

---

# **ACME Dreamweaver CS6 (+5.5 & 5) Basics**

# ACME Dreamweaver CS6 (&5.5 & 5) Basics

These tutorials are designed to get you through the basics of making web sites in Dreamweaver with the most flexibility and the least amount of work. Think of them as supplying you with a toolbox full of tools to start making good websites.

They are deceptively packed with information, and they are designed to also act as reference after you are finished completing them. Web pages across the web have many similarities, and the information here is designed to cover many of these similarities.

The fancy stuff is not covered here—you are on your own for those techniques. But many fancy things are not too difficult to learn if you have the firm foundation that these lessons provide. But as you learn more, keep in mind that fancier is not always better than clean and simple.

It is very important that you try to somewhat understand what you are doing in these tutorials, and I would recommend doing each at least twice, the second time trying not to look at the instructions. Tutorials not done this way have a tendency to go in one ear and out the other.

If you have problems with doing the tutorials, just go back as many steps as needed and try again. Terms in dark red can be defined by touching the word.

These tutorials are designed to be done on a Macintosh, and assume a basic familiarity with the Mac, although you could certainly do them on a Windows computer without trouble. Make sure you update your version of Dreamweaver. The first versions of CS5 had a bug (feature?) which will cause problems in the layout chapter.

The illustrations in these tutorials are from CS6, but the CS5.5 and CS5.5 panels and buttons covered in this book are functionally exactly the same.

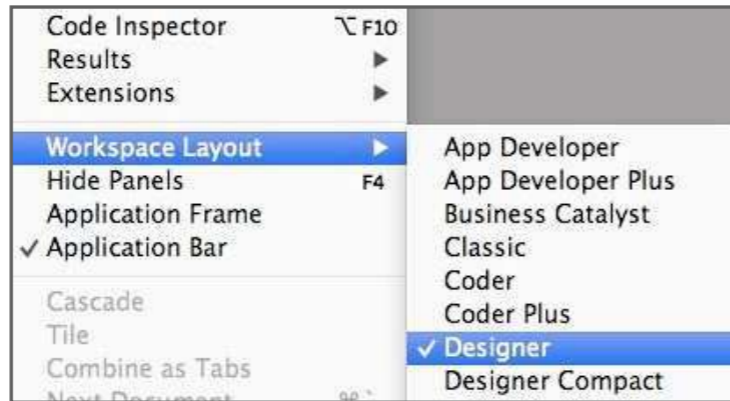
Finally, a disclaimer and a brag here: I am not an expert at programming for the web. I see this as a good thing for you. The terms and methods I present are as simple as they can be while still giving you a basic understanding of how Dreamweaver works.

# Making simple pages and links

*A web page is simply a bunch of words. In a browser, such as Safari, you can go to the menu View, then click View Source, and you can see this 'source code'. The browser's job is to take all this source code and make it into something more visual. Much of the source code in a web page tells the browser how to do this.*

*This is very different from just about any other application where the 'source code' (the stuff that tells the application how to display it) is totally hidden from the person using that application.*

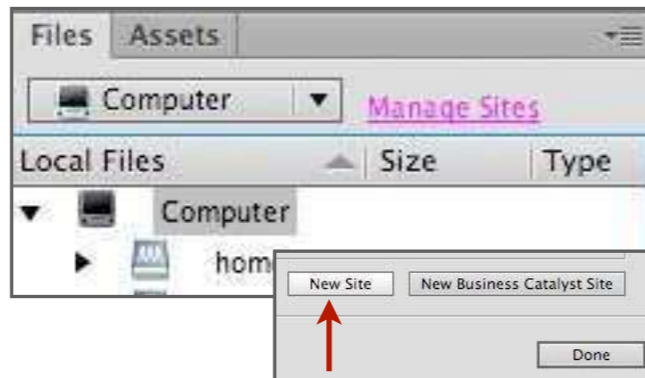
*In this tutorial you will be making a very simple website, and I will have you display the code on the page just to get used to it. You don't ever really need to know what all of it means, but eventually you will have to deal with it on some level.*



Open Dreamweaver. If the welcome screen appears you can disregard it. Go to *Window>Workspace Layout>Designer* (go to the menu *Window*, then choose *Workspace Layout*, and then choose *Designer*). This will get us in sync so that my examples look somewhat like what you will get. If you have moved windows around in the Designer layout, you can reset it to the defaults by choosing *Reset 'Designer'* in the same menu.

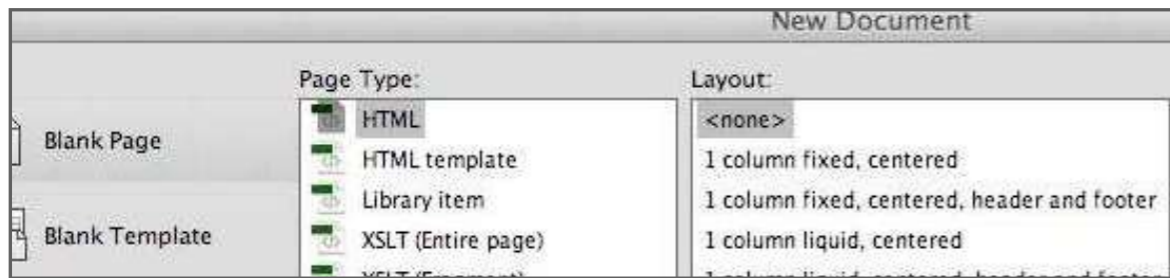
Before we get started making something we have to define a site (somewhere you want to put all your stuff ). Dreamweaver likes to keep all your files straight so that your life is easier. It will come in handy later.

In the **Files Panel** (on the right side of the screen), click *Manage Sites*. Click *New Site* (*New...* in CS5&5.5) to make a new site.



Name your site with something like your name, and specify a location by clicking on the little folder icon. Since this is temporary, you might want to make a folder (I called mine *Web Site Test*) on your desktop to contain the site. All of the information for your site will be in this folder. Click *Save* and then *Done*. In CS6 you may also want to go to *Window>Application Frame*. This will disable the floating window and give you much more screen to work with.





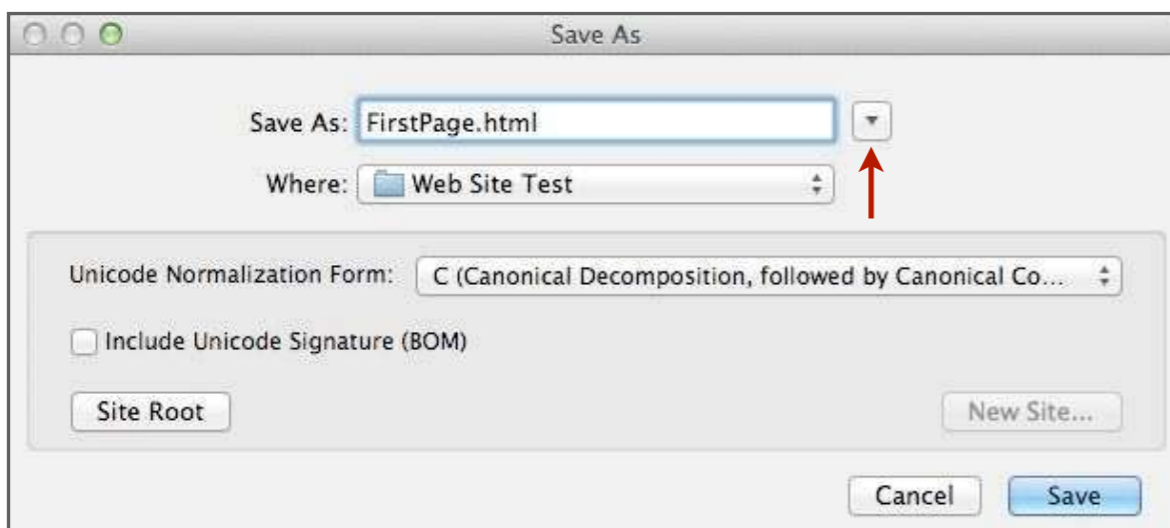
Now we are ready to make a page. Go to the menu *File>New*. There are a lot of options here, but the most useful is the first and simplest—choose *blank page*, *HTML*, and *<none>*.

When the new window comes up, push the *Split* button near the top left of the window to go to **Split View**. Now the window on the left is the **source code**, and the window on the right is the design. Don't get intimidated by this **code**. We are going to have to keep track of it later (sorry about that), but you really don't have to know what everything means.



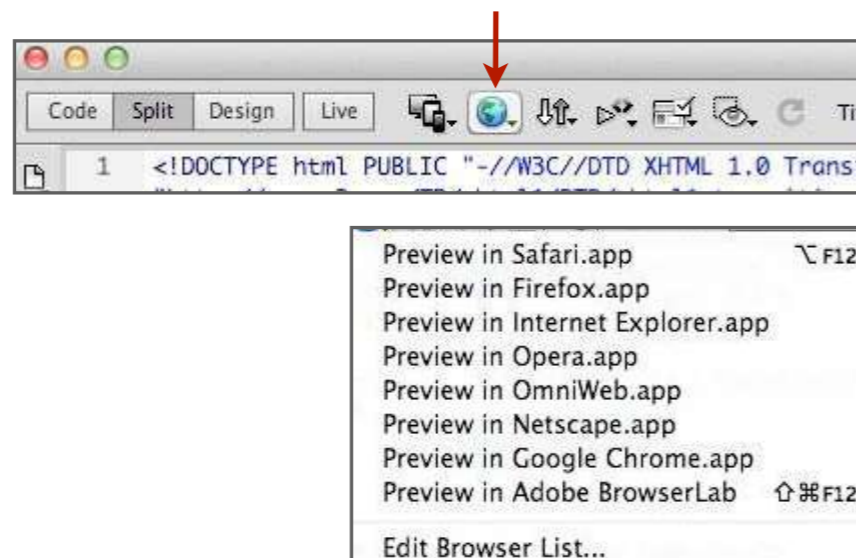
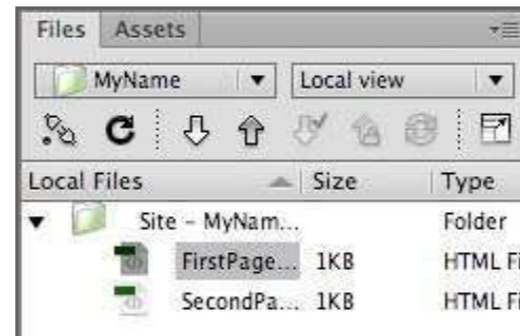
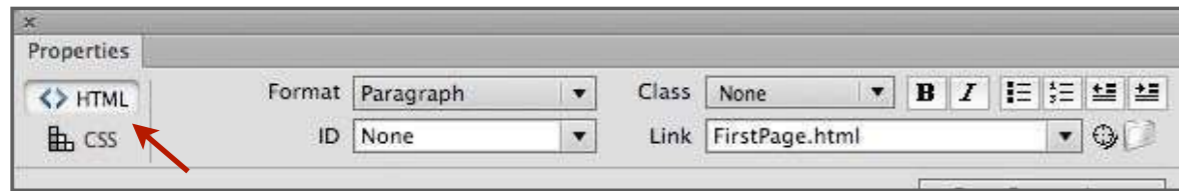
In the **Design View** (on the right) type something like *the first page*, hit the return key, and type something like *go to the second page*. That will be our link to the second page pretty soon.

While you are at it, name your page in the blank box at the top of the window labeled *Title*:. Not very important at this point, just a good habit to title all your pages.



Save your page into the folder you defined before with your name (*File>Save*). Call it something like *FirstPage*. Try to get out of the habit of using spaces (or punctuation) in a filename. You can use *camel-case* like in this example, or use underscores to separate words. If you need to navigate to your web folder, click the small discloser triangle to the right of the filename.





Notice that in the **Properties Panel** at the bottom of the screen the **Link** type box has the name of your first page in it. If you don't see the box, click on the **HTML** button on the left of the panel. You could have just typed it in here, and there are also many other ways to make a **link**.

Save your second page, close the window, and if you closed the first page, open it by double-clicking on it in the **Files Panel** on the lower right of the screen.

Make a **link** for the *Go to the second Page* type the same way you did for the second page, but of course this time specify the second page as the one to link to.

Very important—now save your file since we are going to be testing it.

At the top of the window, find the little globe and choose a **browser** to **preview** your file in. After it opens, you should be able to go back and forth between your two pages. If not, then try this short tutorial again.

What you have just done in this tutorial is make your first website. Here is how to prove that: Quit or hide Dreamweaver, open the folder that you made, and open one of the two files in a web browser (just drag the file onto the Safari icon or double-click the file).

You could take this folder and save it to another drive, open it on another computer, or email it to someone. You could even upload it to a web server, and the whole world could see your creation. As long as the files stay in the same places within that folder, it will work fine.

There is one last step in this tutorial. Open Dreamweaver, and do the whole thing over again. Maybe this time without looking too much at these instructions. Look at the **code**. Try to figure out what is going on.

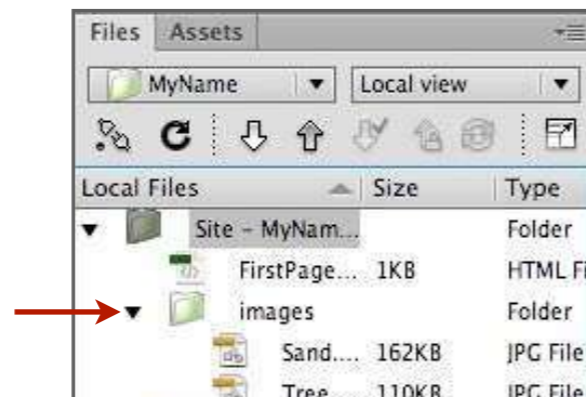
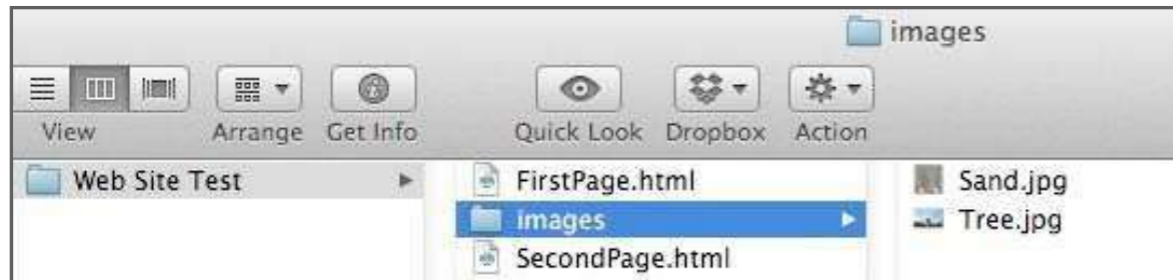
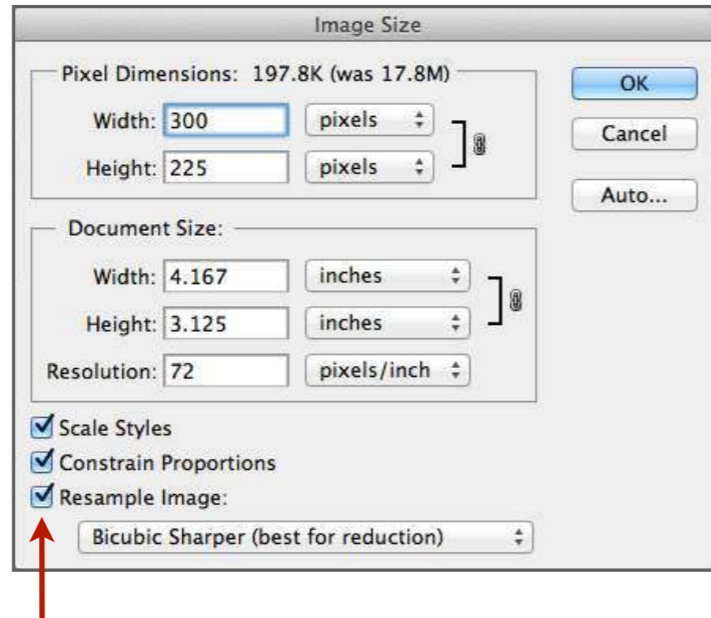
You may want to save your work from this tutorial so you can expand on it in Chapter 2.



# Structure, Images, & Text

*In the last tutorial you learned how to make a web site, and if you just want words and don't care how it looks, you learned pretty much all you need to know.*

*In this tutorial we will be making a web page and adding photographs and text formatting. More information choices, nicer to read.*



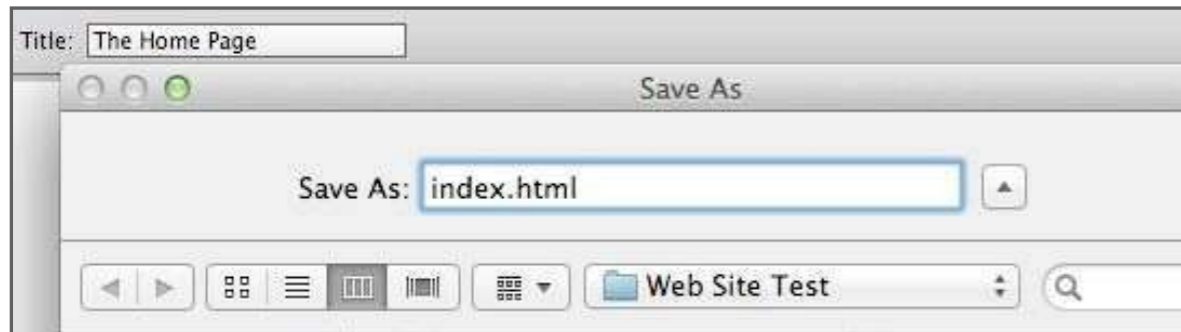
Before you start, go to some web sites and grab at least two JPG (JPEG) photographs. Just drag them off the browser page onto the desktop. Don't get any that are more than 300 pixels wide (very important). Google shows you the **pixel dimensions** when you get them that way. You can also limit your Google search to images which are exactly 300 pixels wide by going to Google's size options on the left side of the image search window. Don't just grab the thumbnail—click on it, then drag the resulting image. Or, get any images and size them to 300 pixels wide in Photoshop (*Image>Image Size* as in the illustration).

Next we need to start doing things right. The folder that you made in the last tutorial only has two things in it, so it is no problem keeping those things straight. In actual practice, there may be dozen, hundreds, or thousands of things in this folder, so we better start organizing it.

Make a folder named **images** in your **web root folder**, which is the folder with your two files in it. Images is not an arbitrary name, but one that convention dictates. Put your photographs from the web in this folder.

Open Dreamweaver, and look at the **Files Panel**—the images folder you made should be in there and look something like the example shown. Click on the little disclosure triangle to show the contents of your images folder.

Make new file in Dreamweaver the same way you did in the last tutorial (*HTML*, <none>). Put any title in the title box at the top (I titled the page *The Home Page*).



Save it as *index.html* in your **web root folder**. This filename is not arbitrary. If your site was on-line, visitors would automatically be directed to the page called “index”. While on the subject, filename extensions (like .html) are very important. In the web world the right extensions are always needed. It is a good thing applications put them on automatically. Just don’t take them off.

The reason we save the file before we have even done anything to it is that when we add photographs, your file needs to know where they are relative to its own location since it only references images and never embeds them. Saving the file gives it a location to be relative to (think about it a second).

This is a good point to take a quick look at the **source code** (you should still have the **Split View** turned on with the button on the upper left of the window).

The first three lines of code you can disregard – these are just telling the browser what type of language it is speaking.

Look at the line that starts with `<title>`. `<title>` is an **opening tag**. You can tell it is a **tag** because of those pointy brackets that are on either side of tags. Following this is what you named your web page, and following that is the **closing tag** `</title>` (often just like the opening tag but with the addition of a forward slash).



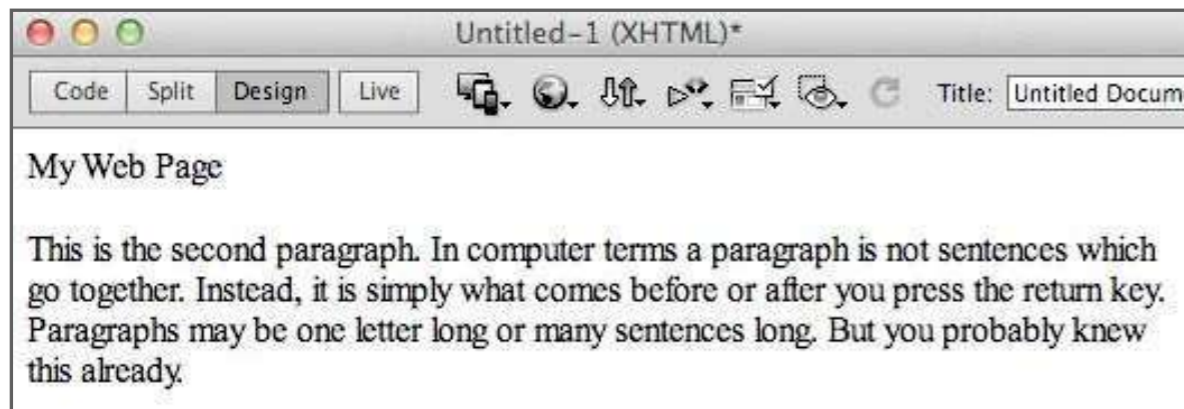
These **tags** tell the browser what you want to do with the text (or whatever else) between them. In this case it is telling the **browser** to put the words *The Home Page* on the top bar of the web page (when viewed in a **browser**).

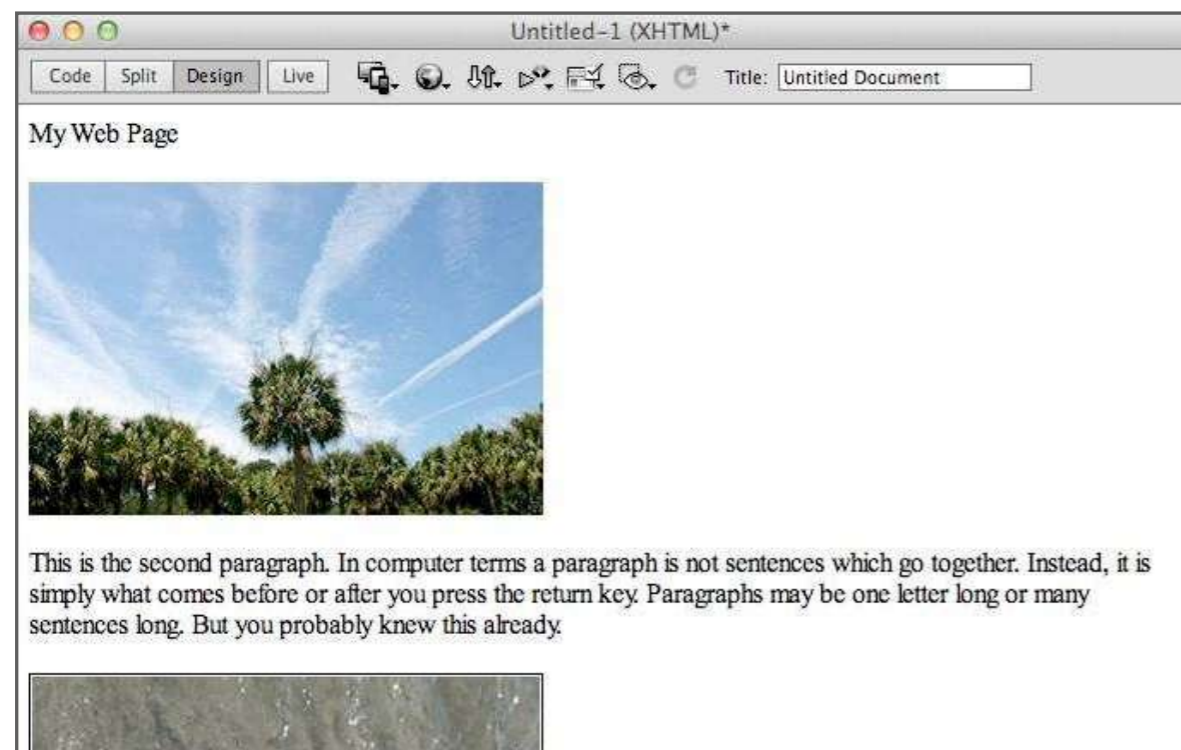
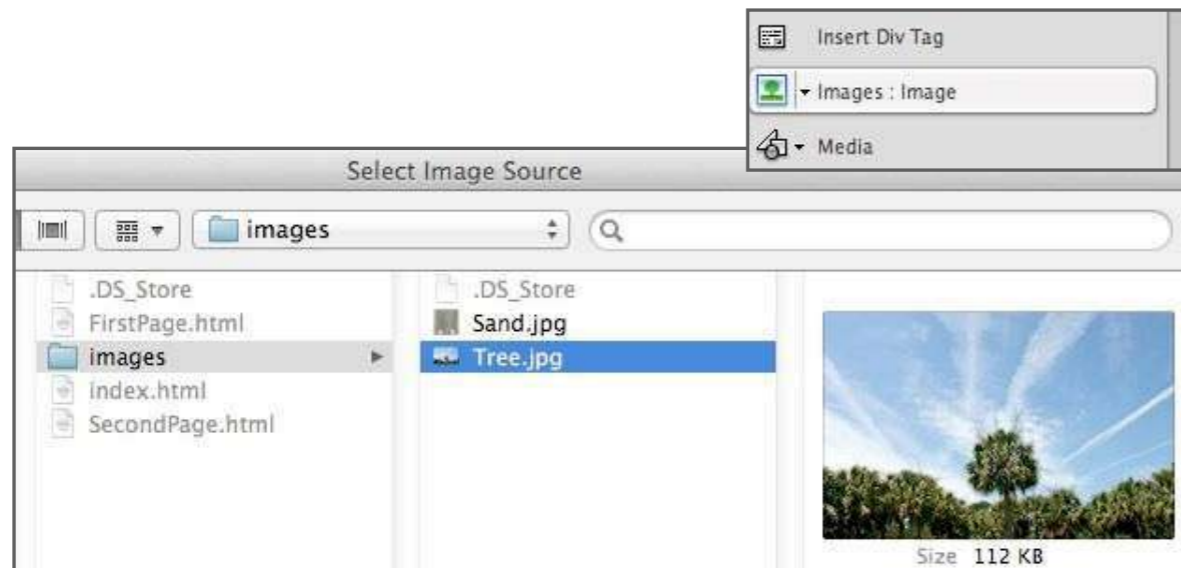
You can change the title of the page right there in the **source code**, and it will be updated in the little box you actually typed it in (you need to give a click in the **Design View** to update the code first). As a matter of fact, you could have typed in the **code** and the text without ever going to that little box (but of course it was easier to type it in the little box and have Dreamweaver put in the tags for you). Some people still do web pages in text editors, and just type it all in. That seems too difficult to me.

This is the way Dreamweaver works—in the **Design View** you type, and the program puts in **tags** and other things for you. The reason you need to deal with the **code** at all is that Dreamweaver will sometimes do it wrong. Not the fault of Dreamweaver, just the way that **HTML code** works.

That is enough code for now, so let's make things simpler to look at by going to **Design View** (the button on the top left of the window).

Just like in a word processing program, a **paragraph** is defined as what happens before or after you hit the return key. Make a very short **paragraph** and a longer **paragraph** in the **design view** window. The short paragraph in the example below is simply *My Web Page*. Do not hit the return key in your longer paragraph—just let the type wrap at the end of the window.





Now might be a good time to check the spelling, since I suspect there is a mistake. In Dreamweaver *Check Spelling* is **enigmatically** placed in the *Commands* menu.

Add an image (photograph) now. With your cursor (insertion bar) at the end of the first paragraph (the short one), hit return to make more space between your **paragraphs**. In the **Insert Panel**, click on *Images : Image* (when you first choose this it will present a drop-down menu where you select *image*) and choose a photograph from the ones in your image folder. It will ask if you want an alternate text for the image. For now we can click through this box (just hit *OK*).

Go ahead and add the other photograph after the second **paragraph**. Remember to put the cursor at the end of the **paragraph** and hit return before you insert the image.

Notice that the images came into your web page the same size as they were in the web page you took them from. You could change the size of images in Dreamweaver, but **DON'T! EVER!** They don't display very well and take more memory (which translates to download speed). You can always size them in Photoshop—the size they are in Photoshop at 100% is the same size they will display in Dreamweaver or in a **web browser**. To resize an image in Photoshop, just go to *Image>Image Size*, check *Resample* and *Constrain Proportions*, and enter either the width or the height in pixels.

While on the subject of good habits, it probably about time you save your file. Just hit the *Command & S* keys if you don't know that already.

Before the end of this tutorial we should format some of the text and layout. This will turn out to be background information since we will later be doing it a bit differently.

Highlight your first **paragraph** (the short one) by dragging your cursor across it just like in a word processing program.

In the **Properties Panel**, make sure that the **HTML Button** is pressed on the left. Change the *Format* button to *Heading 1*.

What this does is put a **tag** around your words which tells your **browser** to make the type bigger in a pre-defined way. In this instance `<h1>` tells the **browser** to start the bigger letters and `</h1>` tells it to stop. Go to **Split View** and check it out in the **source code**. If you still have the **paragraph** highlighted Dreamweaver will also highlight it in the **source code**.



```
<body>
<h1>My Web Page</h1>
<p>This is the second paragraph. In computer terms a p
go together. Instead, it is simply what comes before o
Paragraphs may be one letter long or many sentences l
already.</p>
```



```
</head>
<body>
<h1 align="center">My Web Page</h1>
<p>
<p>This is the second paragraph. In computer terms a par
go together. Instead, it is simply what comes before or
Paragraphs may be one letter long or many sentences lon
already.</p>
```

Let's center this paragraph now. With the paragraph still selected, go to the menu *Format>Align>Center*. You can also see how to move the type back to the left in this menu.

If you look at the **source code** you will see that Dreamweaver has changed the **tag** to specify that the type be bigger and centered. (`h1 align="center"`) The emphasis on the **source code** in these tutorials is necessary—if it wasn't, I wouldn't include it. In the end it does make things easier. On the good side, *almost* nowhere will I ask you to actually write code!

Now you know enough to do an extremist wingnut website (the standards are low with these). Play around with the techniques you have learned so far. Make a **link** to one of the pages you did in the first tutorial, center your images (just click on them to highlight them and center them the same as you did text). Add some more images. **Preview** the whole thing in a web browser (globe at the top of the window) to make sure everything is working right.

You might also experiment with other ways of doing things so that they are more convenient for you. Try dragging an image file directly to the page from the **Files Panel** (not from the finder). Or try to define a **link** to another page by highlighting a word or an image and then dragging that cross hairs icon (in the **Properties Panel** [html]) to the file in the **Files Panel** you want to **link** to.

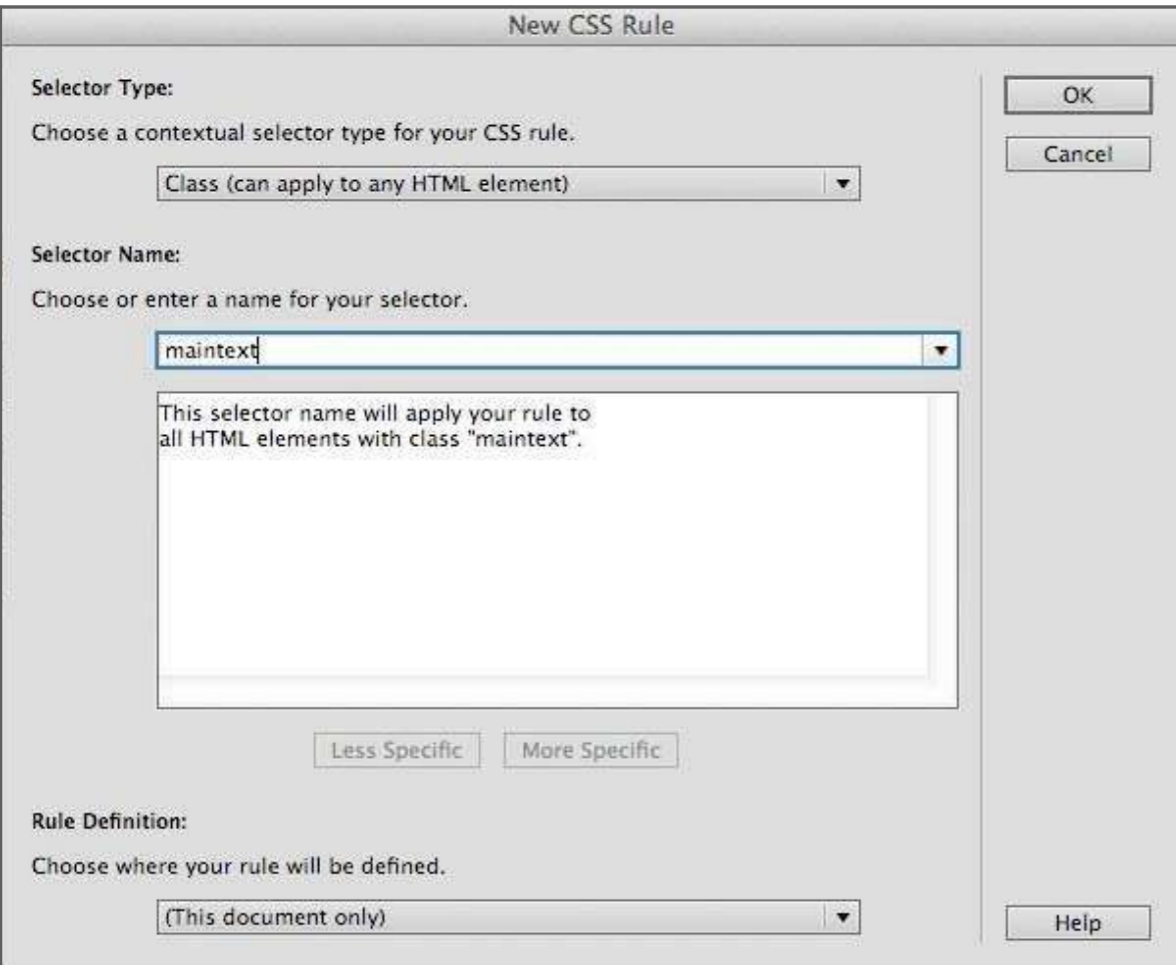
If you would like you can save your files for the next tutorial, or just redo some files like this for practice.

# CSS for Type Formatting

*CSS stands for Cascading Style Sheets. This name will make a little bit more sense after you know how to use them. We will be using them for two things. The first is to format text. The second is to lay out a web page, which will be covered in the next tutorial.*

*All web designers use CSS in preference to formatting by other methods (like the `h1` tag you used in the last tutorial). It is far more powerful and flexible. But it is a bit more complicated. Try to get your head around it in this tutorial, since we will build upon the knowledge in the next chapter.*





If you didn't save your page from the last tutorial, or messed it up quite a bit, then make a new one which approximates it. Now go to **Design View** to start.

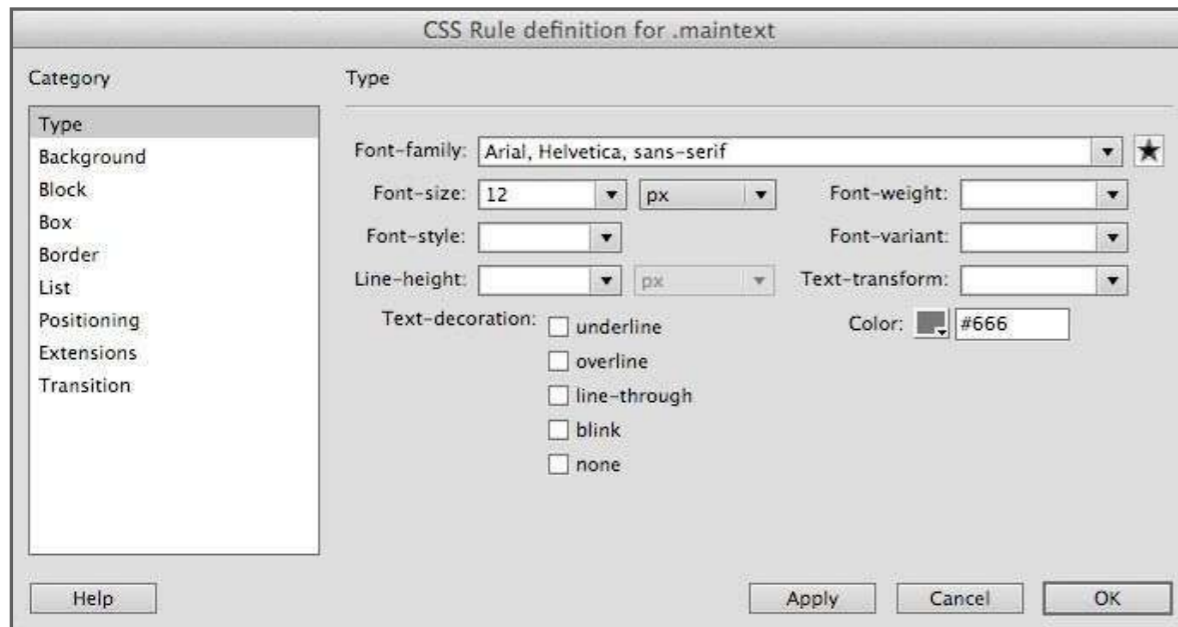
Highlight the second (longer) **paragraph** and in the **Properties Panel**, choose the **CSS** version of the panel on the far left. It is unconventional for an application to have two modes to a panel, so try to remember there are two modes in this one.

Make sure the *Targeted Rule* box is set for *New CSS Rule*, and click the *Edit Rule* button.

The box that follows deserves some special attention. The selector type of your **rule** should be *Class (can apply to any HTML element)*. There are four different types of **CSS rules**, but only three of which we will need to use. This is one of those three.

The name for your selector is up to you. Since I am going to use this rule for my main text, I will call it *maintext*. Convention is to apply names with no spacing, punctuation, or upper-case letters.

One of the very cool things is that you can define rules that you can use in multiple pages, and make changes across multiple pages in just one place. But here we don't need to, so we are going to put the rule within our web page (*This document only* at the bottom of the box). After you have the three settings done, click *OK*.



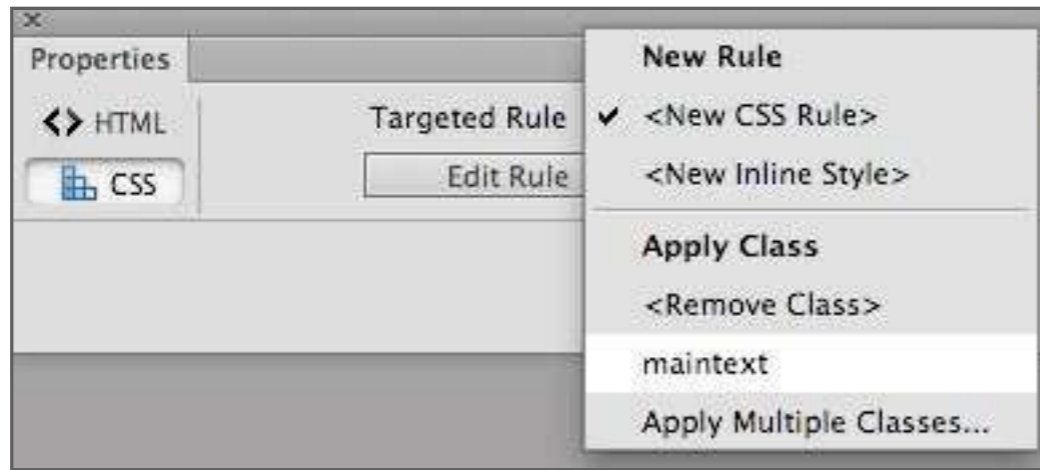
Now that we have told Dreamweaver what kind of **CSS rule** we want and where to put it, we can determine what the **rule** will do to the text we have selected. You can make your own choices. In the example I have specified (with the pull-down button to the right of the box) that I want a sans-serif font, and if the viewing computer has Arial installed it will use it. If there is no Arial, then it will use Helvetica. Web pages don't have their own fonts, so they must use the fonts available on the computer you view the pages on.

I have also specified that the text be 12 pixels high and grey. Go ahead and also look at the options you also have in the *Block Category* of this requester box. You can specify things like tracking and leading. **CSS** formatting gives you much better control than **HTML** formatting (the way we did the *h1* tag in the last chapter).

After you click *OK*, you should see the type you highlighted change to conform to the **rule**. Now is a good time to see what all of that work did to your **source code**. Near the top of your **code** (in **Split View**) is your rule with a period before the name and the definition in curly brackets (`{ }`). That is the **rule** definition. See if you can find out where in the **code** it is applied to your paragraph (look for *main-text* elsewhere in the **code**).

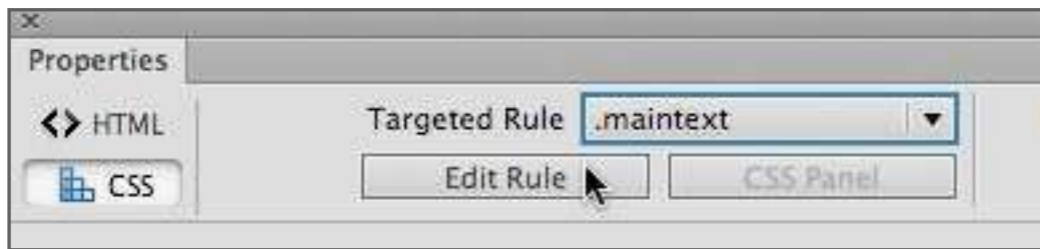
Now is a good time to point out that things that display on your webpage are between the tags `<body>` and `</body>`. Since the definition for your style doesn't display on the page, but is only referred to by something on the page, it is before the `<body>` tag.



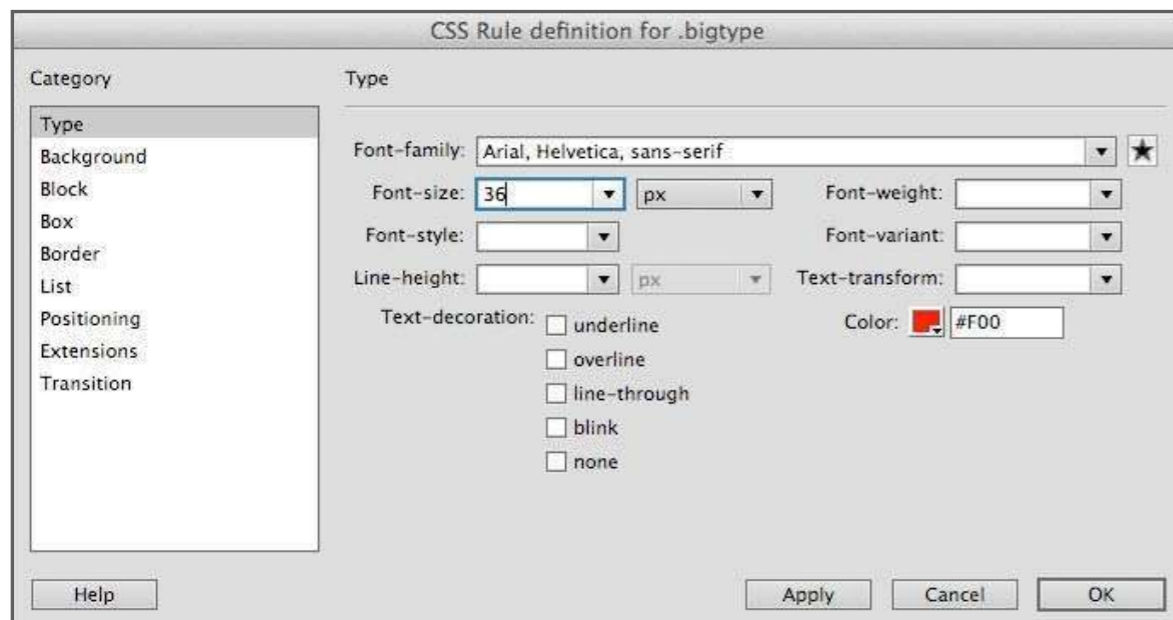


Go ahead and make a third **paragraph** under your second image, highlight it, and apply the **rule** you just made to it. The choice of applying the *maintext* (or whatever you called it) style will be in the *Targeted Rule* selector in the **Properties** panel.

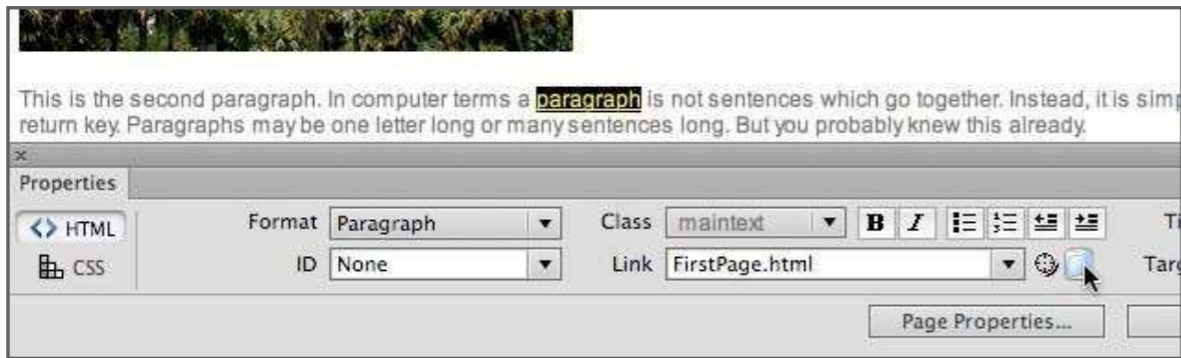
If you know how to use paragraph styles in a word processing application or in InDesign this should be somewhat familiar.



If you want to edit your **CSS rule**, it is easy. Put your cursor anywhere in the text that has the **rule** applied to it. In the **Properties Panel** the *Targeted Rule* will be *.maintext* (or whatever you called it). Just click on the *Edit Rule* button and you can make any changes you would like. Try at least changing the color or the size of the type.



To reinforce this a bit, make another **rule**. This time highlight your first **paragraph** (the short one) and make a rule (flip back a couple of pages if you can't remember how) which changes the color and size of the text like in the first style you made. I called this style *bigtype* since it is what I might use for things like titles. Things on your web page should be looking better. If not, edit your **rules** to make them look better.

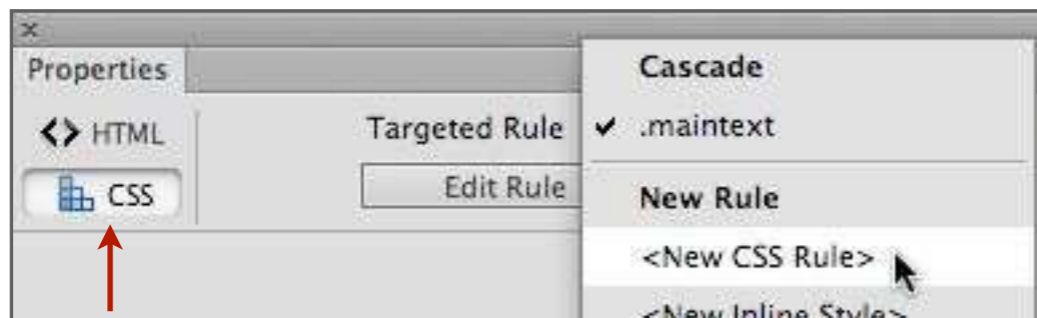


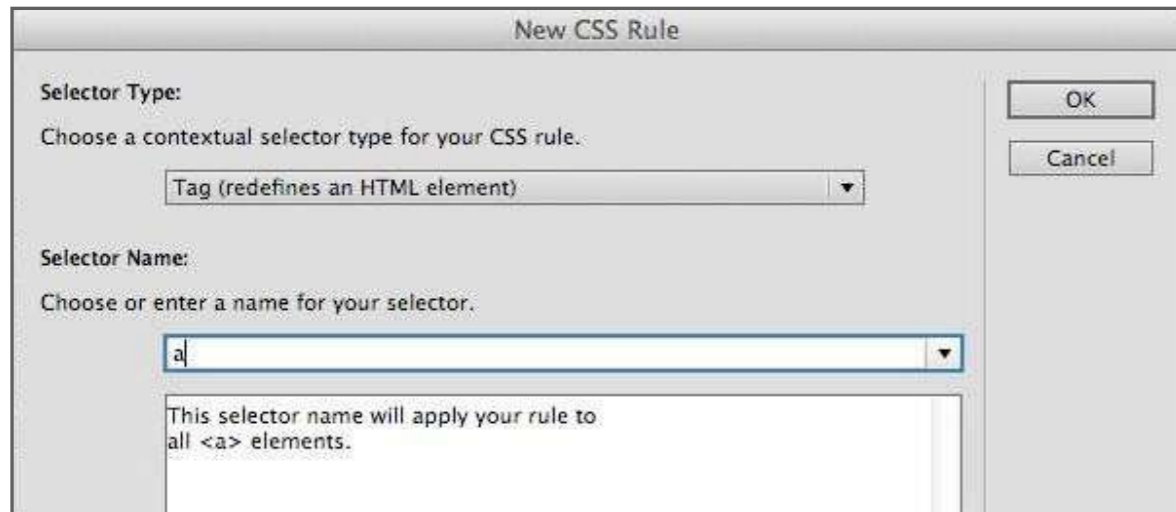
Let's go back to **links**. We are going to make a word in one of your **paragraphs** a link, but I need to show you a bit different (and better) way of doing it. Highlight a word in one of your longer paragraphs and click on the folder icon to the right of the **Link** box in the Properties Panel (make sure the *HTML button* is activated). Choose one of the pages you made for the first tutorial as the link and click OK.

Click anywhere in the **Design View** to deselect the word you highlighted. It should now be blue and underlined, because that is the way links are made to look, and this overrides the **CSS rule** you made (but notice that the type size and style is the same).

We are going to change the look of the **links** on the page, and we are going to do it by applying a **CSS style** to override the **HTML** convention that says links should be blue and underlined.

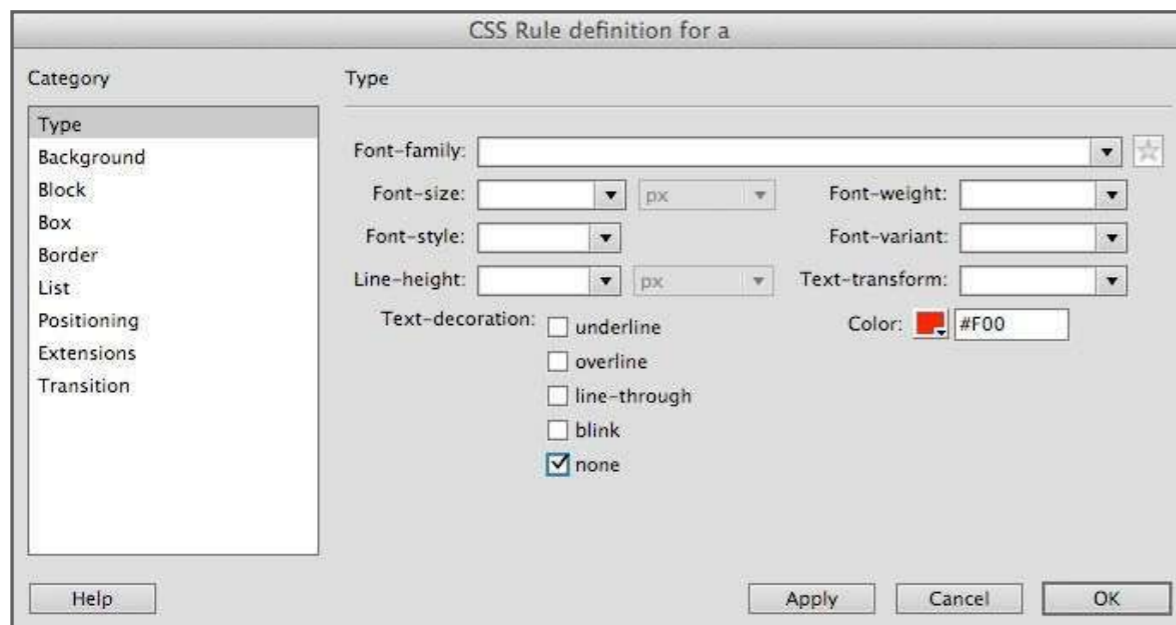
For this one your cursor should be outside the **paragraph**. In the **Properties Panel** (now with the *CSS button* pushed) select **<New CSS Rule>** and click the *Edit Rule* button.





The resulting requester box should be familiar from before, but this time we are going to choose *Tag (redefines an HTML element)* as the *selector type*. This is because we are not really going to define a **rule** from scratch, but edit an existing **tag** to make things look different. You should start to get an idea of why I am having you look at **code** every once in a while.

There are bunches of **tags** in **HTML**, but the one that defines a **link** is just 'a'. You can just type it in or choose it from the very top of the list when you click on the little triangle to the right of the selector name box. You may wonder why it is not called *link* (which is in fact another tag which we will not be using).



In the (by now familiar?) CSS edit box that comes up, choose some attributes that might lend themselves to calling out a **link**. I chose bold, red, and turned off the underline by clicking *none* for the *Text-decoration*.

Click *OK*, and you should now see your text specifications applied to your linking text. It will also be automatically applied to any other **links** you make on the page

You didn't need to specify the font in what you just did unless you want to change it from the font you specified in *maintext*. We have a **rule** (the *a rule*) inside another **rule** (*maintext*). Whatever you don't specify in the inside **rule** will be governed by the outside **CSS rule**. In this case, the text style is still governed by the style we specified for the paragraph containing the link.

You might see in the **code** that there is a **tag** which governs everything displayed on the page—it is called the **body tag**. We could also apply a **CSS rule** to this **tag**, and thereby have a **rule** within a **rule**. If nothing from the larger **rule** is changed by the **rule** within it, then it stays the same. Hence the *Cascading* in **Cascading Style Sheets**. We will see more cascades in the next tutorial.

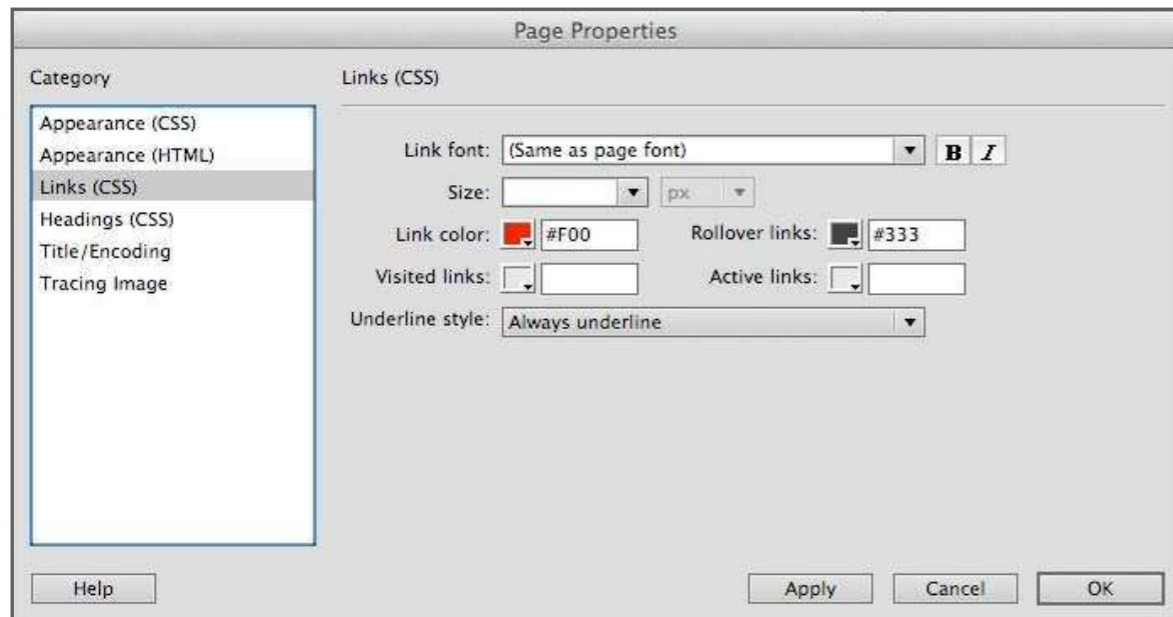
You can actually see the cascade of **tags** (**CSS styles** become tags). In **Design View**, put your cursor anywhere in a **link** you made and look at the bottom left of the window. You will see `<body>` first, since that is what governs the whole page, then you will see the *maintext* **tag** within it (applied to a **paragraph**, hence the *p*), and finally within that the **link** (*a*). More on this very soon.



```
6 <style type="text/css">
7 .maintext {
8     font-family: Arial, Helvetica, sans-serif;
9     font-size: 12px;
10    color: #666;
11 }
12 .bigtype {
13     font-family: Arial, Helvetica, sans-serif;
14     font-size: 36px;
15     color: #F00;
16 }
17 a {
18     color: #F00;
19     text-decoration: none;
20 }
21 </style></head>
22
23 <body>
24 <h1 class="bigtype">My Web Page</h1>
```

Now look at the **source code** in the **Code View Window**. You can also see that the **CSS rule** that changed our *a* **tag** doesn't have a period before it like the other **rules** do. This tells the **browser** that it is a **rule** which changes a **tag**, not a **rule** you made and happened to call *a*.

Now go back to **Design View**. We want to change our **links** so that when we roll over them, the color changes. This **rollover** is also called **hover**.



Go to the menu *Modify>Page Properties*. You could also find a button on the **Properties Panel** that will take you to the same place. One of the more confusing things about Dreamweaver is the multiple ways of getting the same thing done.

In the *Links (CSS) Category* on the left, add a different color to the *Rollover links*. In this case I made that color black. You could have also defined the red color and the bold style of **links** in this box instead of the way we did it. Press **OK**.

**Preview** your page in a **browser** (by clicking and holding the globe at the top of the window) you will see that your **links** change colors when your cursor hovers over them.

Instead of **previewing** in a **browser** to test things like this you can also click the **Live View** button at the top of the window next to the **Design View** button. It will not necessarily look the same as in a **browser**, but it should be close. Just remember to turn off **Live View** when you are done, since when it is on it changes the way Dreamweaver works.

In this tutorial we have bitten a lot off. It would be wise to go back through it and make sure you understand what we did. Experiment with things and try different ways to do things. We used the **Properties Panel** in this tutorial, but there is a **CSS Panel** on the right of the screen devoted to making and editing **CSS styles**. Try using it by clicking on one of your **CSS styles** and then clicking on the edit button (pencil icon) at the bottom of the panel. Save your file for the next tutorial.

# CSS for Layout

*You may have noticed that we have gotten absolutely nowhere in being able to change the layout of web pages so far. With this tutorial that will change.*

*In the past, things called tables were used for layout, and they were pretty simple. But they are not really used much anymore (that is a good thing overall), so layout had to wait until we got a bit of a handle on CSS styles.*

*We will now apply a layout to our page using CSS styles. I am going to use the same page as I was using, but you could use a new page populated by text and images (300 pixels wide or less).*

*A common mistake with beginning students is to design their web pages on a larger monitor than is common. Remember as you lay out your page that your Dreamweaver window may be bigger than a common browser window would be.*



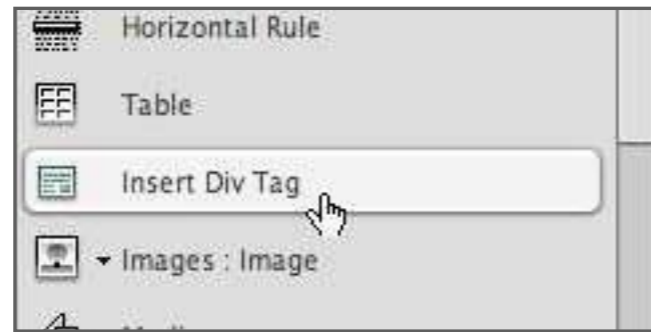
Web pages, unlike a physical page, have only two fixed borders—on the top and on the left. The bottom and right sides have no defined border. The bottom goes on for as long as you can scroll, and the right side will change according to how large your **browser** (or Dreamweaver) window is stretched to.

So, our first task is to make a defined right border. This is almost always done by making a wide column in the middle of the page that moves to center once the **browser** window gets wide enough. Look at a few websites—in most websites if your **browser** window is larger than a certain size then the information centers within it.

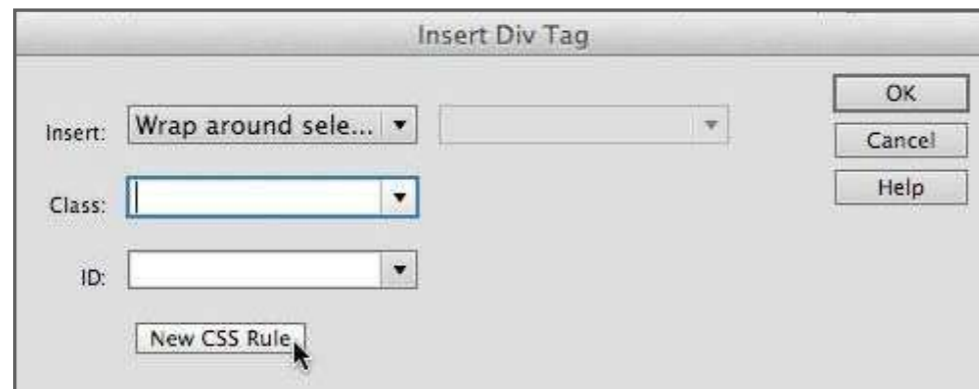
This centered container for the information is the first thing we will make using **CSS layout**. You would think that dragging your cursor over the whole page would select everything it, but it doesn't necessarily.

With your cursor anywhere in the **Design View**, go to the bottom left of the window and push `<body>`. This is actually a button which selects everything within the **body tags**. That is everything on the page. You can verify this by looking at the source code—everything between the body tags will be selected.

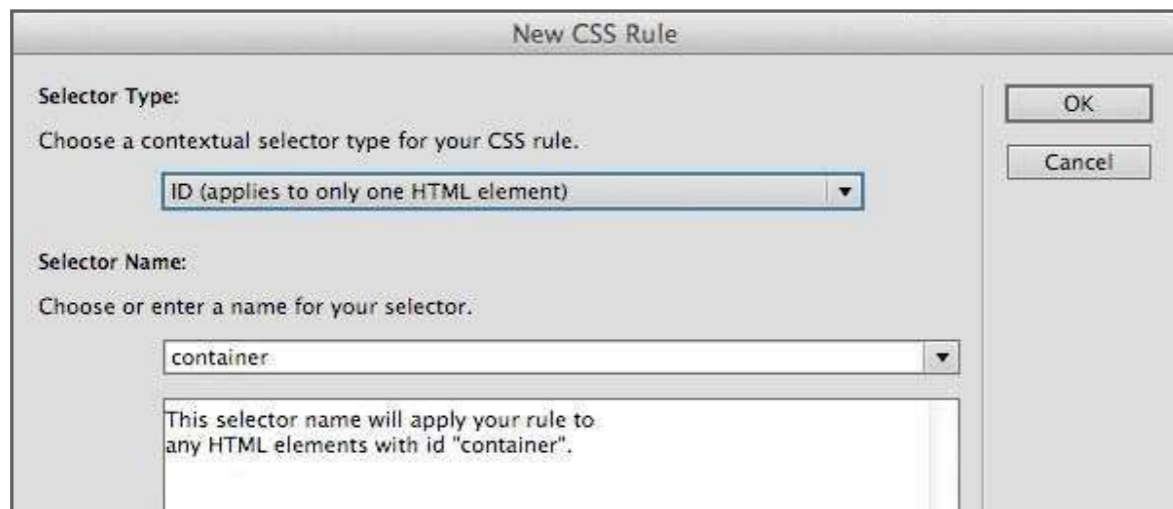




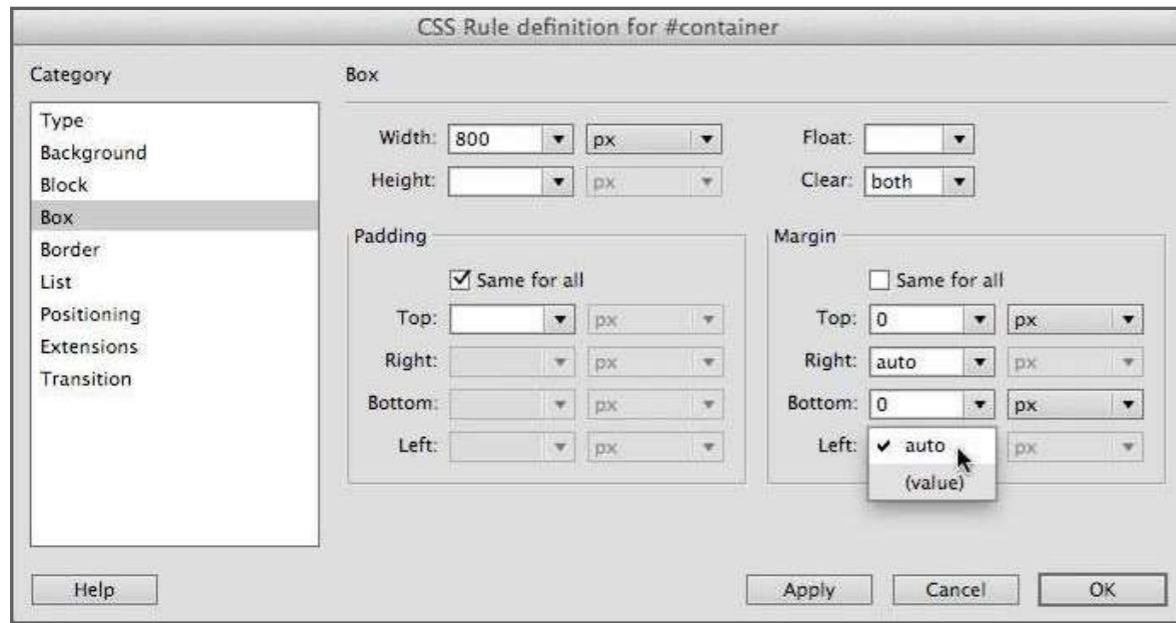
With the body button selected, go to the **Insert Panel** (on the right, remember?) and click the *Insert Div Tag* button. A div tag can be thought of as a layout tag.



In the resulting box you want to make sure to insert the **tag** around around everything, so select *Wrap around selection* (remember we selected everything). Now press the *New CSS Rule* button.



Use the *ID (applies to only one HTML element)* selector type as that is the one most suitable for layout tasks (although you could use others). Call this *container*. That is a conventional name that many use for this purpose. Click *OK*.



We could define the way that text looks here, but now we are only interested in layout. Go to the *Box Category*, and choose 800 pixels as a width for the container. You could make it bigger or smaller. Everything in your page will be in this container and therefore be no wider than 800 pixels. For *Clear*, select *both*; and for *Margin* select *0, auto, 0, and auto* (uncheck *Same for all*). Click *OK*.

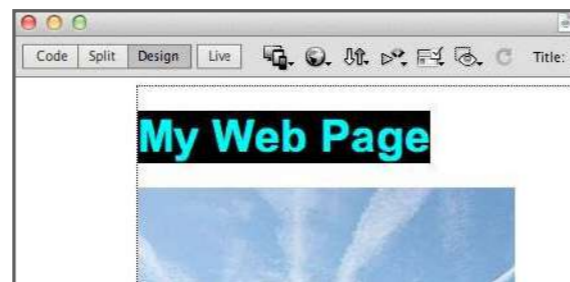
Looking at your **Design View** you can see (if your window is stretched out enough) that there is a dotted box that is 800 pixels wide. The dotted box would not show if you looked at the page in a **browser**—it is a visual aid in Dreamweaver.

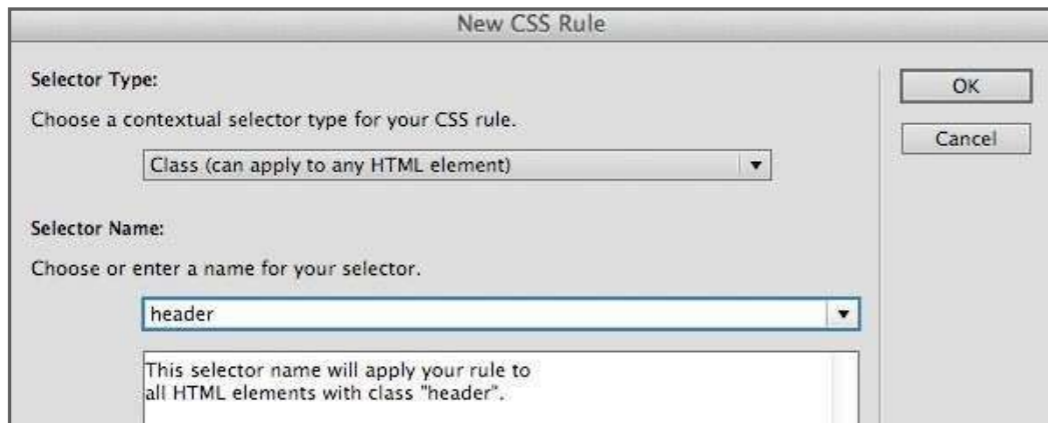
By the way—if your content does not center when you preview it in your browser, you could be using an older version of Dreamweaver 5. Update the application and try this lesson again.

Up to this point everything in this lesson is something you could easily use for every web page you make.

Now I am going to have to make a design decision. Since a two-column layout with a header (information across at the top) is fairly common, let's give our page that layout.

To do this, first define a space for the header. Highlight only your short **paragraph** and click the *Insert Div Tag* button in the **Insert Panel**.

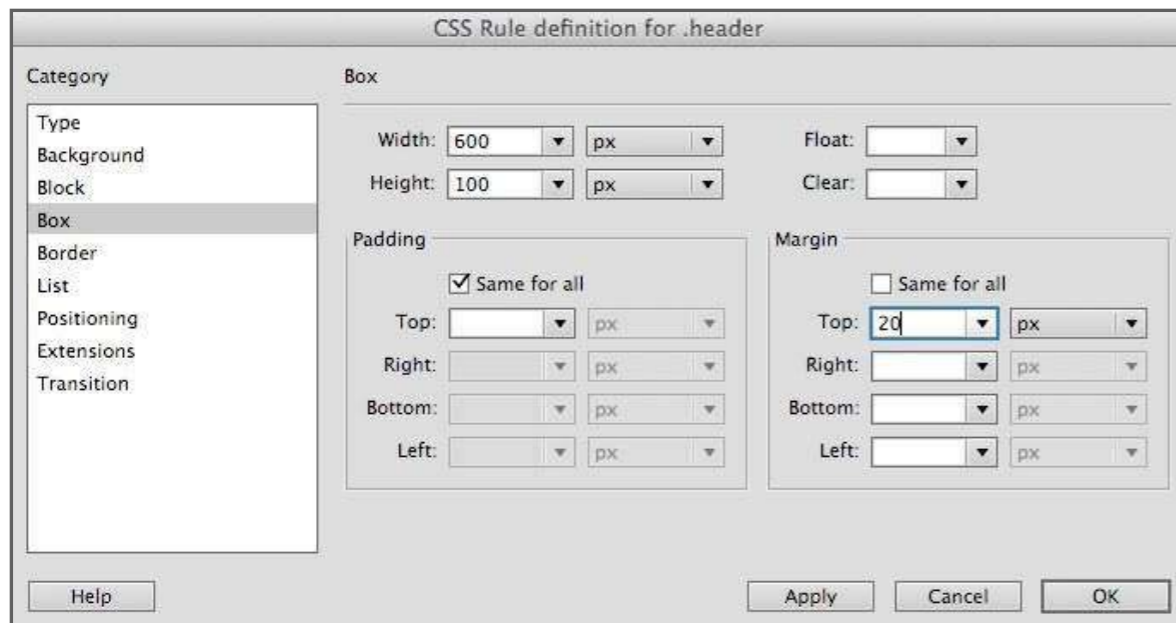




Making sure that the requestor box is set to *Insert:Wrap around selection*, click the button to make a *New CSS Rule*.

Change the selector type to *ID (applies to only one HTML element)* and name the style something descriptive like *header* and click *OK*.

In the *Box Category*, enter in the information as in the illustration. The *margin* is how much room around the outside of box there will be. I set it to 20 pixels just to get it away from the very top of the page. You can always change any of these things later. Click *OK* when you are done.



Now is a good time to point out that in web (as with anything where the screen is the final medium), we don't use inches as a measure. It really has no meaning since screens come in all different sizes and resolutions. *Pixels* is the unit that is used for most things. Get used to thinking in terms of them.

As a frame of reference, the computer you are working on has a **pixel dimension** of at least 1024 pixels (wide) by 768 pixels high. Most newer computers exceed this dimension.

In the **Design View** you can see that the box that we made for the header is shown by dotted lines. We could have made our header

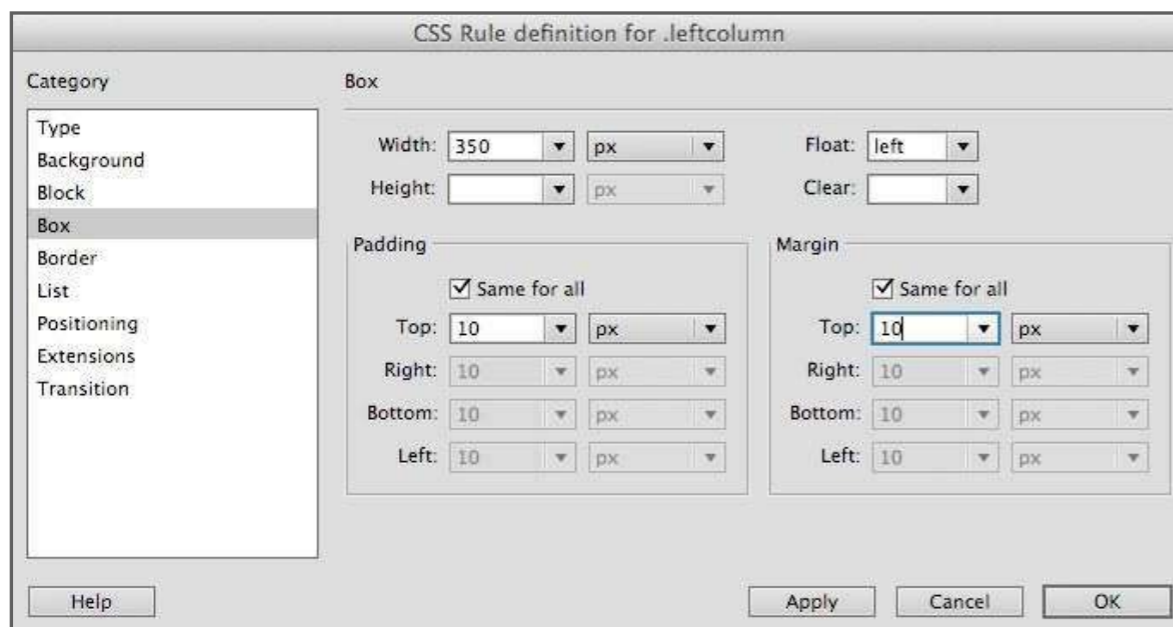
longer than 600 pixels to fill the *container*, but since we are not going to be using the upper right, this measurement works fine. And you can always change it later. Many web pages use an image for the header even if it is type. To do it that way, just highlight the image instead of the type as we did here.

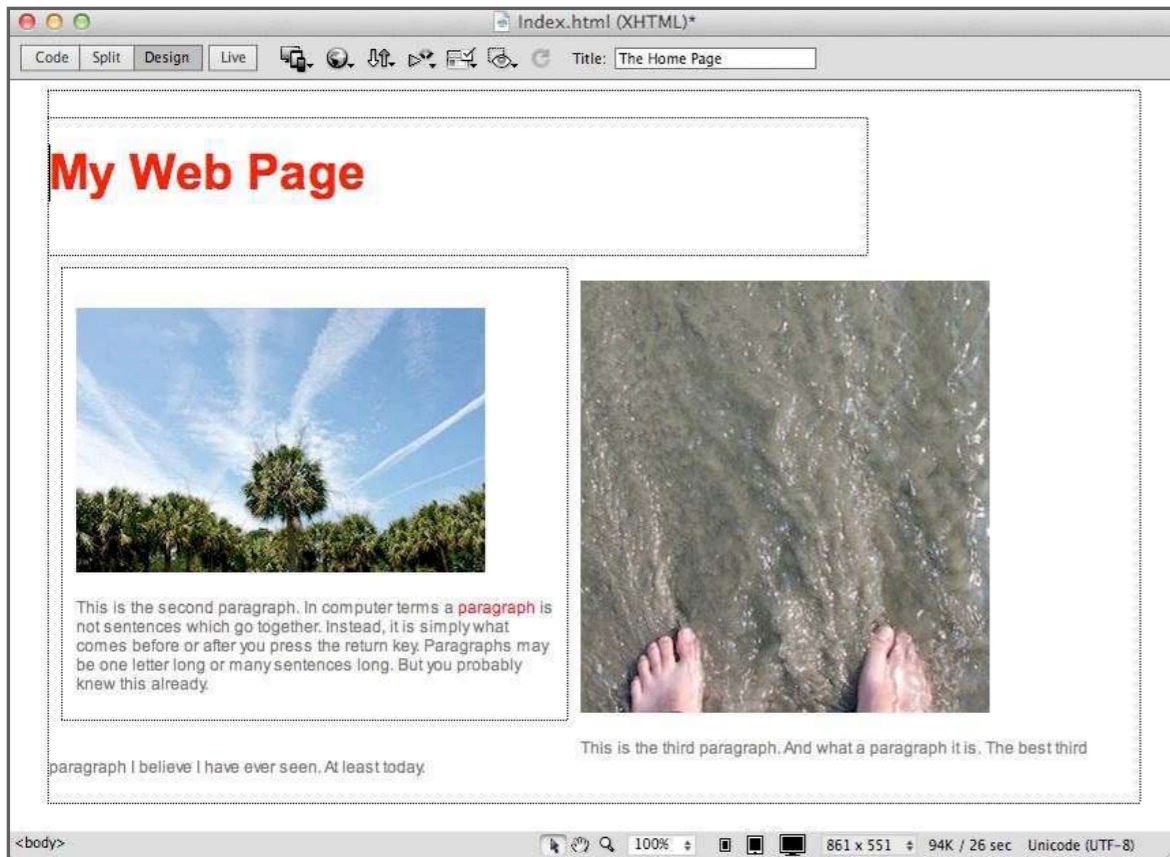
Now let's make a left column in the same way you made the header. Select everything you want to be in the left column—in my case I selected the first photograph and the second **paragraph** (the long one) by dragging the cursor over it all. Click the *Insert Div Tag* button in the **Insert Panel**.

As before, make a new **CSS rule** and change your selector type to *ID*. Name your rule something descriptive. I named it *leftcolumn*.

Enter the information as in the example. Since I will be making two columns, and entered 350 pixels as the column width. I can always change this later. **Float** commands the box to align to the closest left border, which in this case is the container edge. **Padding** is how close to the box text or photographs contained within it come to the edge. **Margins** are how much room around the outside of the box there is.

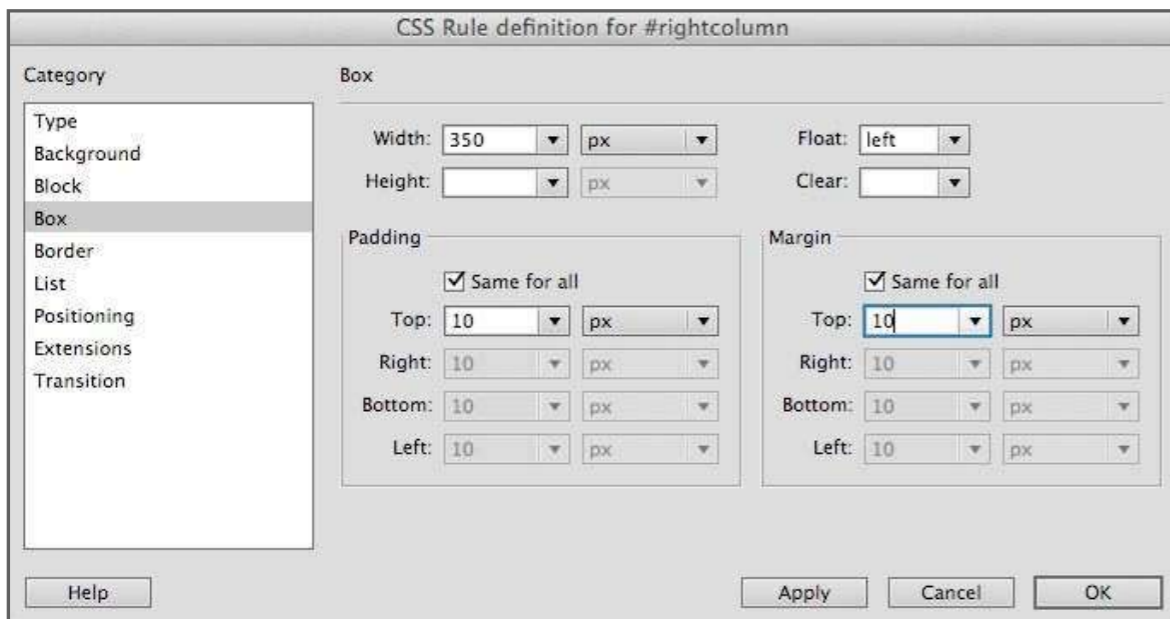
One strange thing is that both **padding** and **margins** are added to the dimensions of the box. That is why I entered 350 instead of 400 (half of 800) for the box width.





If I now look at my design view I can see that my header and left column are where they should be, and the rest of the information comes after it as it will (your example may look different than this—that is fine).

Now you need to add the right column. Select whatever information on your page we haven't dealt with yet. In my case it is the second photograph and the paragraph after it. Click *Insert Div Tag* in the **Insert Panel** and make another new **CSS rule**. As in the last column make sure the selector type is *ID* and name the selector something descriptive such as *rightcolumn*.



Enter the information as in the example. You will notice that it is the same information we entered for the left column. In this case the *Float: left* will cause the box to still align with the closest left border, which in this case is now the left column (we could have also had it float right—to the container edge). If the width of the right column was too large (which it is not), then it would simply slide under the right column to where it would fit. Click **OK**.

Go to your **Design View**, and see that everything is pretty much how it should be. If it isn't, back up and try again. Make sure your photographs are at most 300 pixels wide—if they are much bigger they will prevent the right column from going to the right.

```
#rightcolumn {
margin: 10px;
padding: 10px;
float: left;
width: 350px;
}
</style></head>

<body>
<div id="container">
<div id="header">
<h1 align="center">
</div>
<div id="leftcolumn">
<p>
</p>
</div>
<div id="rightcolumn">
<p>
</p>
</div>
</div>
</body>

.clearfloat{
clear:both
height:0;
font-size:1px
line-height:0px;
}

</style></head>

<body>
```

There is one problem remaining. The container box we first made now looks like it stops with the header. It shouldn't, as it really does contain everything. But it still looks that way according to the dotted lines.

The easiest way to cure this anomaly (and you probably should) is to go into the **code** and change it. This will also give us a chance to work with **code** for just a bit.

Switch from **Design View** to **Code View** (upper left of your window). Near the top of the page you will find where the **CSS styles** are defined (see illustration). Place your cursor right before `</style></head>` and push the return key several times. Empty lines don't matter in HTML, so don't worry about extra returns in **Code View**. Type the following *exactly*, and don't forget the period at the beginning or the bracket at the end (see example). If 'helper boxes' pop up as you type, disregard them.

```
.clearfloat {

clear:both;


height:0;

font-size: 1px;

line-height: 0px;

}
```

```
<div id="header">
  <h1 align="left" class="big
</div>
<div id="leftcolumn">
  <p>This is
after you press the return key.
</div>
<div id="rightcolumn">
  <p>This is
</div>
</div>
</body>
</html>
```



```
<p>This is
after you press the return key.
</div>
<div id="rightcolumn">
  <p>This is
</div>
<div class="clearfloat"></div>
</div>
</body>
</html>
```

We just defined a **CSS style**, and now we need to apply it. In this case right before the **closing tag** of our container.

Go to near the bottom of the code, and put your cursor right before that last **</div>** tag. Hit return a couple of times to make things clearer, and type:

```
<div class="clearfloat"></div>
```

Check to make sure it looks like the **code** in the illustration.

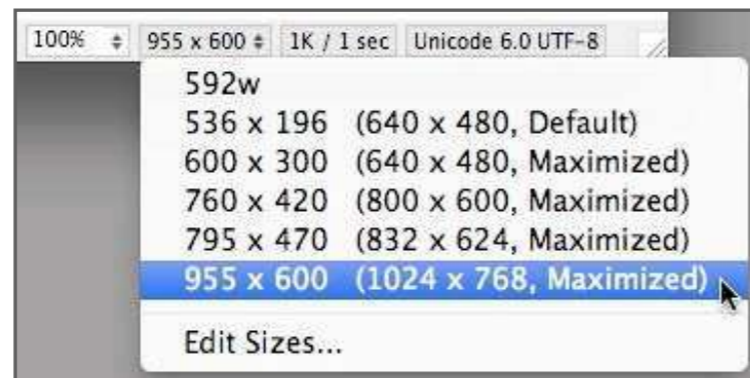
Go back to **Design View**, and you should see that the container box now stretches below the columns the way it should. If it doesn't, check carefully for typing mistakes.

Try editing your column sizes to different widths. How about a main column and a sidebar (edit the style for *rightcolumn* to make it 500 pixels and for the right column to be something like 200 pixels. How about adding more photographs and text to the columns?

Try adding a background color to your whole page by editing the **<body>** tag. There are many ways of doing this, but the one you know is to click **<body>** at the bottom of the window and edit the **CSS Rule** for it in the **Properties Panel**.

Experiment and play around with this stuff. Refer back to the tutorials as needed.





And that's it. You now have a toolbox for doing very functional websites. You can add pages, information, styles, and add photographs. If you add photographs, you should size them in Photoshop or another image editing application to fit within your columns (remember to leave pixels for padding) and you should save them as JPEGs (or PNG or GIF) files.

Actually, you might find that you can just use the pages we made here as a template. Two columns with a header is a very common format—just take out the information you don't want, and put in the information you do want.

Or, if you would like a one column with a side-bar, just make one of the columns narrow. Or we could make a third column (or side-bar) by just adding a third column the same way we added the second.

When you are laying out your own web pages remember that your screen may be bigger than the screens of people viewing your web site. Most web sites today are sized for a screen that is 1024 pixels wide by 768 pixels tall. You can simulate this size in Dreamweaver in the menu at the bottom right of your working window. If you stretch out your window for any reason, go back to this menu to again preview how it will look on smaller screens.

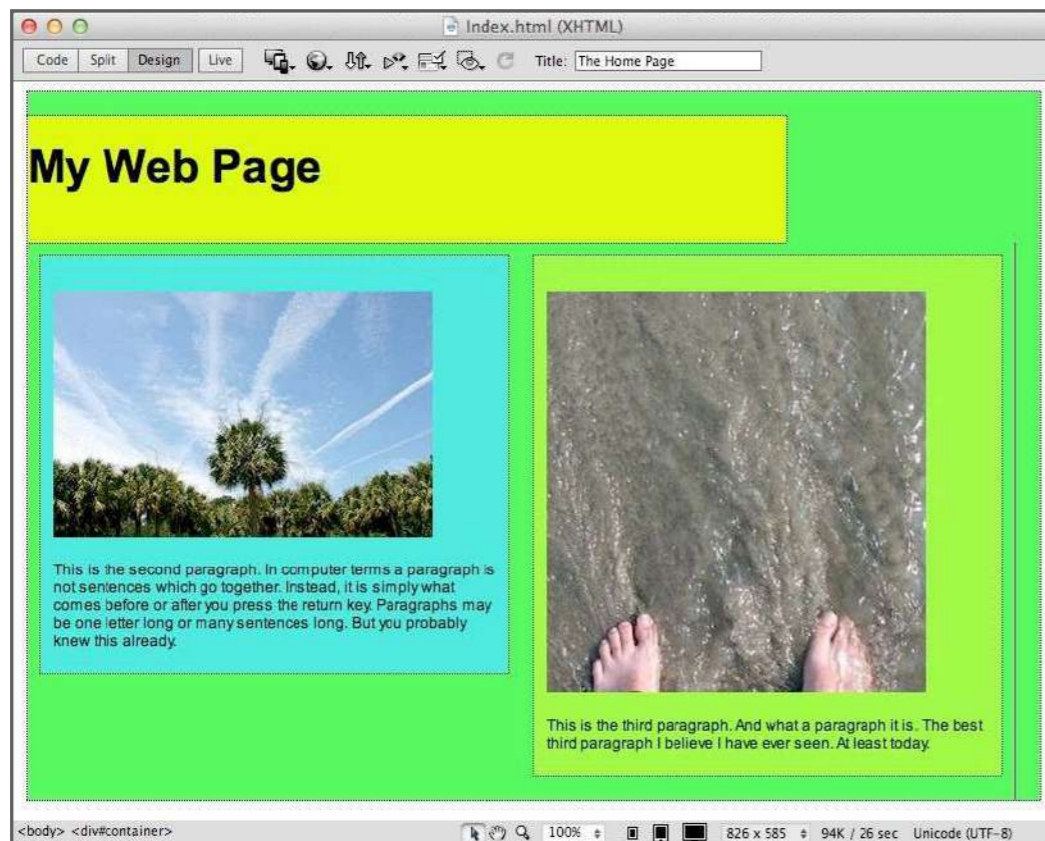
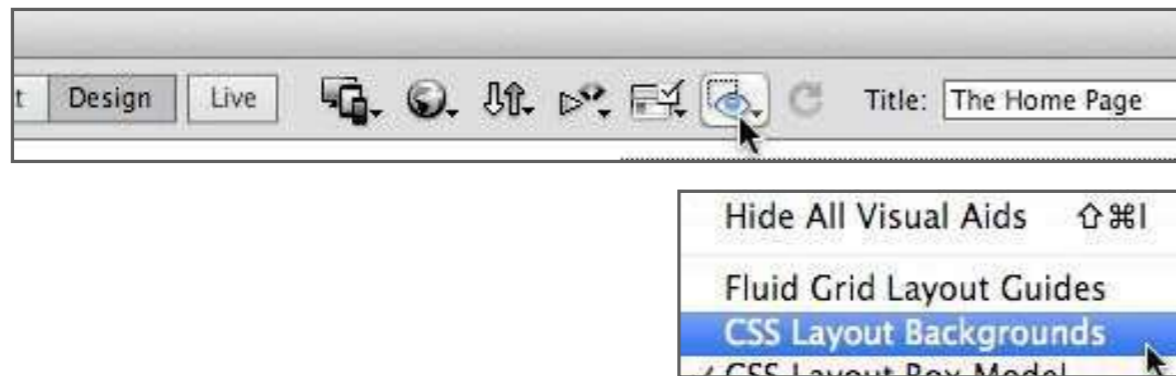
# Some Useful Things

*Now that you have a toolbox full of tools, it is time to figure out a little bit more about the finer aspects of using these tools.*

*But don't forget to learn to the tools we have covered so far in these tutorials. They represent a firm foundation from which to explore further.*

Test your pages in different **web browsers**. Personally, I test in Safari, Firefox, Chrome, and sometimes even Explorer. Remember that little globe at the top of your Dreamweaver window that allows you to test in different **browsers**. You can add more **browsers** by going to the menu *File>Preview in Browser>Edit Browser list*.

Inevitably, things go wrong. One handy way of tracking down what went wrong in layout is to go to the top of the window, click on the eye icon, and turn on *CSS Layout Backgrounds*.



This will show each layout element as a different groovy color. It is useful when you get stumped as to why something is not working correctly.

Look at your code. One of the things that I am always taking out are **paragraph tags** (`<p>` and `</p>`). They screw me up so much that I usually just use line breaks between elements (images or paragraphs). You can insert a line break in **Design View** by hitting shift + return instead of hitting return.

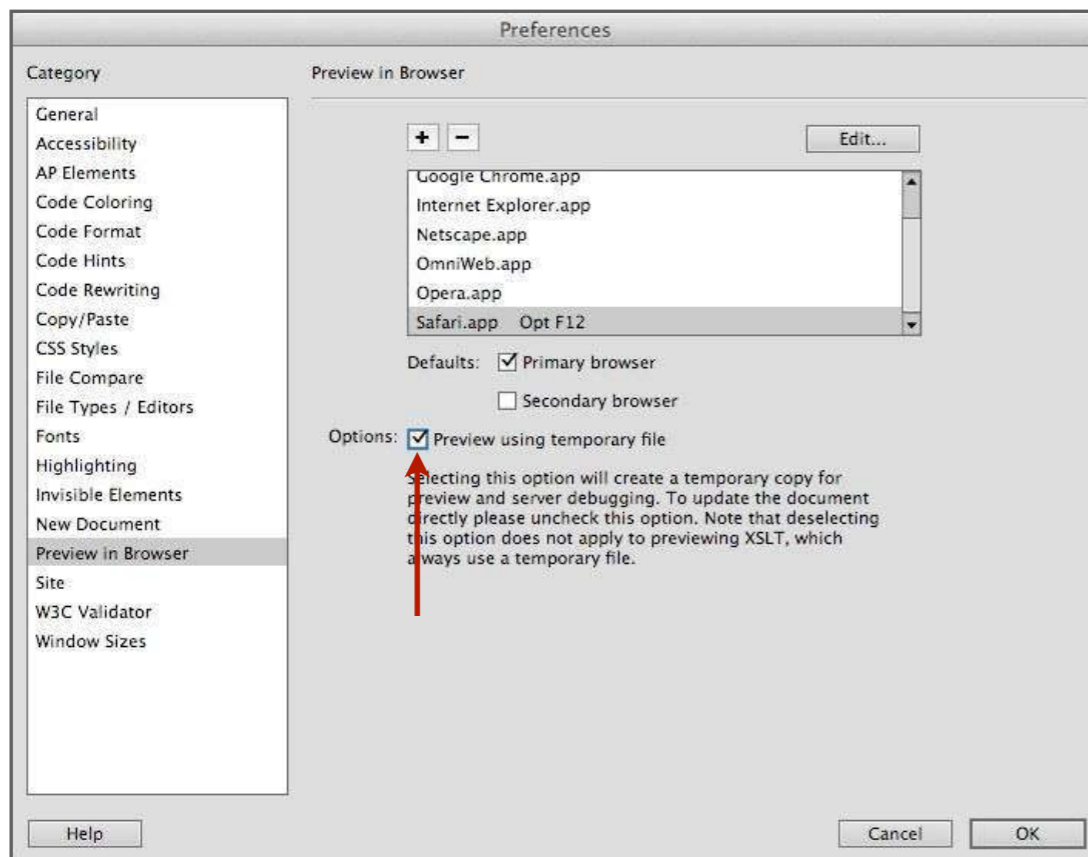
Often times, especially after a lot of editing in **Design View**, the **code** gets screwed up, so something like a **link** will extend across another **link**. **HTML code** uses pretty much plain English, and it is not too hard to correct these things by moving the **code** around a bit. Remember that paragraph returns you make in **Code View** don't effect anything, and you can use them to make the **code** easier to view and sort out.

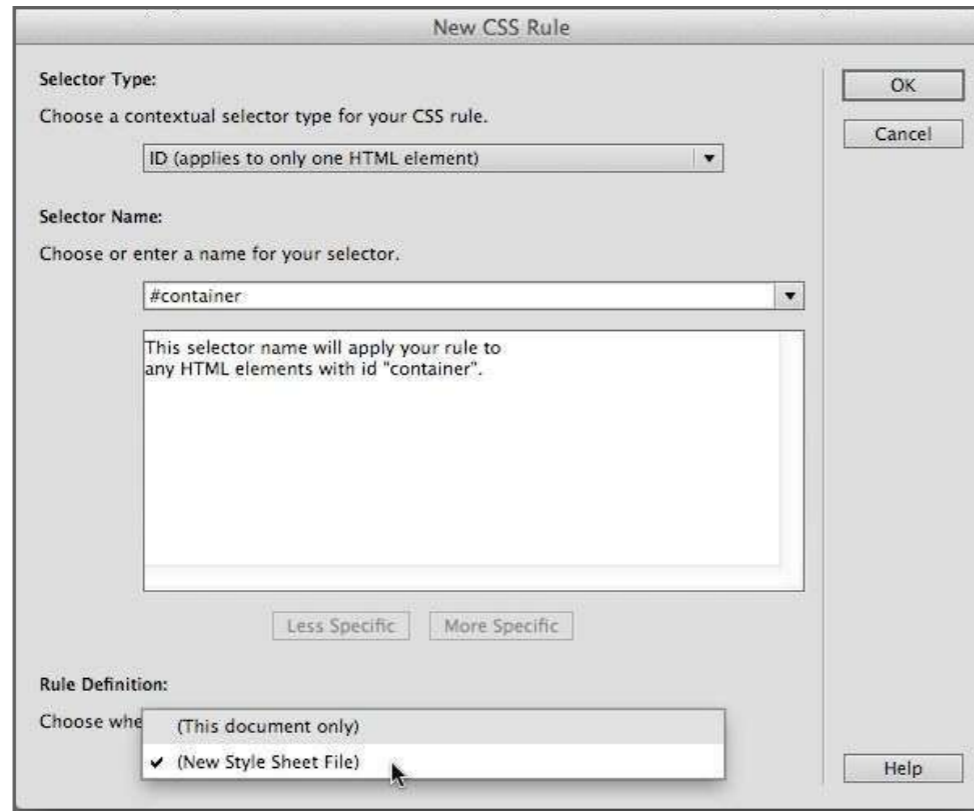
But most of all, try different things. By default, Dreamweaver is set up for 50 undos. Also, if you would like to **preview** your files in a **browser** without having to save them you can turn on this option in the preferences.

Also you can save multiple copies of a file, so that if you make too many mistakes you can always fall back on a file you saved before you really screwed things up.

Look at other people's websites. If you see a rollover effect or something else you really like, you can always 'borrow' it. All browsers allow you to view the source code, and you can copy as much as you need and paste it into your file.

It may be a little hard in this case to see where others keep their **CSS style** definitions, and this leads us to one other thing you should know, which is how to keep your **CSS styles** in a separate place.

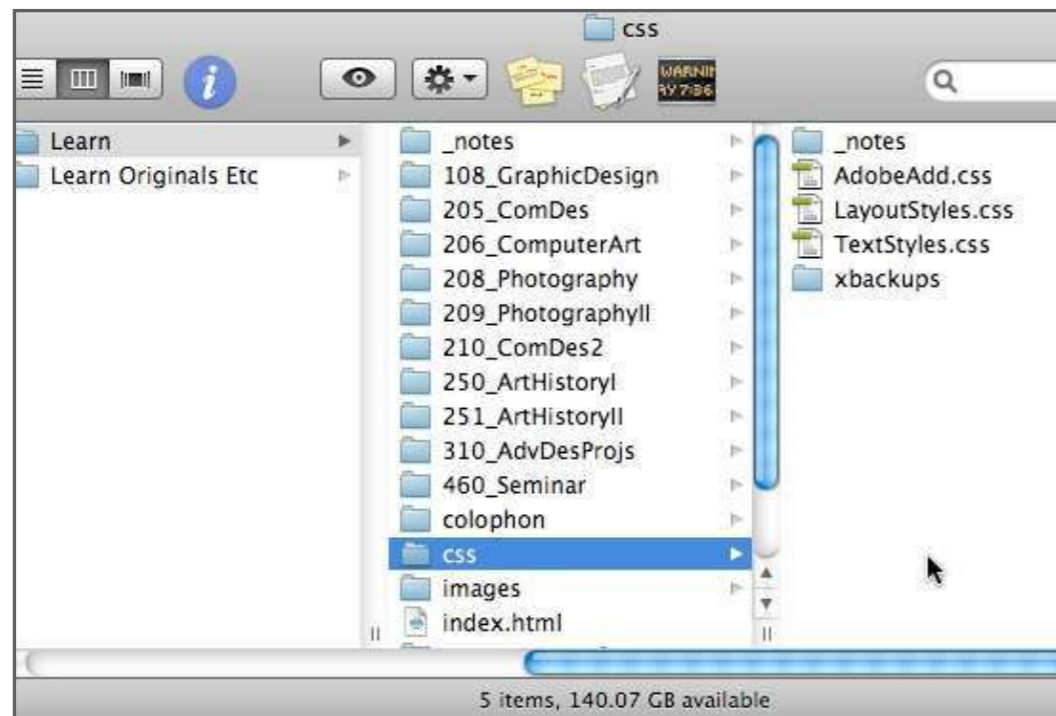




When you make a new **style** you can either apply it to the document you are working on, or you can save it somewhere else (*New Style Sheet File*).

There is one advantage to saving it somewhere else. If you are using the same **styles** across multiple pages (be it two or two hundred), changing your style sheet will change all the pages.

So let's say you make a website, and then a month later decide your link colors are bright, your text is a bit too small, and your columns are a bit too narrow. If you are using what is called an **external style sheet** where the styles for your entire site are on one page, it is a very simple matter to change the look of these things in one place and it will change your entire site.



The example on the left shows the basic folder structure for a website I use for my classes. Notice that there are three style sheets that govern the entire site. The first I called *AdobeAdd*, since this is where I make changes to styles that Adobe Lightroom produces when I have it automatically make web pages of student work.

The second style sheet is called *LayoutStyles*, and contains just that. I like to keep the layout styles separate from the styles that govern text (*TextStyles*), but making three pages instead of one to govern everything is purely personal preference. It would work just as well to have all the styles in one style sheet.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Payne's Learning Resources</title>
<link href="css/LayoutStyles.css" rel="stylesheet" type="text/css" />
<link href="css/TextStyles.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="container">
```

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

a {
    text-decoration: none;
    text-transform: uppercase;
}

a:link {
    text-decoration: none;
    color: #ddd;
    padding: 3px 0px 2px 0px;
    background-color: inherit;
    -webkit-transition: background 0.2s ease;
    -moz-transition: background 0.2s ease;
    -o-transition: background 0.2s ease;
    transition: background 0.2s ease;
}

a:visited {
    color: #ddd;
```

Each page you use a **style** in must be linked to the stylesheet. Here is the **code** for the home page (index) of the above website. I got to it in Safari by just going to the menu *View>View Source*. You can see (highlighted) the part of the page that tells the web browser where to find my **CSS styles**, in this case in the folder named `css`.

Since you can see where the **styles** are, you can enter this information in Safari's address box (in this case I cut `index.html` and added `css/TextStyles.css`). Since Safari sees that it is not a regular web page, it displays the code as text.

In the example on the left you can see that my **links** (*a:link*) on this site have something fancy going on. This is a transition effect that changes the background color of my links gradually as you roll over them (it is a very quick, so it is not very noticeable).

If you wanted a similar effect to your **links**, you could highlight the text you find in the source code and paste it into your **CSS style**.

Even just doing the techniques we covered in these tutorial, you will need additional information. If you use Dreamweaver Help (in the help menu), you might want to go to *Search Options* and turn on *Local* for the search location. If you do not turn on local search, you will get info from (as far as I can tell) all over the place and in no coherent manner. And it will display slowly. Not good for just quickly looking up something.

This book covers the very basics, but there is more. As a matter of fact, a lot more. And a lot of it is not that difficult. The developing HTML 5 standard allows for a lot of functionality in web pages, and new versions of CSS give more control.

Many times, if you want to have something fancy like a slide-show on your web page, it is a simple matter of finding the code and pasting it in. And although Flash as a multimedia tool is becoming more of a special-use tool, there are other programs that allow you to do interactive things and include them in your pages. These applications, which take advantage of HTML 5's capabilities in a user-friendly way, include Tumult Hype and Adobe Edge.

But there are, of course, times when you need someone else programming your web site for you (unless you are going to be a programmer yourself!). Anything to do with the exchange of money is a good example. Or web pages that link into databases, such as for forms or for user-editing capabilities.

Modern web techniques can get very complicated, and best leave the hard stuff to the people who know best. But if you know the basics, you can at least communicate more effectively with programmers when you do need them.

# Reference

*Here are a few things that are somewhat difficult to remember, so they are in a handy place for quick reference.*



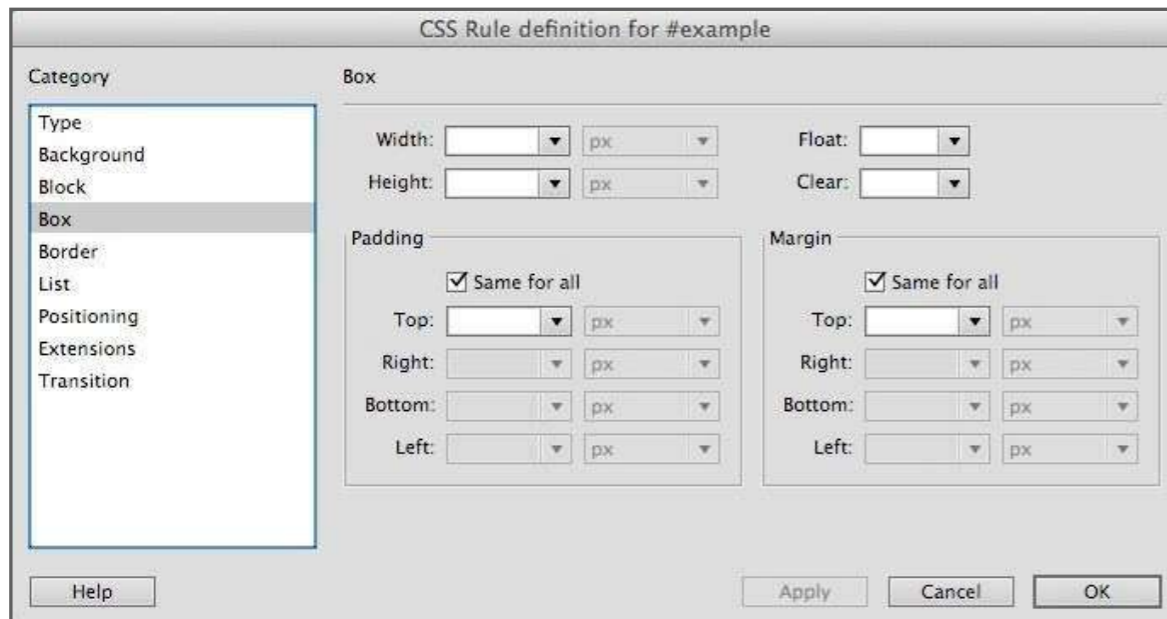


## CSS Type

**Class** Usually used for type formatting. Can also be used for layout if you want the same layout box to appear multiple times on a page. These show up in the CSS Style Panel and code with a period before the name (.bodytext).

**ID** Usually used for layout. Can be used across multiple pages with the same layout or even pages with different layouts (although you wouldn't probably want to do this). These show up in the CSS Style Panel and code with a pound sign before the name (#center-column).

**Tag** HTML has built-in formatting tags for things like bold type, headers (bigger type), and so on. This redefines these tags to appear differently than the built-in appearances. These show up in the CSS Style Panel and code as a name with no prefix symbol (body). Instead of naming these yourself, you should pick from the list of available tags when selecting.



## Box Category for CSS layout

**Width and Height** Width is used more than height, since an unspecified height will allow you to keep adding elements (pages scroll vertically!).

**Float** If you don't want your layout box to have a set dimension from the edge of the container (or whatever contains it), turn on float. This is used quite a bit.

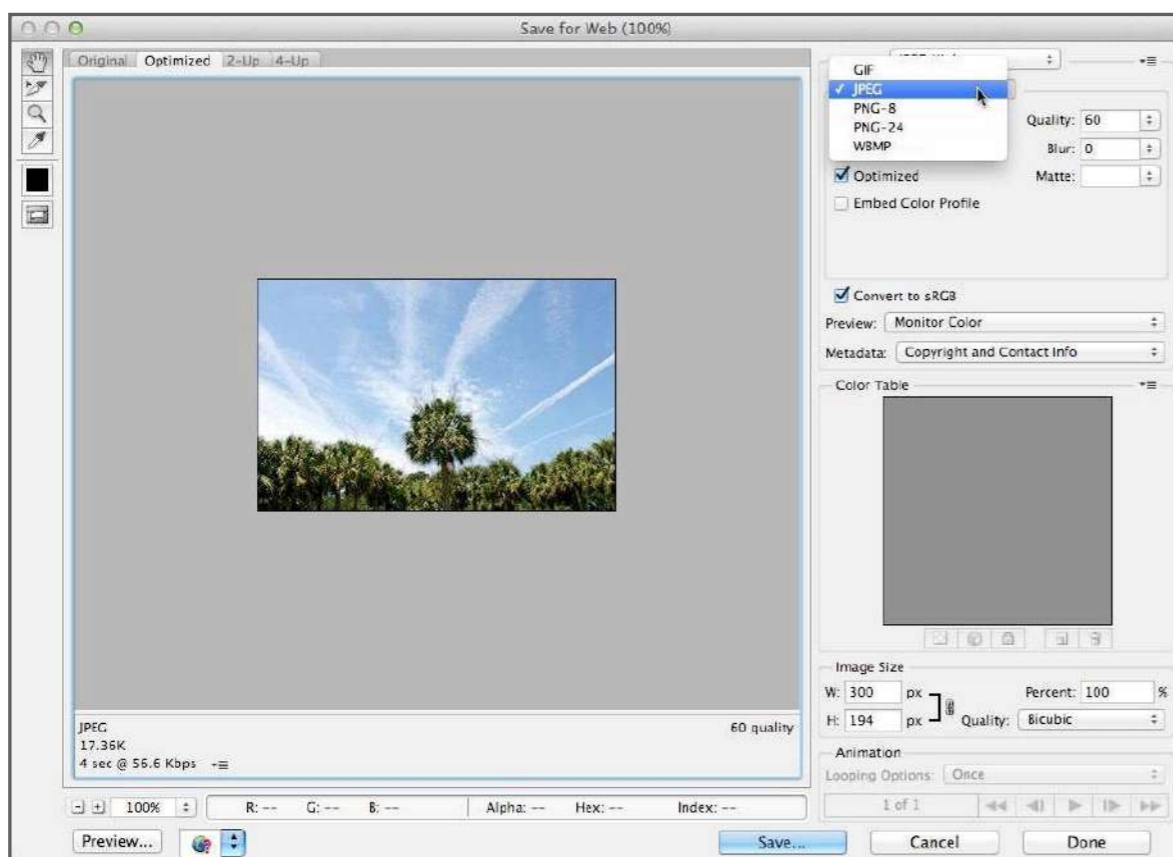
**Clear** If you don't want to have anything after your box to the right of it, but instead have additional things go down below it, specify this. This is not needed much of the time.

**Padding** This is the margin within a box.

**Margin** This is the margin outside a box.

Both margin and padding increase the size of a box (e.g. two 100 pixel wide boxes will not fit into a 200 pixel wide box if there is any margin or padding).

*Remember the help system – in this dialogue box (as in others), there is a help button which will take you to actual definitions.*



## Photoshop File types (Save for Web)

**GIF** Good for type (like titles and other things you don't want to use the browser fonts for) and other things with few colors. Can also save animations. Do not use for photographs.

**JPEG** Best for photographs. If a color profile is saved, make sure it is sRGB.

**PNG-24** PNG files basically replace GIF and JPEG, but with little advantage. The one exception to this is when you need something with a transparent background, such as type (like a title) or a logo over a solid color background. In Photoshop, put your logo, type, et-cetera in a transparent layer and save as PNG-24.

One can also copy type from Illustrator, paste it into Photoshop (it will come in as a smart object), and go back and forth between the applications to tweak and edit the type. Very cool.

# Body Tag

In code view, this is the tag that defines everything that is shown on a web page. By clicking on 'body' at the bottom of the working window you ensure that everything on the page (including code) is selected.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

**Chapter 3 - CSS for Type Formatting**

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

# Browser

A web browser such as Safari, Firefox, Chrome, and Explorer. And don't forget mobile devices—if your web page formats nicely on a phone, more people will be able to use it easily.

---

## Related Glossary Terms

Drag related terms here

---

## Index

### Chapter 1 - Making simple pages and links

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

# Cascading Style Sheets (CSS)

A style sheet is basically lines of code that tells the browser how something (like text) should look. Think of it as a dictionary definition for your browser. When it is referred to in other parts of a web page by a name, it is looked up in the 'dictionary' each time it is used. *Cascading* refers to the fact that you can have styles within styles that further modify them (think boxes in boxes).

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

# Closing Tag

Just about any code that starts governing how something on a web page looks has to end somewhere. That is what the closing tag does. It is usually the same as the start tag with the addition of a forward slash before the tag name. Tags are enclosed in brackets like this < >.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

**Chapter 2 - Structure, Images, & Text**

**Chapter 4 - CSS for Layout**

## Code View

Dreamweaver's main window that shows the web page you are making. Control which view you are in (Design, Split, or Code) by the buttons near the top left of the open window. Code View shows the web page's source code.

---

### Related Glossary Terms

Drag related terms here

---

### Index

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 5 - Some Useful Things](#)



# CSS Layout

Determining where things (like columns, etcetera) appear on a web page is done with CSS layout. Tables were previously used for this, and can still be used, although they are not as flexible.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

Chapter 4 - CSS for Layout

## CSS Panel

Located on the right of the screen in *Design Layout View* or go to the menu *Window>CSS Styles*. The panel with which you can add and edit CSS rules. One of the confusing things is that much of the functionality of this panel is duplicated in other places. To edit a rule here, highlight the rule and click the small pencil icon at the bottom of the panel.

---

### Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

Chapter 3 - CSS for Type Formatting

# CSS Rule (Rule)

A CSS rule dictates how any element on a web page looks. It is defined in code near the beginning of a web page or can be in a separate file. When it is invoked in a web page the browser looks for its definition either before the body tag in the same file or in a separate file.

---

## Related Glossary Terms

[Cascading Style Sheets \(CSS\)](#)

---

## Index

[Chapter 3 - CSS for Type Formatting](#)

**[Chapter 3 - CSS for Type Formatting](#)**

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

# CSS Style

The style of type or anything else that is governed by CSS tags. As opposed to HTML styles where the style is governed by non-CSS tags.

---

## Related Glossary Terms

Drag related terms here

---

## Index

### Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

# Design View

Dreamweaver's main window that shows the web page you are making. Control which view you are in (Design, Split, or Code) by the buttons near the top left of the open window. Design View shows the web page's design without showing the code.

---

## Related Glossary Terms

[Split View](#)

---

## Index

[Chapter 1 - Making simple pages and links](#)

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 4 - CSS for Layout](#)

[Chapter 5 - Some Useful Things](#)

[Chapter 5 - Some Useful Things](#)

# Enigmatically

Mysterious. As in: *He added the term enigmatically to the glossary for no obvious reason.*

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

Chapter 2 - Structure, Images, & Text

# External style sheet

An external style sheet is a file that contains definitions for CSS rules you use in a website. These are usually located in the web root folder. When you use an external style sheet you must also include its location when you use the terms in another web page.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

Chapter 5 - Some Useful Things

# Files Panel

The Files Panel (*Window>Files*, or on the right side in the designer workspace layout) shows the files inside your root folder. From this “mini-browser” you can open pages, make hyperlinks, or connect to a remote server.

---

## Related Glossary Terms

Drag related terms here

---

## Index

### Chapter 1 - Making simple pages and links

Chapter 1 - Making simple pages and links

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text



# Float

Float is a CSS term meaning where the object (such as a box) occurs horizontally on the page, column, or other container (either left, right, or center).

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

Chapter 4 - CSS for Layout

# HI

The h1 tag is one of several pre-defined HTML tags. It is generally not used since using CSS styles is a much more flexible way of defining how type is formatted. It is included in these tutorials as a quick way of seeing how code works.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

# HTML

Hyper-Text Markup Language. Not that you really need to know that. It is the 'language' the code in web pages are written in. Fortunately, it uses enough English that we can many times figure out what it is saying. CSS is also part of the HTML language, but don't let that confuse you.

---

## Related Glossary Terms

Drag related terms here

---

## Index

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 3 - CSS for Type Formatting](#)

**[Chapter 3 - CSS for Type Formatting](#)**

[Chapter 3 - CSS for Type Formatting](#)

[Chapter 5 - Some Useful Things](#)

# HTML Button

Along with the CSS Button, the most overlooked button in the Properties Panel.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

[Chapter 1 - Making simple pages and links](#)

[Chapter 2 - Structure, Images, & Text](#)

# Hyperlink (Link)

A hyperlink (or just plain *link*) is an object such as a photograph or piece of text which takes the user to another web page when it is clicked.

---

## Related Glossary Terms

Drag related terms here

---

## Index

### Chapter 1 - Making simple pages and links

Chapter 1 - Making simple pages and links

Chapter 1 - Making simple pages and links

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

# Insert Panel

The Insert Panel (*Window > Insert*) allows you to insert material into your web page such as links and images. It is on the right top of the screen in the Design Workspace Layout.

---

## Related Glossary Terms

Drag related terms here

---

## Index

**Chapter 1 - Making simple pages and links**

Chapter 2 - Structure, Images, & Text

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

# Live View

*Live View* is Dreamweaver's way of showing how a web page might look and behave if it were in a browser. Your mileage may vary.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

**Chapter 3 - CSS for Type Formatting**

[Chapter 3 - CSS for Type Formatting](#)

# Margins

A margin is how much room around the outside of a layout box there is. Measured in pixels.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

**Chapter 4 - CSS for Layout**

[Chapter 4 - CSS for Layout](#)



# Opening Tag

Just about any code that starts governing how something on a web page looks. Tags are enclosed in brackets like this < >.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

Chapter 2 - Structure, Images, & Text

# Padding

*Padding* is how close to the layout box text or photographs contained within it come to the edge. Measured in pixels (as is everything in web).

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

**Chapter 4 - CSS for Layout**

[Chapter 4 - CSS for Layout](#)

# Paragraph

The text or object placed before or after hitting the return key.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

### Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

# Paragraph Tags

Paragraph tags (<p>) start at the beginning of a paragraph and close at the end of a paragraph (</p>). Saner people like to use line breaks (shift + return in design view) instead of paragraphs. Some would disagree. And they probably know more. Oh well.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

**Chapter 5 - Some Useful Things**

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

# Pixel Dimension

Web pages and computer screens are measured in *pixels* instead of inches or centimeters. The pixel dimension of this iPad is effectively 1024 pixels wide by 768 pixels high. Just to give you an idea of how big they are.

---

## Related Glossary Terms

Drag related terms here

---

**Index**

Find Term

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 4 - CSS for Layout](#)

# Preview

Previewing at a file through a browser such as Firefox or Safari will ensure that your web pages behave as they should in any particular browser. Generally you should preview all your pages in all browsers you would like them to behave properly in, including those on mobile devices such as phones and tablets. The button to preview a file is a small world icon at the top of Dreamweaver's main window.

---

## Related Glossary Terms

Drag related terms here

---

## Index

**Chapter 1 - Making simple pages and links**

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 5 - Some Useful Things

# Properties Panel

*Window/Properties* or near the bottom of your screen in Designer Workspace Layout. This is a multi-purpose panel that has two modes—CSS and HTML. Most of the choices in this panel are also found in other panels and menu items, but are easier and quicker to get to in this panel.

---

## Related Glossary Terms

Drag related terms here

---

## Index

### Chapter 1 - Making simple pages and links

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 4 - CSS for Layout

## Rollover (Hover)

When you put your cursor over an object in a web page it can change (text can change color, images can change, etcetera). This is called a *rollover* or a *hover*. Most mobile devices do not recognize the *hover state* of objects, so it is not a good idea to make them necessary.

---

### Related Glossary Terms

Drag related terms here

---

Index

Find Term

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting



## Source code (code)

The source code (also just called *code*) is made up of words and symbols that tell the browser how to display the information on a web page. All applications have code underlying the what you see, but in Dreamweaver it is necessary to sometimes work with it. At least a little.

---

### Related Glossary Terms

Drag related terms here

---

### Index

#### Chapter 1 - Making simple pages and links

Chapter 1 - Making simple pages and links

Chapter 1 - Making simple pages and links

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

Chapter 5 - Some Useful Things

# Split View

Dreamweaver's main window that shows the web page you are making. Control which view you are in (Design, Split, or Code) by the buttons near the top left of the open window. Split View shows the web page's source code beside its design.

---

## Related Glossary Terms

[Design View](#)

---

## Index

[Chapter 1 - Making simple pages and links](#)

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 2 - Structure, Images, & Text](#)

[Chapter 3 - CSS for Type Formatting](#)

# Tag

Code that governs how something on a web page looks. If a word needs to be in bold, then a tag before the word, called the *opening tag*, (or just *tag*) would tell it to be bold. After the word a *closing tag* would tell it to not be bold anymore. Tags are enclosed in brackets like this < >.

---

## Related Glossary Terms

Drag related terms here

---

## Index

### Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 3 - CSS for Type Formatting

Chapter 4 - CSS for Layout

Chapter 4 - CSS for Layout

# Web Root Folder (Root Folder)

A web browser needs to 'know' where things like images and linked pages are. Therefore, it is crucial that all of your files are in the same 'master folder'. This folder is called the *web root folder* or simply the *root folder*.

---

## Related Glossary Terms

Drag related terms here

---

Index

Find Term

Chapter 2 - Structure, Images, & Text

Chapter 2 - Structure, Images, & Text