
IMAGES

So far you can build a web page with text and links. You can modify the text to look how you want it to look, but you haven't yet learned to make your pages into fine shining examples of heavenly art. You're going to need some color, and images. Without further delay, let us learn:

Background Color

Setting a background color is easy. Remember how the tag had modifiers, like COLOR, SIZE, and FACE? The modifiers affected the font tag, and were listed *inside* the tag itself (example:). Well, Background color is a modifier for the <BODY> tag. It is important to keep in mind that you can only have one body tag. You are going to add the background color modifier to the *existing* body tag.

A body tag that includes a background color looks like:

```
<BODY BGCOLOR="whatever_color">
```

Remember the "one command, one word" rule about modifiers? In HTML, each command is its own separate word. BGCOLOR sounds like it might be two words, but you're only telling the web page to change the color, which is only one command. So, BGCOLOR is one word.

1. Open up an HTML page and change the background color. Experiment. Have fun

Be careful with your background color choices. Remember that the web is still a written medium. It's very hard to read over glaring yellow.

Image Types for the Web

Of course you'll want some pictures to spice up your website. Images are also used as buttons, banners, and design elements. There are different types of image files for the web, and if you use the wrong one, your web pages will look bad. The different image types compress images to make them download fast in different ways and different compression strategies work better for different types of images. Luckily for you it's pretty easy to decide which compression type to use. Images and their compression types are easily identifiable by a three letter file extension that immediately follows the file name.

.JPG(pronounced jay-pegg)

JPG stands for Joint Photographers Expert Group. Although it's not super necessary for you to memorize what all the initials for all the file types stand for, some people like to know. The compression method JPG uses works very well on an image that is (industry term coming) *non-continuous tone*. Tone means color, right? Continuous means "keeps right on going". So, "non-continuous tone" means colors that don't keep right on going. What we're talking about here is a photograph. If you took a picture of my face, you'd have one color here and another color there. JPG was designed to show color photographs on a computer. If you try to save a picture that *is* continuous tone as a JPG, it'll look messed up.

.GIF

.GIF stands for, "Graphics Exchange Format". Later on we will get much deeper into the different compression methods that different image types use, but for now, you should know that a GIF is generally intended to be used for images created on a computer, with solid colors and text.

Here are some examples of images saved as correct and incorrect image types:



GIF (incorrect)



JPG (correct)



GIF (correct)



JPG (incorrect)

Okay, where should I get these images?

Create them yourself.

You can use image creation software like Macromedia's Fireworks, or Adobe's Illustrator. More about these programs later.

Scan them in.

As a web designer, you should have a scanner. Luckily you don't need a super expensive one. Hewlett Packard makes very good scanners, and they've got a very nice one for about \$100.00. We'll talk more about scanning later, but you should know a little about resolution right now. Resolution, or DPI (dots per inch), refers to the number of dots that make up one linear inch on a picture. On the web all images are 72 dpi, because that's how many dots per inch a standard monitor can show. When you scan, your scanning software will give you a choice about resolutions. Choose 72 DPI.

Get images off of the web.

It's easy to take images off of the web. Simply right-click the image you want (ctrl-click on a Mac), and choose "save image as". Make sure you save the image somewhere in your root folder. Now you can use that image in your website. Note that using someone else's image without their permission may

constitute a copyright violation. Search for "Free Images" on the web, or try various free image web pages, like www.screamdesign.com

☑ **Buy a disk with images on it.**

Any major computer store will sell you disk with thousands of images for around \$40.00.

☑ **Hire an Artist**

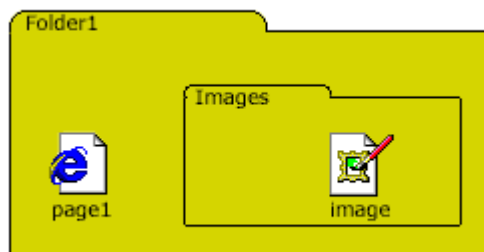
Artists can be expensive, but here's a nice tip, try local art schools. There are geniuses just begging for work they can use for their portfolio. You can have logos and other original designs created for very inexpensive rates. Try the Academy of Art Collage in San Francisco at: 415.296.2088

Now I've got images, how do I place them in my web page?

The tag you use to place an image in your web page is:

People refer to this tag as the "image source" tag. Don't forget about file paths!

1. How would you write the tag in the example below to insert image.gif into page1.htm?



The tag would be ****

Image Tag Modifiers

The image tag has some very important modifiers. You can use any or all of these modifiers in the same image tag.

Height and Width

You can add height and width settings to your images. Height and Width is measured in pixels. A pixel is a dot. Remember that the Internet uses 72 dots per inch?

An image tag with a height and width of 100 pixels would look like:

```
<IMG SRC="any.gif" WIDTH=100 HEIGHT=100>
```

It is important to note that you will not change your image's size with these settings. If you do, you will either end up with a bad looking image or an image that takes too long to download. Change you image sizes with an image editing program like PhotoShop or Fireworks. So, if you can't change the actual size of your image, why would you even have Height and Width modifiers? Good question. The answer is, you will use a height and width modifier because they allow the web page to load faster and in a more pleasing manner. When the browser reads a height and width tag, it inserts a placeholder the same size as the image. The browser continues loading text, and then comes back to finish loading the image later. This provides a much nicer experience for people with slower modems.

The ALT tag

ALT stands for "alternative". The ALT tag allows a viewer to see a word in place of the image while the image is loading. There are, believe it or not, some people who browse with images turned off. There are also sight-impaired users who have their web pages read to them by the computer. The ALT tag, added to an image tag might look like this:

```
<IMG SRC="any.gif" ALT="alternative text here" WIDTH="100"  
HEIGHT= "100">
```

☑ **ALIGN**

You can put two images next to each other. You can put an image next to text. HTML treats an image like it was just another word. Text will line up next to the image, but only one line of text.

The text will not "wrap". Later you'll learn tricks to get around this. By default, if an image is taller than a image or text next to it, the smaller element will center itself vertically with the taller object. You can use modifier tags in the larger image to affect orientation.

☑ **BORDER**

If you'd like to, you can assign a border to an image. The border size is measured in pixels.

Remember that you can use any or all of these modifiers in the same image tag.

Using Images as Links (Buttons)

To use an image as a button, simple insert it between the opening and closing hyperlink tag. Any image modifiers can be used, but when an image is used as a button the browser will automatically insert a blue border around the image, unless you tell it not to with BORDER="0"

Background Images

Images can also be used as a background to a website. When using an image based background you have to be very careful about download time. A very large image will take a long time to download. There are tricks you can use to keep file size small even if the image itself is large. We'll learn those tricks later. Right now you should know that an image used as a background will tile itself until it covers the entire page. The computer will download the small image quickly, and duplicate it using the processor. The image would continue duplicating until it covered the whole page.

The tag used to add a background image is a modifier of the existing BODY tag:

<BODY BACKGROUND="whateverimage.gif or .jpg">

Notice that the modifier is a different than BGCOLOR. You cannot have a background image and a background color.

LISTS

Without lists the world may very well cease to function. No one would get any shopping done. "To-Do" lists would disappear along with any semblance of productivity. Directions would become horribly muddled in paragraph form, and fancy key features, devoid of their lavish bullet points, would feel rather unimpressive, indeed.

In HTML, you have the choice of designing with "ordered lists" that are numbered or using unordered lists, which use bullet points to get your, um, point across.

Ordered Lists

Ordered lists require some kind of marker denoting a specific order. You can use numbers (1,2,3), letters, (A, B, C), or Roman numerals (I, II, III) if you want to get really swanky. Ordered lists are made to be used when listed information is displayed in order of importance or as directions.

The two tags that you will use in your ordered lists are:

**** and ****

 stands for "ordered list", and is used to start a list.

 stands for "list item", and is used to start any item in your list.

The list will automatically number itself. Nice, eh?

To make a list that reads:

1. Get up

2. Turn off alarm
3. Go back to sleep

You would enter in the following code:

```
<OL>  
    <LI> Get up </LI>  
    <LI> Turn off alarm </LI>  
    <LI> Go back to sleep </LI>  
</OL>
```

Notice that I've indented the tags. It's not really necessary that you do so, but I've done it for a reason. In a minute, the lists will get a bit more complicated and visual organization is important. Also take note that I've closed the tag. This is very important; otherwise your browser will assume that everything after this list is also part of the list.

NESTED LISTS

What if you want a list that says:

1. Get up
2. Eat food
 - a. Open mouth
 - b. Insert food
3. Go back to bed

You've basically got two lists here, right? One list contains "get up", "eat food", and "go back to bed". The other list contains "open mouth" and "insert food". The "open mouth, insert food" list will be built directly inside of the larger list.

When a list is built inside of another list it is said to be "nested". Actually, when any HTML code is built inside of code of the same type, the code is said to be nested.

"Nested" is a pretty important term; you'll see it more often later on.

To build the above "get up", "eat food", and "go back to bed" list with it's nested sub-set:

```
<OL>
  <LI> Get up </LI>
  <LI> Eat food </LI>
  <OL>
    <LI> Open mouth </LI>
    <LI> Insert food </LI>
  </OL>
  <LI> Go back to bed </LI>
</OL>
```

Notice that the "open mouth, insert food" subset is a completely different list. It has it's own and . Closing each list is important. If you forgot to close the "open mouth, insert food" subset, the last point of the main list, "goes back to bed", would be included in the sub-set.

You don't have to tell the sub-set to use a different numbering style, it knows all on its own. Again, the indenting is done to keep track of where you are in the list, not because it's required.

Sub-sets can have sub-sets, too. You can have as many sub-sets as you feel like.

Can you build the following list?:

1. Build a list
2. Feel really proud
3. Get high paying job at www.Lists.com
 - a. Buy Porsche
 - b. Retire

Go ahead, try to build the list. Practice is a good thing.

Okay, here's what the list would look like:

```
<OL>
  <LI> Build a list </LI>
  <LI> Feel really proud </LI>
  <LI> Get high paying job at www.Lists.com </LI>
</OL>
  <OL>
    <LI> Buy Porsche </LI>
    <LI> Retire </LI>
  </OL>
</OL>
```

Don't forget to close the main list!

Try to build some lists on your own.

Changing the Way Lists are Numbered

"But I'm proud of my Italian ancestry." You say, "I'd rather use roman numerals." Never fear. You can make a list with numbers, letters, or roman numerals. You simply add a **TYPE="modifier"** to the tag. For example:

```
<OL TYPE= "a"> (standard lettering, lowercase)
```

```
<OL TYPE= "A"> (standard lettering, uppercase)
```

```
<OL TYPE= "i"> (roman numerals, lowercase)
```

```
<OL TYPE= "I"> (roman numerals, uppercase)
```

```
<OL TYPE= "1"> (standard numbers)
```

Remember that when you close `<OL TYPE= "I">` you only have to use ``

Can you make a list that looks like:

- I. First Point
 - a. Sub-Point

UNORDERED (BULLETED) LISTS

Bulleted list are used when points are important, but in no specific order. HTML capitalizes on the fact that bulleted lists are in no specific order, and calls them "*unordered lists*". The tag for ordered list was ``. What do you think the tag for an unordered list is? That's right:

``

Other than the difference between `` and `` ordered and unordered lists work exactly the same way. Nesting is available in an unordered list. You can even nest an unordered list in an ordered list!

Go ahead and build this list:

- 1. First thing.
 - Bulleted sub-point

In HTML it would look like:

```
<OL>  
<LI> First thing. </LI>  
<UL>  
<LI> Bulleted sub-point </LI>  
</UL>  
</OL>
```

Nice.

Unordered List Modifiers

Don't like those boring bullet points? Well, experiment with the following:

```
<UL TYPE="circle">
```

```
<UL TYPE="square">
```

```
<UL TYPE="disk">
```

This is one of the only times that HTML is case specific. You have to write the unordered list types in lowercase or it won't work. Don't ask me why.

If you'd like to use a graphic for a bullet point, you can do it, but you don't do it with a normal HTML list. You have to build the list from scratch. You'd need to place everything in a table. Lucky for you, you're learning tables next!

NOTE: You're probably not going to be building lists everyday, but the concept of "nesting" is very important, and will be used again later on. It is important that you understand the concept of nesting. Basically, you are nesting code when you start a tag inside the area affected by a tag of the same type.

In the following example, you see a nested tag:

```
<FONT COLOR="GREEN" SIZE="5">
```

These words are green and size 5

```
<FONT COLOR="RED">
```

These words are red and size 5

```
</FONT>
```

These words are green and size 5

```
</FONT>
```

These words are neither green, nor size 5

TABLES

Okay, now you can really build web pages. You've got text, color, images and links. But your sites probably aren't organized exactly the way you want them. To design professional looking pages, you'll need to organize your information in tables. Tables are basically boxes that can contain text, colors, and graphics, and arrange those elements on the page.

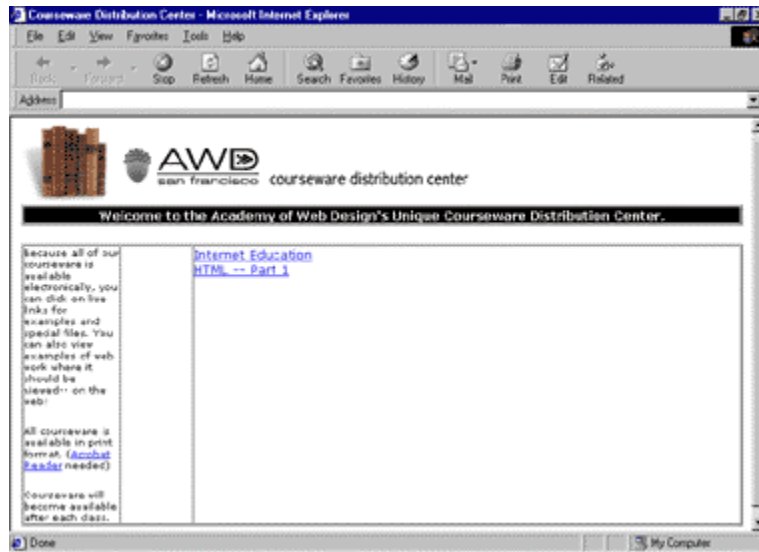
Most often you can't see the tables themselves; you can only see their contents.

The below example is a simple web page built with tables:



Notice that you don't see the table, only the information contained within.

The below example is the same web page with the table borders visible:



Sizing HTML Tables (pixel vs. percent)

You can control the size of the table itself and the cells within it. You can instruct a table to take up an exact pixel width (remember that a pixel is a dot), or you can direct the table or cell to occupy a certain percentage of the viewable screen size or available area.

Percentage-Based Tables

If built to take 100% of the screen, a percentage-based table will resize itself to occupy 100% of the screen, no matter how large or small the screen becomes. Many designers feel that it is important to occupy all of the screen size, or “real estate”, presented to them. Accordingly, they base their tables on percentage width, which is usually set at 100%. When a table is built using percentage based widths, images and text will rearrange themselves accordingly, but will not change size. Because text will seamlessly rearrange itself, a site that is heavily text based is easily manipulated using percentage based tables.

Pixel-Based Tables

Table sizes can also be exactly set. For example, if a table is set at a width of 200 pixels, that table will remain 200 pixels no matter how many pixels the viewer’s monitor is displaying. If the user has their monitor set to display 800 pixels across and 600 pixels down, a 200-pixel table will occupy one-fourth of the screen. The same exact table on a monitor with the viewable screen size set to 600x460 pixels

would occupy one-third of the screen. Designers use pixel based tables to preserve the relationship of objects to one another. If it is very important that a column stays a certain width for design reasons, a pixel-based width will be used.

Using Pixel and Percentage-Based Widths Together

Most people agree that a website that occupies the whole screen is more visually pleasing. But most websites contain elements, like navigation columns, that should remain a fixed width to preserve their design. What is a web designer to do?

The answer is to use both pixel and percentage based measurements on different columns. For example, the following table is set to occupy 100% of the viewable screen size. However, the widths for the individual cells can be fixed, and in this case pixel based widths were used on the end cells. The middle cell does not have a stipulated width, and will resize if needed so that the table can meet it's obligation to occupy 100% of the screen.

<-100 pixels->	<-No set width->	<-100 pixels->
----------------	------------------	----------------

Using this fancy trick, web designers can have the best of both worlds.

Some Rules to Keep in Mind:

Column Width

Knowing this rule will save you untold hours of pitiful sobbing at your computer. In HTML, columns are the same width *all the way down*. That means that you will not have a table that looks like this:

column a	column b
column a	column b

There are fancy ways to make a table that looks like the one above, but for right now, burn it in your mind: If a cell has a width, in either percentage or pixels, the cell directly beneath and above MUST be the same width. You don't have to stipulate the widths for the other cells, they'll resize all on their own. If you break this rule, your tables will... Well, they'll really freak out.

Cell Height

Usually you'll set cell width, and let the elements or text in the cell determine the height. Remember that if a cell resizes, elements will realign but will not resize. If a cell containing text in a percentage based table resizes itself, the text will re-align and will need to take up more or less height. Setting a height can cause serious problems if text or images don't have enough room to correctly display. For this reason, don't set a cell or table height unless you really need to, for example to stretch your table past the bottom of the screen if your website is small.

Standard Screen Size

The average Internet user is viewing pages on a 15" monitor, set at 800x600 pixels. The viewable screen area on the Internet Explorer and Netscape (not including borders) is 760x420. A website that requires side scrolling is considered very un-professional. Therefore, if you are building a pixel based website it's not a good idea to make it larger than 760 pixels wide.

Table Elements

There are basic elements to all tables that translate directly into tags.

Lets go over the main tags we will use to build tables. Don't bother trying to build a table yet, I just want you to see the necessary tags and become familiar with them.

First, you have the table itself. To let HTML know you're starting a table, you must use the table tag:

<TABLE>

Tables also have rows. Whenever you start a new row in a table you must use the table row tag:

<TR>

Tables contain boxes, usually called cells. HTML, harboring a secret desire to be rebellious, decided not to call these individual boxes cells, but instead "Table Data Areas":

<TD>

You must use a <TD> whenever beginning a new Table Data Area (cell)

Each of these tags has a closing tag </TABLE>, </TR>, <TD>

Notice that we didn't name columns as a table attribute. This makes sense when you realize that if a row has three <TD>'s, or cells, that it also has three columns. HTML is attempting to avoid redundancy.

Lets Make a Table!

1. Make a new HTML page. Make sure you include all the basic elements, like <HTML> <HEAD> and <BODY>
2. Type the following code:

```
<TABLE>
<TR>
<TD> First Cell </TD>
<TD> Second Cell </TD>
</TR>
</TABLE>
```
3. Save your file as "table.htm" Remember to add the ".htm"
4. View your page in the Internet Explorer (File-> Open-> Browse)
Don't be freaked out if you don't see what you thought you'd see.

Remember that tables have a border. We didn't insert the border tag yet. The border tag is a modifier of the <TABLE> tag

1. Modify your code to read:

```
<TABLE BORDER="1">  
<TR>  
<TD> First Cell </TD>  
<TD> Second Cell </TD>  
</TR>  
</TABLE>
```

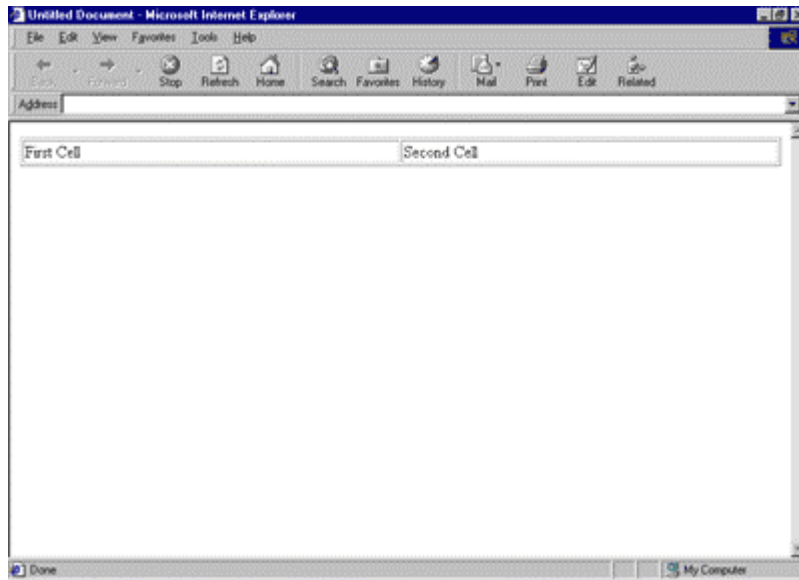
2. Check out the table!

You can see the table now, but it only takes up as much room as it needs. We need to add width modifiers. Width modifiers can apply to either the <TABLE> or the <TD> tag. In this case we'll want to apply a WIDTH modifier to both the <TABLE> and one of the <TD>'s

3. Modify your code to read:

```
<TABLE BORDER="1" WIDTH="50%">  
<TR>  
<TD WIDTH="50%"> First Cell </TD>  
<TD> Second Cell </TD>  
</TR>  
</TABLE>
```

4. Check out your table with the Internet explorer. It should look like:



If your table doesn't look right go back and check your code. Did you close all of your tags?

Practice what you know so far. There's more to know about tables. We'll come back to them in a little bit.