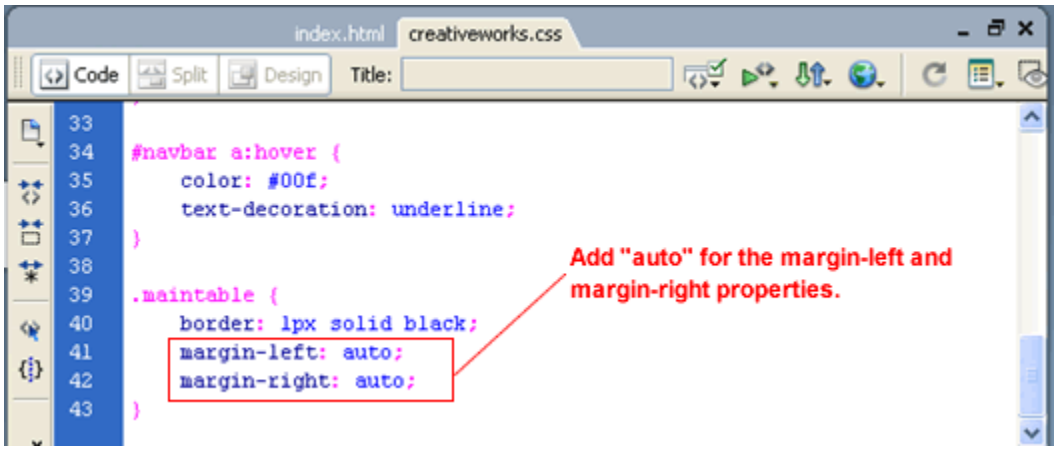
We will now alter the CSS rules such that the pages are always horizontally centered regardless of how the user resizes the browser.

See What Class Controls Our Main Table

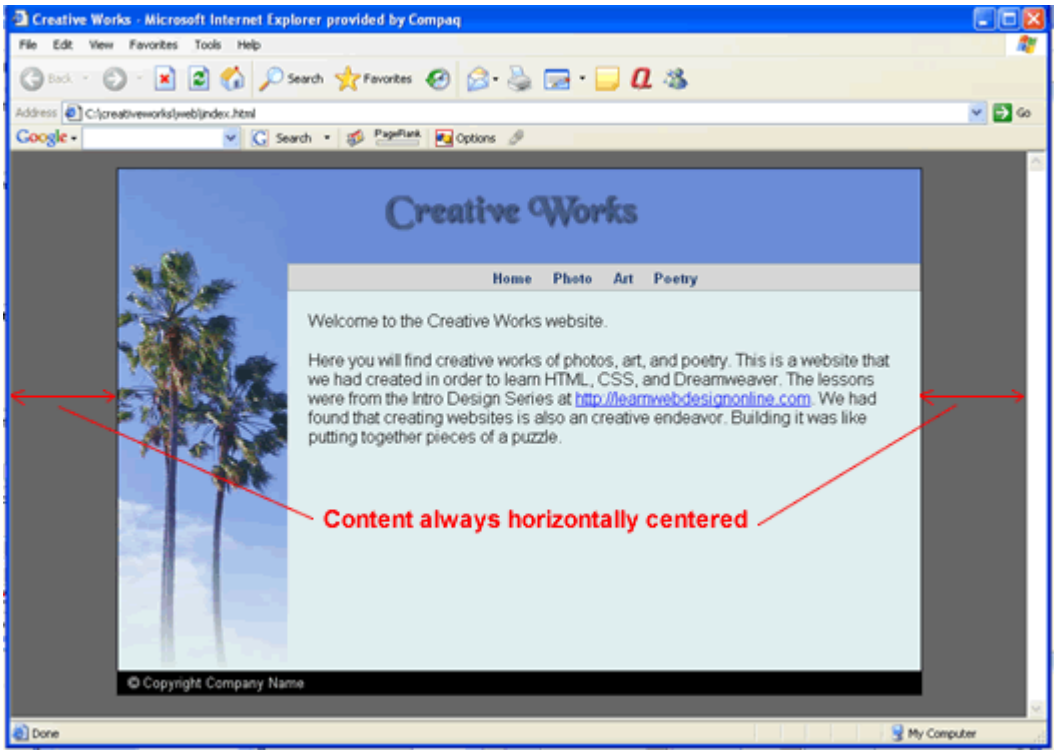
If you open up the code view of **index.html**, you will see that our main layout table is controlled by class style called **maintable**.



We want to alter the CSS style rule for this class. So we open up **creativeworks.css** and change that rule to ...



By adding the value **auto** to the **margin-left** and **margin-right** properties for that table. The browser will automatically adjust the left and right margins of the table so the table is horizontally centered. Note that for this technique to work, the **table** tag must have a fixed width. Looking at the above code, we see that indeed that table has a **width="758"** attribute.



Test this out in both Internet Explorer and Firefox and you see that it works. In addition, it works for the **photo.html**, **art.html**, and **poetry.html** pages as well since these pages also references the same **.maintable** CSS rule.

View Live Code
Download: **code7.zip**

Conclusion

This concludes the lessons in the Introductory Web Design Series where we built a website from start to finish. In the process, we learned the foundations of HTML and CSS and learned to use Dreamweaver and Fireworks. Hope you have enjoyed the lessons.

To continue your learning into more advanced topics, visits **LearnWebDesignOnline.com**. If you had found any errors and typos, please inform us through the website so that we can make improvements in future versions.