

Tutorials for TXGenWeb

Basic CSS Tutorial

I've been asked about CSS quite often and I was asked to create a simple tutorial to help you understand what CSS is and how to use it. During and at the end of this tutorial, I'll provide you with any code you might want and a condensed version of the site. The main reason I learned CSS is because I'm lazy; I realized a long time ago that redecorating a site is a lot easier when there's no previous formatting on the pages. I still build new templates all the time but I only have to copy and paste the "meat" of the page to the new template and old formatting doesn't get left in it to make a mess.

Before you can understand CSS, you need to know a little bit about HTML and the difference between the two. HTML, in a nutshell, is a computer language used to build a web page. The only thing you really need to know is that HTML is mostly a collection of "containers" - I call them boxes. The boxes I use most often on my sites are headings (h1, h2 h3, etc.), divisions (div) and spans. With these three boxes and some line breaks (br), you can do a lot of amazing stuff. The commands used to create these boxes are called TAGS - open <div> and close </div>. Images (img) and line breaks (br) do not close.

CSS is a set of directions on how to decorate HTML. You can easily set colors, sizes and backgrounds for any box AND it's contents. If you don't like the defaults for a box, you can modify them (like the blockquotes I used here). In addition, you can use CSS to change the way a site looks on a cell phone or other small device. So STEP ONE: set up a basic web page in whichever way you feel most comfortable but DON'T decorate it all. No font color, no background, nothing. If you prefer, you can copy this one. Be sure to name it with an .htm or .html extension. (They're the same thing and it doesn't matter which one you use. I'm lazy so I don't use the L at the end.)

Once you have that page created and saved, close it in your HTML editor and open it in your text editor. You'll also want to open it in your favorite browser - the easiest way is to open the folder where you created the page, right-click the page name, choose Open With, then choose your browser.

If you don't have the option to Open With (older Windows versions), hold down the Shift key when you right-click.

Cleaning Up

I created my page in SeaMonkey and only typed one thing into it. The first thing I'm going to do is "clean it up" a little.

With a WYSIWYG, the editor does a lot for you and I don't like most of it. It's up to you to decide what is most comfortable for you.

I like all my text aligned to the left.

I have a head section that works for me and I modify it for all my sites.

I don't want ANYTHING inside the body tag.

I like my line breaks at the beginning of the line.

I've cleaned my page and made it the way I like it and I need to get it ready for decorating.

Adding Important Lines

Even if you never use it, it doesn't hurt to add the code for small devices. If you do decide to use it later, you'll be glad you don't have to go back and add it to all your pages.

A FAVICON is the little image you see in the browser tab and is used for your bookmarks. In new versions of HTML, you can name it anything you want so you can have several. I like them because I can quickly identify something in my bookmarks - if it has a Texas flag, it's most likely something from TXGenWeb or one of my TXGenWeb sites.

And finally we add the request to use our SEPARATE CSS file. CSS stands for Cascading Style Sheets. That means you start with the first CSS file, then go to any others that are listed, then any STYLE in the HEAD of the page and finally, any style that has been applied INLINE. It cascades (flows) from one to the next in a "downhill" direction. I'll go over these options in just a bit.

If you look at the images above, you'll see more examples of boxes - HTML, HEAD, TITLE and BODY are all containers. As a general rule, if you can put text in it AND it has a closing tag, it's a box. META tags are not boxes, no closing tag. Line breaks, no closing tag. You can't format tags that don't have closing tags (with a few exceptions and the only one you're likely to use is IMG, no closing tag and you can't add text to it, BUT you can format and decorate it).

There are TWO kinds of boxes. Most of our boxes will be BLOCK type. By default, they start on a new line and go all the way across your page.

This is a div

A span is an INLINE box. It starts wherever you put it and only takes up as much room as what is inside of it, you can't change it's size. An image is an inline box BUT you can manipulate its size. I don't want to confuse you so I won't get into WHY right now. This is a span. A hyperlink is also an inline box.

Creating A CSS File

A CSS file is just a text file that you've named with the .css extension instead of .txt. NotePad will allow this in most versions of Windows, I'm not sure about TextEdit. If you forget to change the extension (or can't), just open the folder where it's located and rename it there.

I know that, right now, all this code is going to look like gibberish and it can look overwhelming at first. But if you practice each lesson several times, you'll start recognizing items and being able to read the gibberish. I usually start a formatting sequence with a very basic line and then add the components one at a time until I have what I want. Trying to type it all into a class at one time is too complicated and, if I make a typo, I have a harder time finding it. My most common mistakes are using (instead of { and forgetting to put the semi-colon at the end of a value, especially when I put the whole thing on one line. I have a hard time remembering that there is no font-color - it's just color. Some formatting for fonts starts with text instead - text-decoration. I've spent hours trying to figure out why a complicated style isn't working only to finally notice that I forgot to put a semi at the end of one of the values or my ending curly bracket is actually a parentheses. So these days when I start a line that requires a closing I do that first <div></div> or .color {;} - especially links . Then I can add the complicated stuff without worrying that I'll forget to close. I'm also the queen of copy/paste; I have a file that contains the items I use most often and I can simply copy and paste them to whatever I'm working on.

Basically, you type in the property and the value you want. You don't need to memorize all the properties and values, you can Google something like CSS border property and choose the link for the W3 school to find a page that tells you all about it.

Let's Get to Work

The first part of your CSS file should contain the basics. The style for your body and links should go first. After those, you should always alphabetize your styles so you can find them easier. Since I'm lazy, I usually open a CSS file that works and copy the parts I want, generally the body, links and horizontal line break. So this page will use Arial font, the text size is 1 em (the default size for the visitor's browser), the font color is black, background is white and links are blue. I've also set the margin to 0 (zero); there is a default margin for the BODY tag of 8 pixels all the way around. I want my page to cover all the available space in the browser.

This is how I type my CSS files so that they're easy to look through. For the rest of this tutorial, some will be smashed together so I can fit more in my images and the pages aren't so long. I have a sample file here so you can see that I do take pains to make them neat. You can copy this sample to your CSS file and change it to suit your tastes.

You'll see that I've added two "classes" before I even get started - bold and color. Sometimes I want to emphasize a word or phrase so I always add these classes to my style sheet. I use the word color (instead of blue) for my class so I can change it without feeling silly if it's actually red or green. You can name your classes anything you want. I like descriptive names. If you use a multi-word name, either run them together "blueborder" or separate the words with a - dash, CSS doesn't recognize that "blue border" is one class; it has to be "blue-border". Now when I want something to stand out, I can wrap it in a span and add one or both of these classes. You'll want to add a class to center text.

```
.center {  
text-align: center;  
}
```

I also have the a img style in my link section <a> - now I don't have to type border="0" on each of my linked images.

If you have a very complicated class or special set of classes that you want to keep together (maybe a really nice menu), or you have to add something out of alphabetical order, you can add COMMENT lines to your CSS file. (I recently needed to find a class called content for an iframe but it was in the Is instead of the Cs so I added a comment line in the place where I expected to find it. Next time, I'll know to look under iframe.) So far, our CSS file is very short but it could possibly grow to thousands of lines for a large website so comment lines and alpha ordering are very useful.

Use a slash and an asterisk to open the line, then reverse them to close it. In a very large CSS file, you might want to use uppercase for your comments.

```
/* This is a comment line */
```

Remember the word LINE, a comment line is only ONE.

```
/*IF YOU NEED SEVERAL LINES FOR YOUR*/  
/*COMMENTS, YOU'LL NEED TO START AND*/  
/*end each line with this special*/  
/*combination of characters.*/
```

This also provides a way for you to make notes in your CSS file while you're learning. You can add them above and below a group:

```
/* My fancy menu starts here*/  
/* My fancy menu ends here*/
```

or you can add them to the right side of your styles:

```
.center {  
text-align: center; /*THIS IS FOR CENTERING TEXT INSIDE A BLOCK*/  
}
```

(Pretend that all my square brackets [] are actually angle brackets < > so I don't have to type < all the time.)

There are three ways to classify your CSS items. The first is a CLASS. [span class="bold color"]

In your CSS file, a class name begins with a period - .bold and .color.

The actual formatting begins and ends with curly brackets {}

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

the PROPERTY name (font-weight) is always followed by a colon and the VALUE (bold) is always followed by a semicolon.

```
.color {color: blue;}
```

Next is an ID. [div id="header"]

You can only use this id ONCE per PAGE but you can use it on EVERY PAGE. If in doubt, use class instead.

An id always starts with the pound sign - #header and #footer - and the rest of it is just like a class.

```
#header {background-color: blue;}
```

The last one is a TAG for use when you want every item of that type on your site or page to have the same properties, in which case, you don't need to add anything to the tags on your HTML page.

In your CSS file you type the name with no leading character - a:link and img.

```
a:link {text-decoration: none;}
```

```
img {width: 300px; height: 300px;}
```

This means NO link on your site will have an underline and EVERY image will be 300x300 pixels.

Before you do this, know that it can be a problem if you start uploading a lot of pictures of different sizes so you need to plan ahead. Other than links and the hr, I usually only use this in the HEADs of pages with tables where I want all the cells in the table to have the same properties. I used it on this tutorial to 'float' all my images to the right and put a red border on the left sides (and I've had to add style to the heads of a couple of pages to unfloat and remove borders).

td {padding: .5em;} is almost a standard for me - it adds just a little white space around the contents of every table cell on my page.

So any CONTAINER on your site can be formatted; all you have to do is add a CLASS or ID.

```
<div id="header"> <img class="thumbnail"> <h1 class="center">
```

File, Head or Inline?

None of them are "best" and they each have a purpose. I prefer to have a separate CSS file that covers most of the formatting on all my pages so I can change everything all at once. (Remember I'm lazy, if you put all your CSS on each HTML page, you have to change EVERY page.) For folks who just want to do a few simple things, or if you want one item on a page to be different, inline could be the best way to go.

I have a few websites that have more than one CSS file requested in the head. They're read by the browser in the order listed BUT the browser actually applies formatting as it reads it. So, if my first file calls for a blue background and the second calls for white, you'll end up seeing a white background and, on a very slow connection, you might see it flicker from blue to white. If you look at my Sherman County in Texas or my Woodruff County in Arkansas, you might see some of the flickering as the browser loads the Google fonts, which are actually called from CSS files.

So you use a separate CSS file for items that affect ALL your pages.

Whether you want to format a whole page or something on a page (like a special table) differently than everywhere else, you can put the information in the head of your page using the tags `<style> </style>`. These tags will go immediately above the closing `</head>` tag. This particular measurement will add a LOT of blank space inside a table cell.

If you find that you have a cell that needs less blank space than the rest, you can change it "inline". This will change ONLY the cell where you add it.

```
<td style="padding: 3px;">
```

Inline styling is also useful if you run across something that needs a little "extra" - perhaps a book title.

```
<span style="text-decoration: underline;">The Good Wife</span> produces The Good Wife
```

You can add as many properties as you want when working inline.

```
<span style="text-decoration: underline; font-weight: bold;">The Good Wife</span> - The Good Wife
```

You can even mix and match.

```
<span class="bold" style="text-decoration: underline;">The Good Wife</span> - The Good Wife
```

I think that covers all the basic stuff you need to know. There really isn't much here, it only looks that way because of the horizontal rules, images and my tendency to give long-winded examples.

Formatting Decisions

The first thing to do when creating a page with CSS is to decide how you want it to look. You should set your link colors, font color and size, background color and background image, if you want one. In CSS, there are two ways to set a color. If it's a simple color, you can use the name that's been assigned to it - white, black, blue, red, green, etc. You can find a list of those colors online. You can also use a hexadecimal code prefaced with a pound sign (#ffffff). If you use a 'color' that is one repeating character such as white, black or the 14 shades of gray between them, you can shorten it to three digits (#fff).

IF you want a background image on your page, you should wait until after you learn how to create a box with a solid background so you can read your text - probably after lesson 8 or 9.

To add a background image, add the line inside the body section of your css file. You can't call for it with a full URL but must use a 'relative' link. If your css file is in a subfolder, the URL will be different - ../images/background.jpg. My CSS is in the include folder and my background would be in the images folder. The ../ means go up one level in the directory structure and THEN look in the images folder for this file. More help with understanding relative links.

Similarly, if all your files are in the same folder, use
background-image: url("background.jpg");

The next thing I like to do is decide how I'm going to use my headings. I don't usually use the two largest, h1 and h2, but I do use h3 and h4 quite often and I prefer them to be centered. If I use h5, I like it left-aligned. Go down to where the hr styling is and, just above it, add
h1, h2, h3, h4 {text-align: center;}

Now go to your page and add two headings. Save the page.

```
<h1>This is a basic web page.</h1>  
<h5>First Section</h5>
```

Go to your browser and refresh the page (F5); you should get something like the image below.

For any BLOCK container, you can simply use 'text-align: center;' inside its tag to center everything in it.

Add A Table

The really cool thing about tables and css is that you control every piece little of it. The really bad thing about tables and CSS is that you have to control every little piece of it. Since I'm a lazy person, I've come up with my own way of formatting tables with the least amount of effort.

In HTML, all you had to decide is whether or not you wanted a border and add a number to indicate the thickness of the border. In CSS, it's a little more complicated. BUT instead of choosing border and color information for the TABLE, you format the CELLS instead.

Add this basic table to your page (copy and paste). Then look at your browser and refresh the page (F5). You won't see anything because the cells are empty, there's no border and no background color - nothing to see. But it is taking up space on the page.

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

Now go to your CSS file, to the end of the page and add some style to your table cells. There are three main pieces of a border style, the thickness of the line (width), the style and the color, and you can set all three pieces in one statement. In order to use border width and border color, you MUST use a border style.

For border style, there are many choices - dashed, dotted, double, groove, inset, outset, ridge, solid or none. For some reason, they also gave us the choice of hidden. If you don't want to show a border, the easiest thing to do is don't add one. Perhaps, if you're experimenting with your border and don't want to delete what you have when your page goes live, you can hide it until you're ready.

For width, you can use a word (thin, medium or thick) or a measurement (3px, 1em, 1in, 2%, etc). There are numerous units of measurement in CSS. The three in the example are pixels, ems, inches and percentage. Some units are absolute, like pixels and inches, they have a specific width. Some units are relative, like ems

and percentages, their size is RELATED to something else. I don't recommend them for border thickness but they have good uses in other areas.

Border color is just about anything you can imagine and translate to hexadecimal. You can ask Google for the hexadecimal equivalent of salmon or even desert camo (three different colors).

I'm keeping it simple for this tutorial, you can experiment all you like.

```
td {  
border: thin solid black;  
}
```

Save your page and your CSS file and refresh your browser.

Since there is nothing IN our table, no size set and no table border, those little tiny boxes are our table cells.

So I put something in my boxes. The browser adjusts the sizes of the cells to accommodate the TALLEST item in each row and the WIDEST item in each column. Now let's learn about cellpadding and cellspacing, remember them from HTML?

Cellpadding and Cellspacing

In CSS, we don't have either of these specific options. We do have PADDING, which works INSIDE any box and BORDER-SPACING, which works inside a table. If we want to add some padding INSIDE something, it's quite simple. I don't like the way my h5 is right up against the left side of my browser, so I'm going to add some padding to the body section of my CSS file.

```
padding: 1em;
```

Save and refresh.

Now there's some blank space between my information and the left side of my page.

I can add this same property to the cell properties at the bottom of the CSS file.

```
td {  
border: thin solid black;  
padding: 1em;  
}
```

Notice that the padding is added to all four sides of the cell, top, bottom, left and right. We'll discuss applying properties to one side of a box a little later.

Now we can add some space between the cells or merge them to a single thin line. I want a single line between my cells but I'll demonstrate the border spacing first. At the bottom of your CSS file, just above td, add

```
table {  
border-spacing: 1em;  
}
```

Save your CSS file.

Refresh your page in your browser.

Now add your border (copy and paste from the td) to the table.

Save your CSS file and refresh your browser.

I want a single line between the cells of my table so I'm going to REPLACE the border-spacing line with

```
border-collapse: collapse;
```

Page Borders

Now that you've learned about borders, you might want to add a border around your page. I do add borders on most of my sites but I usually add a box around my content and add the border to that instead. You can try it both ways. To add a border to the page,

go back up to your body section in the CSS file and paste that border in and then change the width from thin to 10px.

Notice that the border stops at the bottom of your content, because that's where your page stops. While block containers go all the way across the page, they're only as tall as the tallest item inside them. That's why I prefer to add a box around my content and add the border to it. Then I can set the background of my page to the same color and have a short page or pretend that the blank space is just more decoration (ah, laziness).

To add a box, go to the top of your page and find the `<body>` tag. Add a line just under it and type (or paste) `<div id="outer-wrapper">`.

Go down to the bottom of your page and find `</body>` and add a line just above it and type `</div>`. I like to go one step further and use `</div> <!-- outer-wrapper -->`. This tells me what that closing tag is for and comes in handy if I have 5 or 6 of them down there. The exclamation point and dashes make it a COMMENT line that isn't visible on your page. Scroll down to the bottom of this page for a view of our final set up.

Now we'll format it in the CSS file. Remember that this is an ID, not a CLASS so it starts with a # and you want to insert it into your file in alphabetical order (before table).

```
#outer-wrapper {  
width: 90%;  
border: 6px double red;  
}
```

Refresh your browser and have a look. We'll get into how to center this box a little later.

We added a very narrow padding to our page but now this border is INSIDE that padding and my h5 is up against the border so I want to add some padding to the wrapper.

```
#outer-wrapper {  
width: 90%;  
border: 6px double red;  
padding: 1em;  
}
```

Now there's a nice white buffer between both my h5 and table and my border.

NOTE: Some borders (including this one) have a minimum width required. If you try to make this border too narrow, you won't get the separation and it'll look like one thick border. The groove, inset, outset and ridge also require a little extra width to make them look 3D - another good experiment project.

I like to get really fancy sometimes so I also use an #inner-wrapper. I ALWAYS add these to my pages. I may not use them but they're there if I want them when I redecorate. Add the `<div id="inner-wrapper">` immediately UNDER the outer wrapper on your page and put its `</div> <!-- inner-wrapper -->` immediately ABOVE the one for the outer wrapper - like putting a box into a bit larger box and it's called nesting.

Add a few more boxes...

Immediately under the inner wrapper on your page, add `<div id="header">` and immediately below it put its `</div> <!-- header -->`

Under that one, add `<div id="content">` and put its closer at the bottom immediately ABOVE the one for the inner wrapper.

```
</div> <!-- content -->
```

Now our headings and our table are all inside the content part of our page. The other good thing about using the content box is that the actual page contents (as opposed to headers and footers) is easy to find and copy. ONE MORE. We need a footer. It's a good place to add our copyright, email address, search engine, last update, or anything else that is needed on every page.

Just UNDER the CLOSER for the CONTENT, add `<div id="footer">` and put its closer on the line after it.

```
</div> <!-- footer -->.
```

If you want a really busy page, you can create a border for each of these boxes and add a background color. Each one can have a different border and different background. You can also use the background image statement from the body section if you want a background image in every box. This is where you set a white or very pastel color on your CONTENT box so that you have a place to put legible text. (And now you can add a background image to the page - from lesson 5.)

Floating!

One of the hardest things for me to learn in CSS was floating.

The biggest challenge in CSS is learning WHAT to add a property to. You may be adding a property correctly - to the wrong box. This is doubly challenging when floating a box.

Back when I first started building pages and wanted to put two items side-by-side, I would create a table with two cells and put one item in each cell. OR if one item was a picture, I could align the picture to one side or the other and the other stuff would move up beside it. With CSS, we use the FLOAT property.

When you assign a float to an object, you're instructing it to stick to a side of the space available and EVERYTHING under it in the HTML page will FLOAT up and stop at a level even with the top of the floated object. So you're not actually floating the object, you're floating the stuff that comes after it. If you only want to float some of the stuff, you have to turn off the float for something underneath it. Before I tell you how to float, I want to tell you how to stop it so it doesn't get out of hand.

Most of the images in this tutorial are floated to the right. The reason I have horizontal rules between my sections in this tutorial is because I use them as a container to unfloat everything. The property (for this page) is `clear: right;`

```
hr {
clear: right;
margin: 2em 10em;
color: #008080;
background-color: #ff0000;
border: 1px solid #ff0000;
}
```

When I build a template that has a menu down one side, it's floated and I use the footer div to unfloat the footer and keep it at the bottom of the page.

You can also stick in an empty div to unfloat `<div style="clear: both;"></div>`. The important thing to note is that you might have things floated on both sides of the page and you only want to stop the float on one side. You can clear the left, right or both.

Open this sample file and copy and paste the text into your page UNDER the closing tag for the table and above the closing div for content.

```
</table>
Paste here.
</div> <!-- content -->
```

Save and refresh.

I want to format our table so that the text will float up beside it; it shortens the distance people have to scroll.

Go to the table properties at bottom of your CSS file and add the line `float: left;`

Save and refresh.

We added padding to the inside of our page and our outer-wrapper so that text wouldn't be smashed up against the side of the page. Now we're going to use MARGIN to add some space on the OUTSIDE of our TABLE (use the table section at the bottom of your CSS file).

```
margin: 1em;
```

But I don't want ALL that text to wrap around the table this time. I only want the first few sentences up there.

In your page, find the end of the fourth sentence, push it down two lines and, in the blank line, put the empty div from above `<div style="clear: left;"></div>`

Floating is especially useful for having two COLUMNS on a page.

Add a Property to PART of a Box

I have my text added to my page and I have part of it floated next to the table but now, because of the margin we put around the table, the text is sitting too high and looks odd. We can level it with the top of the table by taking away the margin at the top of the table.

When we use `margin: 1em;`, we're adding that margin to all four sides of the table. We really only need the margin on the right side and across the bottom for this particular table. When you use multiple values in a property line margin or padding, the browser understands them to read in a clockwise direction, starting at the top - top, right, bottom, left.

In the table section of your CSS file, change your margin to `margin: 0 1em 1em 0;`

That's still not quite what I want for the text but the table has moved left and up to the limits of its space. Spacing around a floated object can be tricky but we're going to handle this issue by wrapping that floating text in a div and adding an inline margin above it.

Just under the `/table` in your page, add a new div `<div style="margin-top: 5em;">`. Put its closing `/div` at the end of the fourth sentence.

In the first example, we added multiple margins in one statement. In the second, we specified which side of the box we wanted to change. We can use `top`, `right`, `bottom` or `left`. Padding works the same way - `padding-bottom: 1em;` - and there are several other properties, including `border`, that can be specified in this way. So we can set each of the four borders on our page in a different style, width or color - or all three.

```
border-right: 10px dotted #ff00ff;
```

Sometimes the order in which you list things in your CSS file matters. The links (a) pretty much HAVE to be in the order listed in the original CSS file from Lesson 3. Sometimes the properties also have to be listed in the right order. To add that right border, I have to put it AFTER the black border OR remove the single black border statement and add separate ones for each of the other three sides. This, again, is because your browser loads each property as it comes to it. So leave the black margin and add the dotted one under it. The browser will load the black and then load the dotted. If you put the dotted first, you'll see the black.

Centering

If you want to center something horizontally that's inside a BLOCK-type box, all you have to do is add your text-align: center; class to the tag for the box - <div class="center">. Centering the box itself takes a little more work and depends on how you set up your page.

If you want to center everything on your page, you can add text-align: center; to the body section of your CSS file but if you have any BLOCK boxes they won't move because they take up the whole width of your page. Everything inside of them will be centered inside them but not necessarily centered on the page. Even the contents of our table will be centered because cells are INLINE boxes.

Our outer-wrapper is no longer taking the full the full width of the page but, because it's a BLOCK, it still won't move unless you add a margin. The handy tool we have for that is margin: auto;. Using this, if our page is 1000 pixels wide and our outer-wrapper is only 800, the browser does the math and moves the outer-wrapper over so that it's centered - auto margin simply calculates the difference in width between the two objects and divides by two. So now our outer- wrapper has a margin of 100 pixels on both sides. BUT margins, padding and borders take up some of that space so, since we have all three on either the page or the outer-wrapper, the formula for calculating the difference is a lot more complicated. Not to worry, computers were created to do complicated math problems and they do them at the speed of light.

In order to center something using the auto margin, you have to assign a width to it. It can be a relative width like a percentage but it can't be an auto width (yes, we have that too in some cases). We've already assigned a width to our outer-wrapper, so go down to its section in your CSS file and add margin: auto;. Keep in mind, that when we only use ONE value for a margin, it affects all four sides but the default for auto margin is 'don't do anything to the top or bottom'. So you go UNDER that line and add a margin for the top or bottom or both if you want. Now our outer wrapper is balanced on the page.

Centering an inline box is simple as long as it's inside a block box. You can center it without but it's more complicated than we want to go into in a basic tutorial. So, if you want to center a link in the middle of your page, you'll need to add a div, a paragraph, a heading or similar block, put the link inside that and then use text-align: center; on the block box.

<div class="center">Link</div> will produce

Link

One more way to center something large like our outer-wrapper that takes up most of our page, is to add padding to the body and let the outer-wrapper fill the rest of the space instead of setting a width for outer-wrapper. If you have 200px of padding all the way around your body (or just the left and right sides), the outer-wrapper will LOOK centered because it starts at the pixel next to the left padding and goes all the way over to the pixel next to the right padding.

Centering Captions

There is a special tag for captions in HTML and CSS but it seems overly complicated to me. The easiest way to put a caption on an image (or any other item except a table) is simply to put them both in a div. I've added a light border so you can see what's happening. A standard caption would be `<div>
My dog, Spot</div>` and produces.

My dog, Spot

If you want them centered, add your center class to the div.

```
<div class="center"><br>My dog, Spot</div>
```

My dog, Spot

To center only the caption, add the WIDTH of the IMAGE to the DIV to keep it from stretching across the page.

```
<div class="center" style="width: 197px;"><br>My dog, Spot</div>
```

My dog, Spot

If your image has a border or margin, or if your div has a border or padding, you'll need to add those measurements (both sides) to the width of your div.

```
<div class="center" style="width: 209px; padding: 5px;"><br>My dog, Spot</div>
```

You can put the caption on top

```
<div class="center" style="width: 197px;">My dog, Spot<br></div>
```

My dog, Spot

Or leave out the line break, float the image, and widen your div to put the caption on one side or the other (only useful for very long captions). Don't forget to turn off the float!

```
<div class="center" style="width: 297px;">My dog, Spot</div>
```

This is a picture of my dog, Spot. He's cute and he likes to chew on his bone and play ball with me. Tomorrow we get to go to the park with my mom!

Decorate Your Header

Having the same header or a specific logo on every page on your site lets visitors know that they're still on your site.

The title for our page is already there as a heading, you can change the wording if you like. First we need to move it into the header section of our page. (Cut and paste it.)

Now we create a formatting group for our header.

The font is already a nice size for a website title but we can make it a different color.

Add a bottom border to separate it from the rest of the page.

and some padding to balance it.

Use what you've learned to do more to your header - whatever you want.

And Your Footer

You'll want to add some text in the footer section of your page and create a footer section in your CSS file.

Special characters for HTML are also available online and the copyright equivalent is ©.

How it looks. Header and footer added and decorated.

Negative Values

In some cases you can use a negative value to style something on your page, most notably, margins. My sample page has two headings in the header and they come with a default margin that makes them have what looks like a blank line between them. I want them to be closer together. What I can do is manipulate the bottom margin on the h1 or the top margin on the h2. I've already added a left margin to the h2, so I'll move it up. First, I'll try setting the top margin to 0, that works some of the time. `margin-top: 0;`

You might also try setting the BOTTOM margin of the h1 to 0. `margin-bottom: 0;`

In this case, it didn't work so I'll start with a low negative value. Since my default font-size for the page is in ems, I'll use `margin-top: -.5em;`. Notice that there's a decimal in front of the 5, so I'm changing the margin to 1/2 em - half the height of the text INSIDE the heading. While my page has a font size of 1 em (about 16 pixels), the default value for a second-level heading is 1.5 em (about 24 pixels) and since I'm changing the heading, not the page, the value is based on the heading's font-size; -.5 isn't enough and I've settled on -.9.

My header is fixed and I'm ready to create a template from this sample.

Creating A Template

Once you have a page looking exactly the way you want it, you'll want to create a template. If you added the extra boxes in Lesson 8, creating a template and/or server-side includes is simple. COPY the text in your HTML file into a new file and name it template-1.htm. The 1 means it's for the main directory.

If you want the SSIs, cut everything BETWEEN

```
<div id="footer"> and </div> <!-- footer-->
```

paste it to a new file and save it in your includes folder. Do the same thing for your header. Leave the divs on the page.

BETWEEN your divs, add the calls for the SSIs

```
<div id="footer">
```

```
<!--#include virtual="include/footer.htm"-->
```

```
</div> <!-- footer-->
```

Delete everything BETWEEN `<div id="content">` and `</div> <!-- content -->`.

Save the template again.

To create a template for any subdirectories, copy and paste your -1 template and add your `../` to any of the CALLED files.

```
<!--#include virtual="../include/footer.htm"-->
```

Never use your template without first copying it to a new file.

This tutorial copyright ©, Gina Heffernan, 2019.