

Learn code!  
Impress your nerdy  
friends!



# “Get Your Geek On” with.....

# HTML

## Book 1

A tutorial for beginners or  
those just needing a refresher!

By: Teresa Williams



## Getting Started

---

Hypertext Markup Language (HTML) is considered to be the language of the World Wide Web. Did you know that web pages are nothing more than TEXT files? Within the text are special commands called markup that instruct the browser to do certain things, such as display an image, create a table, or jump to another page by way of a hyperlink. This tutorial is designed to get you started with easy, entry level **HTML**.

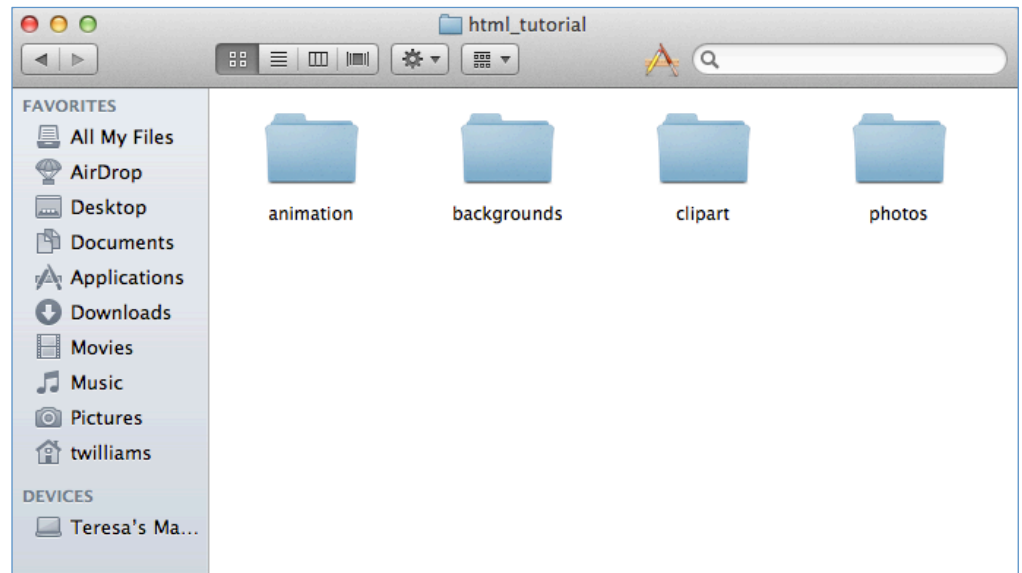
"Markup" refers to the commands for the browser that appear in between angle brackets. A typical "markup" might be: ``. This is an image source command that instructs the browser to look in the "clipart" folder for a file named "dog.jpg" and display it on the web page. We will go through, step by step, how to write your HTML files with a variety of markup, also referred to as HTML "tags."

A web programmer needs two pieces of software to get the job done, a **text editor** to create the web page, and a **browser** to view it. We will use a very basic text editor for our programming. I use TextWrangler on my Mac. It's free and can be downloaded online. For Windows users, you can simply use Notepad, an accessory program that comes with Windows. Take a moment to download or locate your text editor.

Next, **organization** of your files is crucial as you begin learning HTML code. Create a folder on your computer where you can save all of your files. Within this folder, create folders to store our practice files. I recommend the following folders: clipart, animation, photos, and backgrounds. Book 1 (this tutorial) covers the basics and will not utilize images (and therefore these folders), but you will be organized and ready for the 2nd tutorial, "Images and Backgrounds." **VERY IMPORTANT:** do not use any spaces or "weird" characters in your file and folder names - EVER! Only use letters, numbers, and underscores ( \_ ). Also, I recommend sticking with lower case. Upper case letters will work fine, but I will consistently use lower case in this tutorial. In general, web programmers tend to use lower case.

At right is a screenshot of my folder. Notice how I don't have any spaces or "weird" characters in my folder names?

Now, a little bit about browsers: You will view your web pages in a browser that you frequently need to refresh to see changes in your



code. Most browsers work beautifully in this respect, but some versions of Safari make it difficult to refresh by requiring you turn on the Developer's Menu and empty the cache each time. Not fun! So, to make life easier, I recommend you use Firefox, Chrome, or Explorer. Of course, software is changing all the time, so you might find that your version of Safari works great. So, by all means, check it out after you've had a little experience with code and know how this process should work.

Before we start writing commands, a little history about HTML: There have been several versions of HTML over the years. Currently, the standards are moving toward HTML5. A lot of code that you will see on the Internet is actually HTML 4.01 or transitional HTML with elements of both, or elements of something called XHTML. My code will be set up with the basic commands found in HTML5 and HTML 4.01. In order to make this an easy venture, I will also use some tags that are being deprecated (slowly phased out) in favor of somewhat more complicated Cascading Style Sheets. I will try to indicate when this is the case. CSS will be the subject of a future book, but it's a bit overwhelming to start off with it your first time programming. Just remember, HTML is constantly evolving, and this will never change.

Are you ready? Here we go!

## The Basics

---

These are the basic commands that will appear in every web page file that we create. This will be referred to as the "template code" from now on.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>html template</title>
</head>
<body>
Your web page content goes here
</body>
</html>
```

Here's a breakdown of what it all means:

**<!DOCTYPE html>** *This tells the browser that an HTML5 file is coming. DOCTYPE (this is one case where it is typically written in upper case) commands for older versions of HTML look much more complicated! Without the DOCTYPE command your file would still work, but it would be in "quirks" mode, meaning the browser doesn't know exactly which type of HTML file is coming, but will do its best to display it properly.*

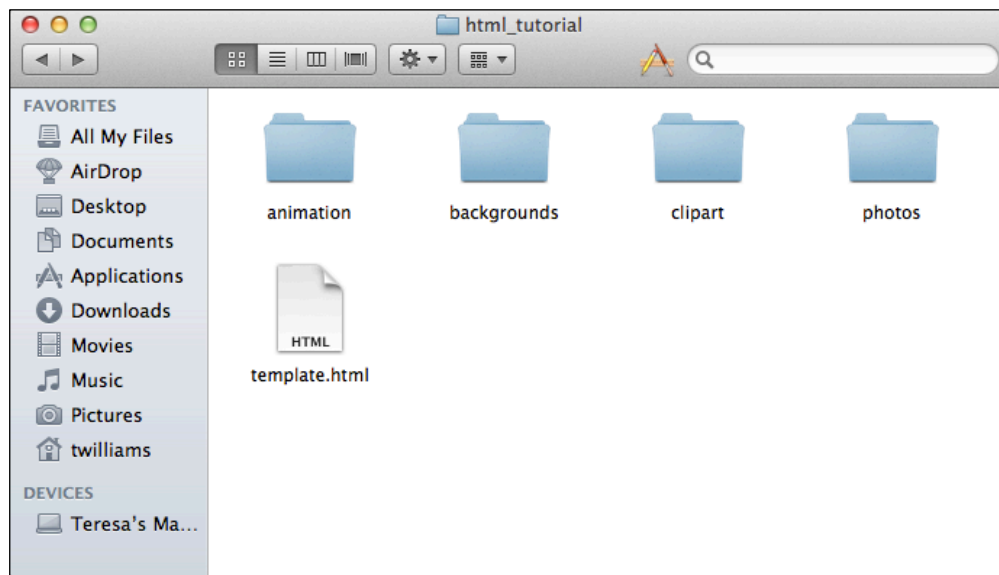
**<html>** Signals to the browser the beginning of a web page file.

**<head> ... </head>** The slash (/) means it's a closing or ending command (tag). A "tag" is another name for an HTML command that appears between angle brackets. The "head" section of a web page contains the "title" which appears in the title bar or tab of the browser window. The words between the <title> tags also show up when you bookmark your site. The "meta" tag is found in the head section too, and contains information about your file, such as the fact that it is a "text" file. Again, your file would probably work just fine without the <meta>. The "head" section can also hold lots of important commands when you get into more advanced programming (such as Cascading Style Sheets and Javascript).

**<body> ... </body>** This is where all the action takes place! Whatever appears between the "body" tags will actually show up on your web page. My template file above has placeholder text - "Your web page content goes here." So, that's exactly what we would see on our web page, just those words. All link, image, list, table, and font commands will go between the body tags.

**</html>** You guessed it - this tells the browser that it has reached the end of our web page file.

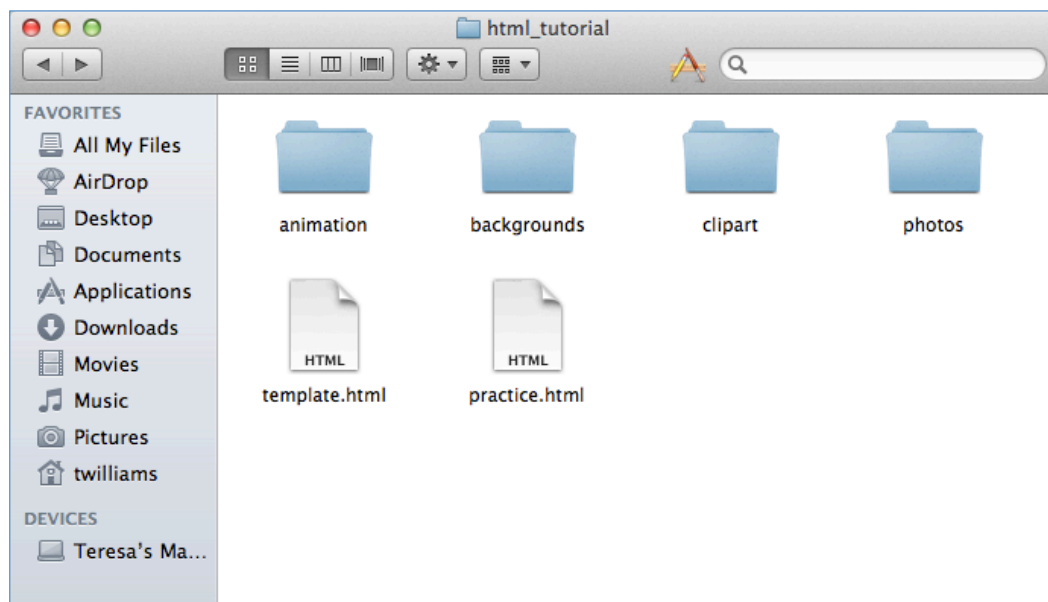
Our first web page: Always start with the template code. You may want to type it in Notepad or TextWrangler and then save as **template.html** in your folder. Be sure to use .html as the file extension! Double click on this file in your folder and see if the placeholder text appears before moving on. If it doesn't, check your closing title tag </title>. Whenever a web page comes up blank, always check the title commands first. Still not showing? Check line by line for typos: missing quotes and angle brackets are common mistakes. Once the template.html file is error-free, you can open this as a starting point for all new files. Be careful not to save over your template, though!



Browsers ignore all paragraph returns, tabs, and extra spaces, so we can space out our code to make it more readable to us as programmers. I like to tab in the code between head tags. Make the following changes to your template code and save as **practice.html** in your folder.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>Practice web page</title>
</head>

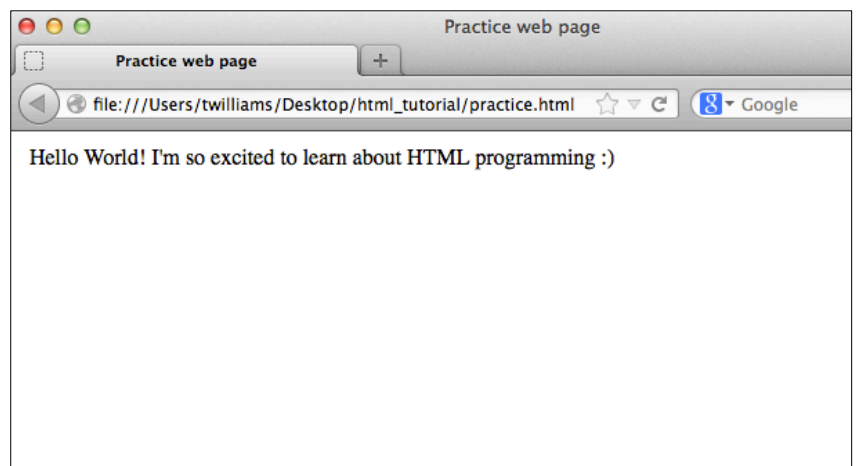
<body>
Hello World!
I'm so excited to learn about HTML programming :)
</body>
</html>
```



Go to your folder and double click on the **practice.html** file. It will open in a browser and this is what you should see:

#### Note:

The title appears in the browser tab, NOT the window. The two statements appear on the same line, even though I pressed return between them. The browser ignores returns and must be told to produce a line break. So, how do we force a break?



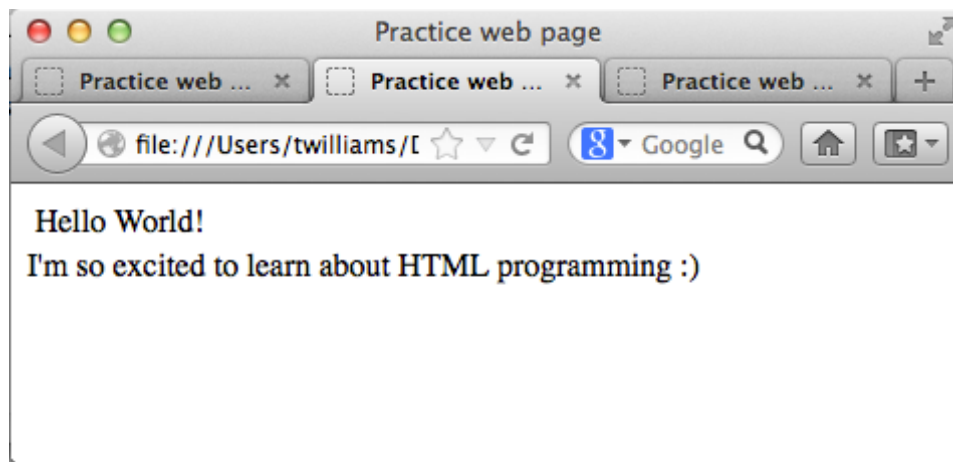
Try adding the following changes highlighted in yellow to our existing code.

\* Wait - did you accidentally quit Notepad or TextWrangler? Double clicking your file takes you to the browser, so how do you get back to code? Try right clicking (Windows) or control clicking (Mac) on the file icon and then choose OPEN WITH ....(Notepad or TextWrangler). Now, for those changes ..... simply add the `<br>` tag as shown below:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Practice web page</title>
</head>

<body>
Hello World! <br>
I'm so excited to learn about HTML programming :)
</body>
</html>
```

SAVE in the text editor and REFRESH in the browser. The refresh button is typically a curvy arrow in the browser's button bar at the top. Here is how it should look now that we've added the `<br>` tag.

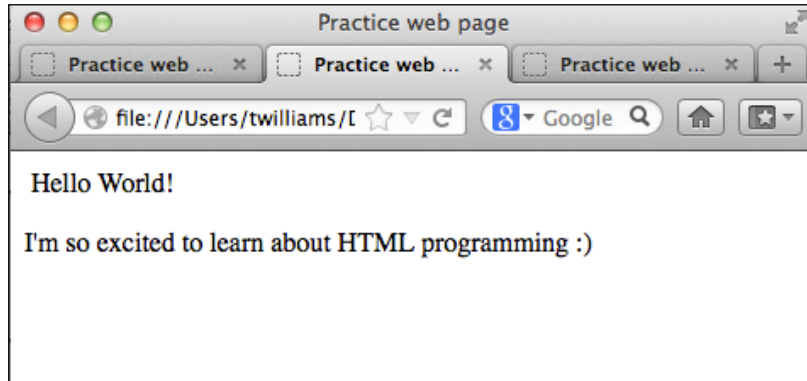


**`<br>`** *Instructs the browser to move down to the next line, creating a "line break." The tag `<br>` does NOT have a closing tag. You may see it written as `<br />`, with the extra space and closing slash included in the tag. This is the older transitional style of code.*

Now, delete the `<br>` and add the `<p>` and `</p>` commands as shown:

Hello World!

`<p>`I'm so excited to learn about HTML programming :) `</p>`



`<p>...</p>` Instructs the browser to move down to the next line and skip a space. This is a "paragraph return." Unlike `<br>`, the `<p>` tag does have a closing tag. HTML is a little goofy like this, but after a short time, you will get the hang of it.

## Changing Text Size

---

There are two ways, in basic HTML, to specify text size. They don't provide as much flexibility as CSS (Cascading Style Sheets), but it will be enough for our purposes in this tutorial.

(1) **Heading sizes** `<h1>` through `<h6>`: all text in one line will appear in the heading size specified. All `<h?>` tags have a "built-in" line break.

<code>&lt;h1&gt;...&lt;/h1&gt;</code>	Largest heading size
<code>&lt;h2&gt;...&lt;/h2&gt;</code>	Larger heading size
<code>&lt;h3&gt;...&lt;/h3&gt;</code>	Medium heading size
<code>&lt;h4&gt;...&lt;/h4&gt;</code>	Medium/small heading size
<code>&lt;h5&gt;...&lt;/h5&gt;</code>	Smaller heading size
<code>&lt;h6&gt;...&lt;/h6&gt;</code>	Smallest heading size

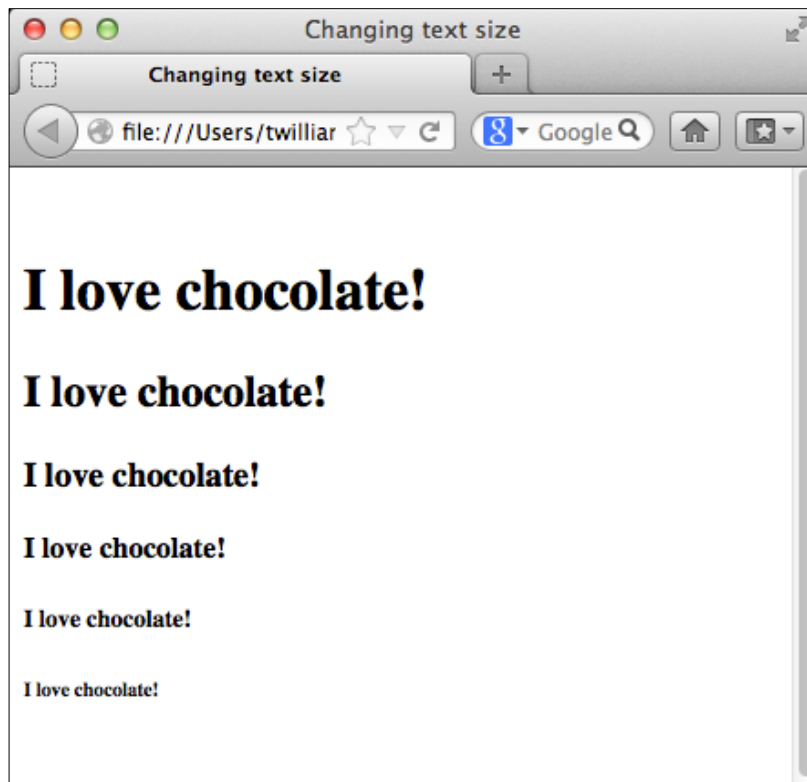
Open your template file and save as **headings.html** in your folder. Be sure to make use of copy and paste when creating this file!

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Changing text size</title>
</head>
<body>
```

```
<h1>I love chocolate!</h1>
<h2>I love chocolate!</h2>
<h3>I love chocolate!</h3>
<h4>I love chocolate!</h4>
<h5>I love chocolate!</h5>
<h6>I love chocolate!</h6>
```

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

Save again and open in your browser. This is how it should appear:



(2) **Font tag:** Although this tag is being deprecated, it is incredibly easy to use and still works in all browsers. The `<font>` tag lets you specify font sizes 1 through 7, with 1 being the smallest and 7 the largest (yes, that is the opposite of heading sizes!). The nice thing about the font tag is that you can have words or letters of multiple sizes on the same line. The `<font>` tag makes use of several attributes; size, color, and face. We will look at size here, and then color and face very soon.

Open your template file and save as **fonts.html** in your folder. Be sure to make use of copy and paste when creating this file!



```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>The font tag</title>
</head>
<body>
  Default text <br>
  <font size="7">This is the largest font size!</font>
  <font size="1">This is the smallest! Can you read it?</font>

</body>
</html>
```

Save again and open in your browser. This is how it should appear:



## Stylizing Text

---

**<b>...</b>** *Instructs the browser to display the text as bold.*

**<strong>...</strong>** *Instructs the browser to display the text as strong, which for most browsers (if not all) looks just like bold.*

**<i>...</i>** *Displays text as italicized, slightly slanted.*

**<em>...</em>** *This stands for "emphasis" which displays text as italicized.*

**<u>...</u>** *Instructs the browser to display the text underlined, which can be very confusing for the person viewing your page. Underlined text is usually associated with links. Avoid using this.*

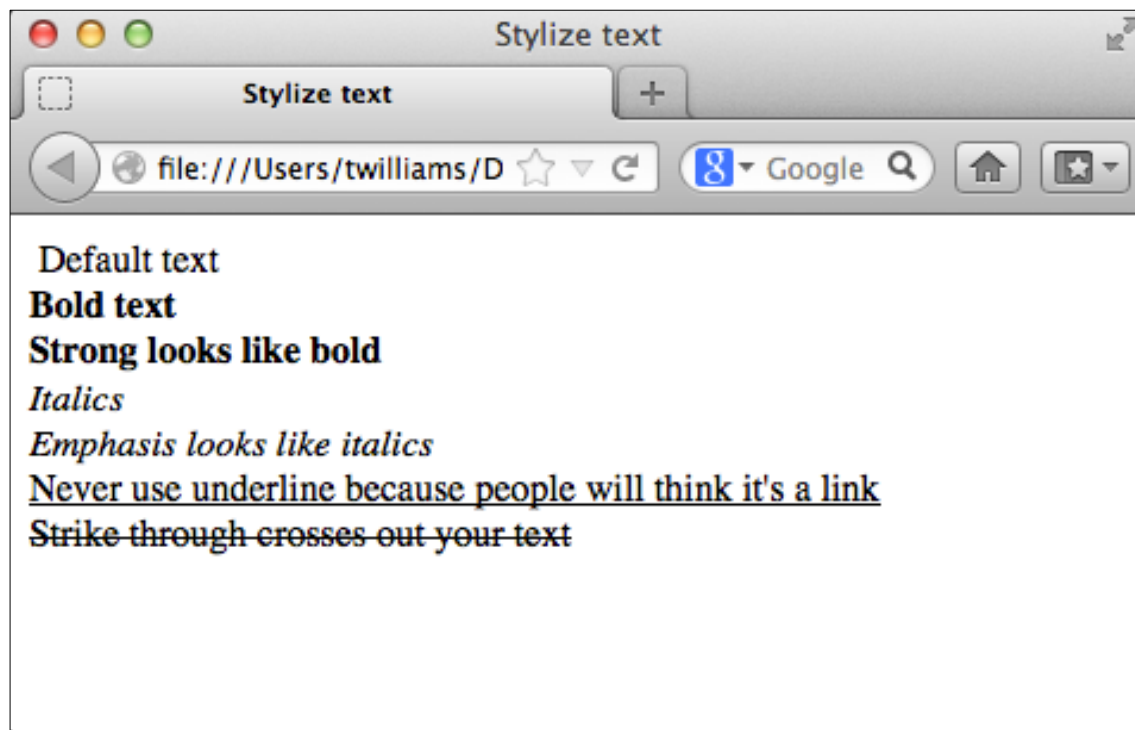
**<strike>...</strike>** *This stands for "strikeout" which displays text with a line running through the center of it.*

Open the template code again and save as `stylize.html`. Try each of the tags as shown below:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Stylize text</title>
</head>
<body>
```

```
Default text <br>
<b>Bold text</b> <br>
<strong>Strong looks like bold</strong> <br>
<i>Italics</i> <br>
<em>Emphasis looks like italics</em> <br>
<u>Never use underline because people will think it's a link</u> <br>
<del>Strike through crosses out your text</del> <br>
```

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



## Changing Colors with Hexadecimal Code

---

Hexadecimal Code (oh yes, it's just as nerdy as it sounds!) is a base 16 numbering system used to represent all the web-safe colors used in web page design. You can also use standard color names, but that's not nearly as much fun! More about that later. Take a quick peek at the HTML color chart I made and included on the LAST TWO PAGES of this tutorial. Don't be alarmed, though! It will all make sense very soon.....

Our base 10 number system is called the "decimal" system, with 10 digits, 0 through 9. For a base 16 number system, we have the 10 digits 0 through 9, but what about the other 6 needed to reach 16? That's where the letters come in. Letters "a" through "f" are used to stand in for numbers 10 through 15. You might see a crazy looking code for a shade of blue that looks like **#0033FF**. Think of the hexadecimal code as representing how much **RED**, **GREEN**, and **BLUE** are mixed by your RGB monitor to create your color. The first two digits (00) represent the level of red, which in this case is zero. The next two digits represent the level of green (33), in this case a small amount. The last two digits (FF) represent the level of blue, in this case "full-on" blue, the largest amount.

Monitors use the RGB color model to display color. This is a completely different color model than the one we learned in elementary school, where red, blue, and yellow are the primary colors that mix to make all the other colors. With RGB, full on red, green, and blue (**#FFFFFF**) is actually white. The absence of red, green, and blue (**#000000**) is black. It's hard to imagine the colors mix in this way, but they do.

The "purest" colors would be:

<b>#ff0000</b>	Red
<b>#00ff00</b>	Green
<b>#0000ff</b>	Blue

Notice how I have expressed them with lower case just above, but earlier with upper case letters. It works either way. You will also notice a duplicating pattern if you take another look at the chart at the end of this tutorial. Some web programmers take advantage of a shortcut for duplicating digits.

Some shortcut examples:

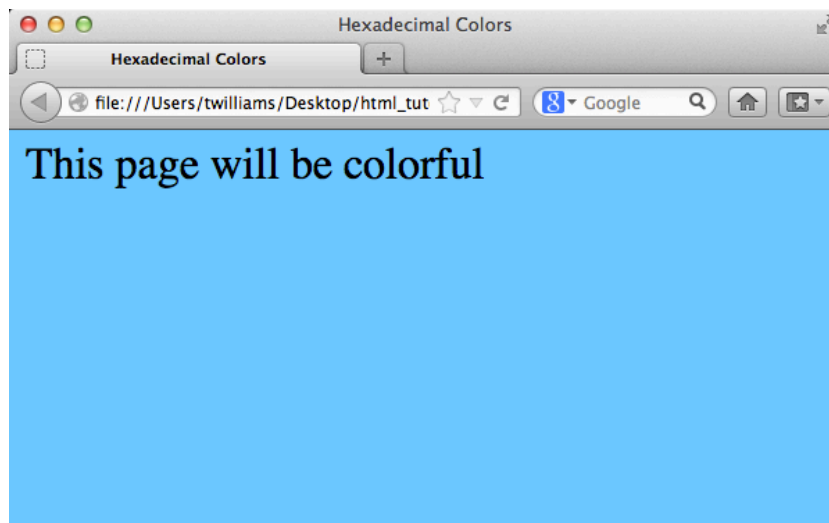
<b>#3366FF</b>	<b>#36F</b>
<b>#000099</b>	<b>#009</b>
<b>#CC3366</b>	<b>#C36</b>

I seldom use the shortcut, but feel free if you like. Notice that the number symbol ( **#** ) always appears in front, though.

Now, let's use colors! How about changing background color first with a simple `<body>` attribute, `bgcolor` (stands for background color).

Start with your template code and save as **colors.html** in your folder. When adding the bgcolor attribute, don't forget the space in front of it. Browsers can't interpret your attributes without spaces separating them)

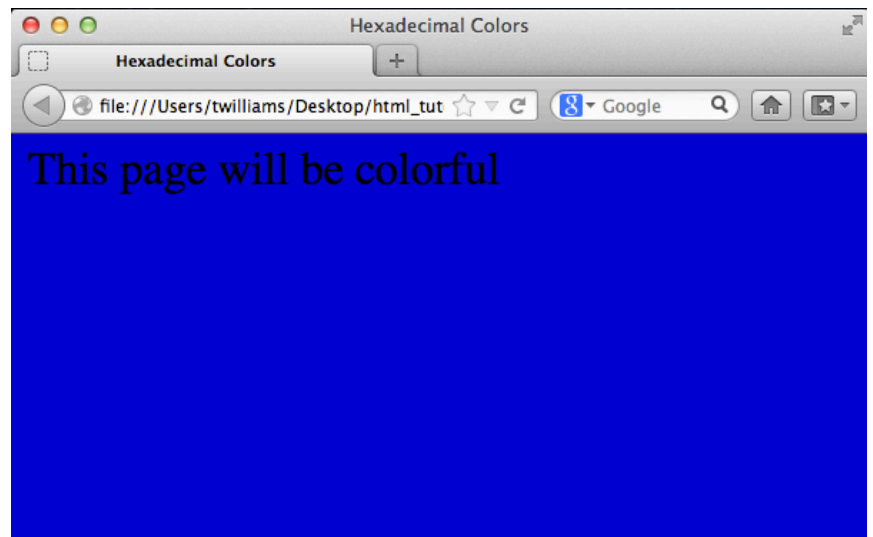
```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Hexadecimal Colors</title>
</head>
<body bgcolor="#66ccff">
  <font size="6">This page will be colorful</font>
</body>
</html>
```



This looks fine for my choice of color, but what if I go darker? Try the following change, then save and refresh:

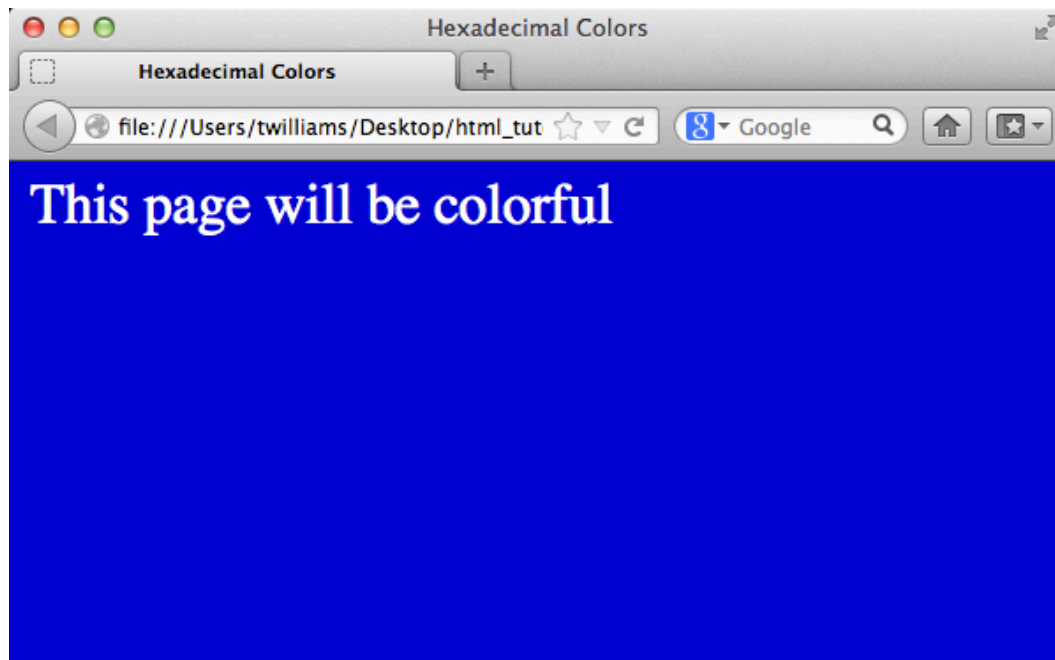
```
<body bgcolor="#0000cc">
```

It's much more difficult to read the text now. Let's try changing the default text color with a simple **text** attribute added to **<body>**.



Try the following change, then save and refresh (don't forget the spaces between attributes!):

```
<body bgcolor="#0000cc" text="#ffffff">
```



Ahhh! Much better.

Make sure you are not adding extra <body> commands. There should be only one <body> tag in a web page file! Your code should now look like this now:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Hexadecimal Colors</title>
</head>
<body bgcolor="#0000cc" text="#ffffff">

<font size="6">This page will be colorful</font>

</body>
</html>
```

In the next section, we will change the colors of some of the letters/words in *THIS FILE*, so don't close it just yet.

## Formatting Font Colors

---

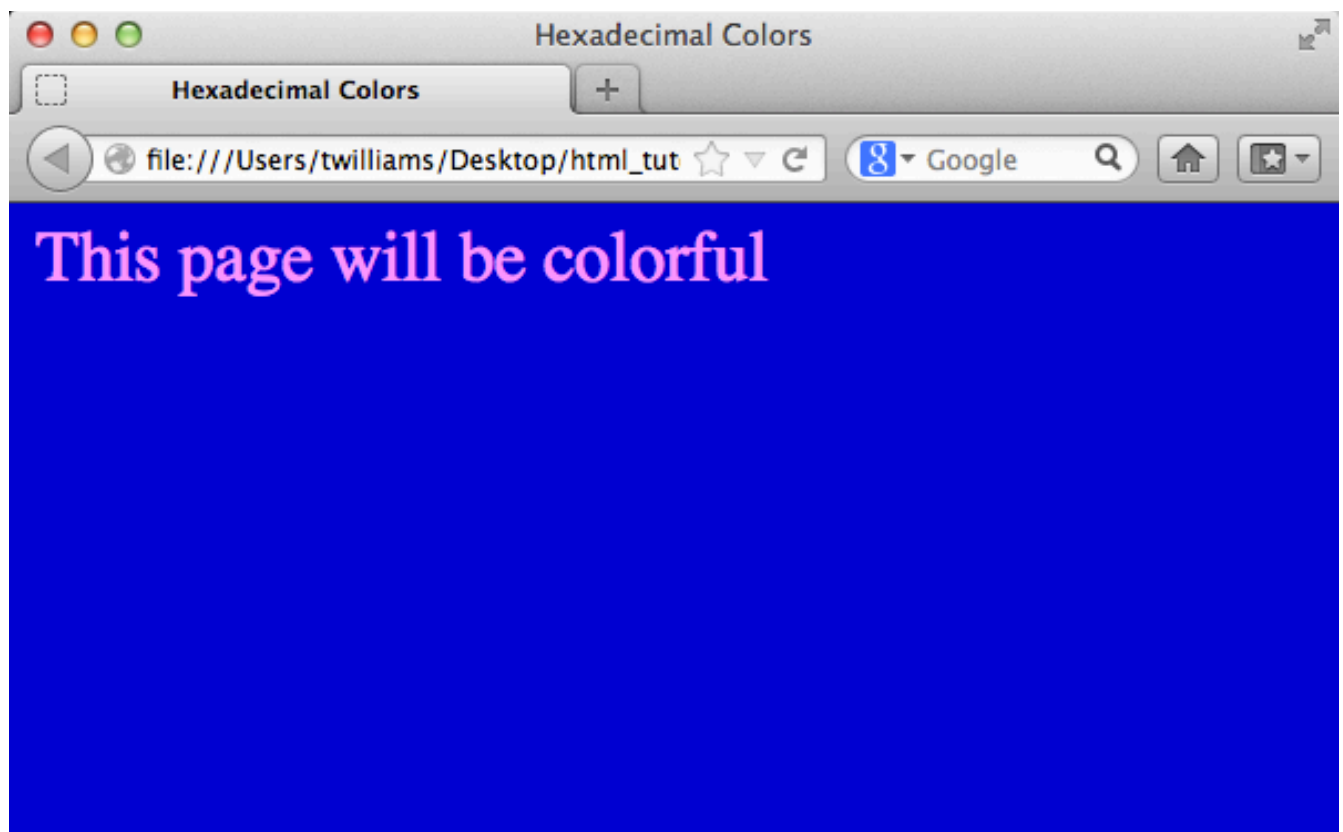
We will use our code for the `colors.html` file as we continue to learn about adding colors to our web document. Make the following change to `colors.html`, save and refresh.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Hexadecimal Colors</title>
</head>
<body bgcolor="#0000cc" text="#ffffff">

<font size="6" color="#ff99ff">This page will be colorful</font>

</body>
</html>
```

Your text should now appear pink, as shown below:



Try adding the following code:

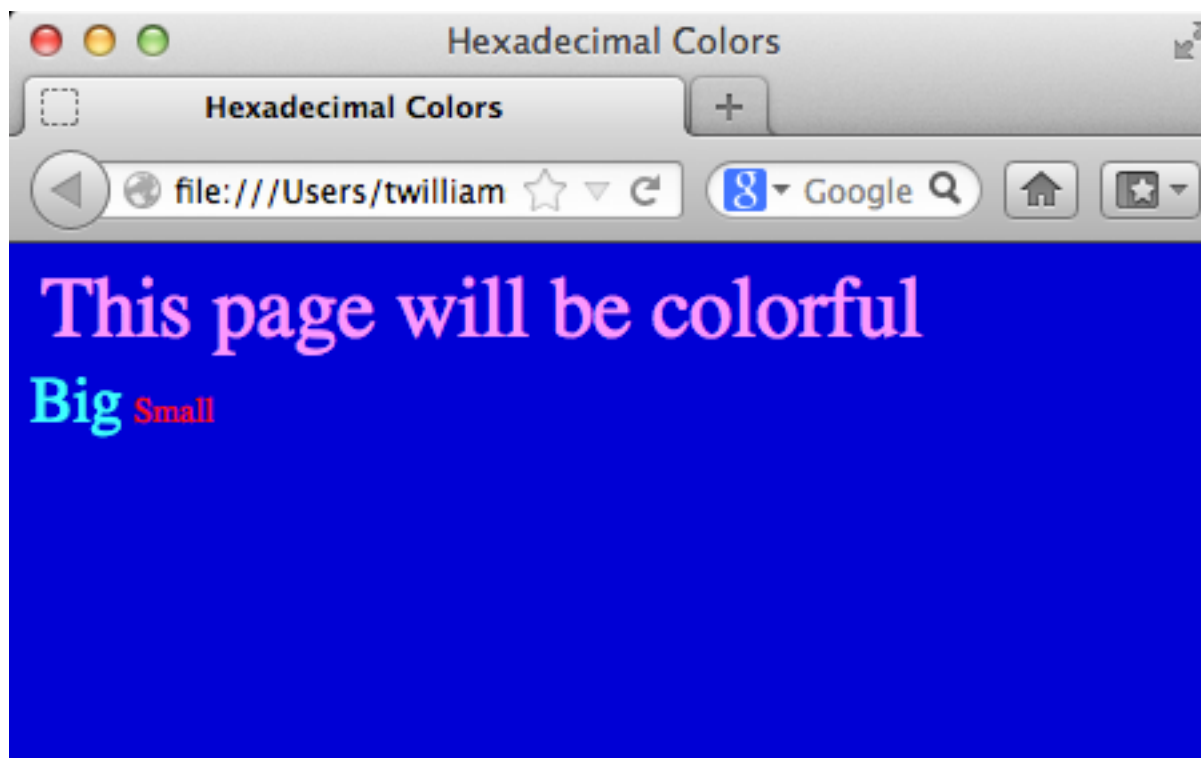
```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Hexadecimal Colors</title>
</head>
<body bgcolor="#0000cc" text="#ffffff">

<font size="6" color="#ff99ff">This page will be colorful</font>
<br>

<font size="5" color="#00ffff">Big</font>
<font color="#ff0000" size="2">Small</font>

</body>
</html>
```

(\*\* Notice: there are no `<br>` commands around the added text, so the words appear on the same line. Also, notice how the order of the attributes "size" and "color" does NOT matter. As long as "font" comes first, the attributes can be in any order.)



## Formatting Font Faces

---

What about the actual fonts themselves? So far, we've used the default font face set in our browser, with no control over which one is used. The following example shows how to specify a font face. Keep in mind, though, there is no guarantee that the person viewing your page will have the same fonts. In that case, their default is used. Create the following file and save in your folder as `font_faces.html`. You may need to check your computer for fonts if your system does not have the two I've used. If a font name is expressed in multiple words, type it as displayed on your computer, including any spaces.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>The font tag</title>
</head>
<body>

<font size="6" face="impact">I used the Impact font here</font>
<br>
<font size="6" face="hobo std">and the Hobo STD font here.</font>

</body>
</html>
```

My file looks like this:





## Creating Hyperlinks (Links)

---

**<a href = ....>** *Instructs the browser to create a link, either with text or an image. The "a" stands for anchor, and the attribute "href" stands for hyper-reference. Here is a link to Google that displays the word "Search" as the linking text.*

```
<a href="http://www.google.com">Search</a>
```

Notice that the closing tag is simply `</a>` because "href" is an attribute, or something "extra" for the tag. We can't create a link without the "href" attribute.

Link colors: The default settings for link colors are as follows:

Link .....link .....blue (site has not been visited yet)

Visited link....vlink .....purple (you've been to the site recently)

Active link.....alink .....red (some browsers change to this color as the link is clicked)

After learning how to create links, we will learn how to control their colors. By default, links are underlined.

Open your template code in Notepad or TextWrangler and save as `links.html`. Try adding a few links to your file as shown. Choose any links you like.

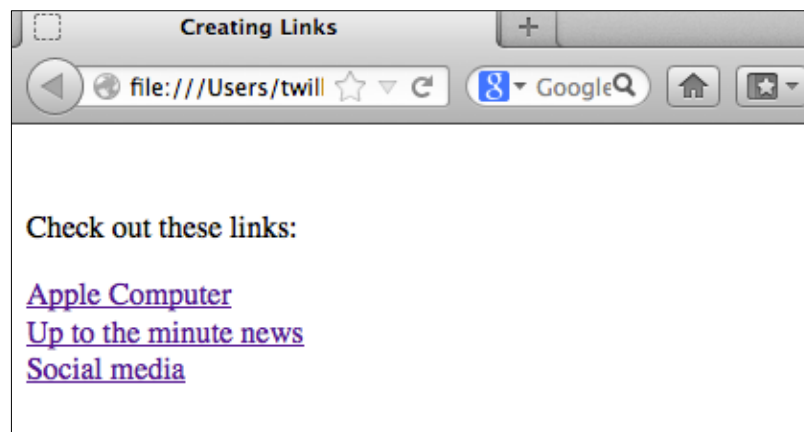
```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Creating Links</title>
</head>

<body>

<p>Check out these links:</p>

<a href="http://www.apple.com">Apple Computer</a> <br>
<a href="http://www.cnn.com">Up to the minute news</a> <br>
<a href="http://www.facebook.com">Social media</a> <br>

</body>
</html>
```



\*\* Reminder: browsers ignore extra spaces, tabs, and returns in our code. I have added spaces to make it more visually pleasing to me. Also, the commands can appear on separate lines in the code. I could have written the above HTML file as shown below and had the exact same results: (do NOT make these changes as it is not necessary).

Here is my "ugly" code - same result as the previous file, just much harder to read.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>
        Creating Links
    </title>
</head>

<body>

<p>
Check out these links:
</p>

<a href="http://www.apple.com">
Apple Computer
</a>
<br>
<a href="http://www.cnn.com">
Up to the minute news
</a> <br>
<a href="http://www.facebook.com">Social media
</a> <br>

</body>
</html>
```

\*\* This is not nearly as easy to read, for me. You will use returns, tabs, and spaces as YOU like when writing your code.

Changing link colors with <body> attributes: Make the following changes to your `links.html` file and save as `links_colors.html`. Notice that I've changed two links to sites I have NOT visited lately so that I can see if the link color works.

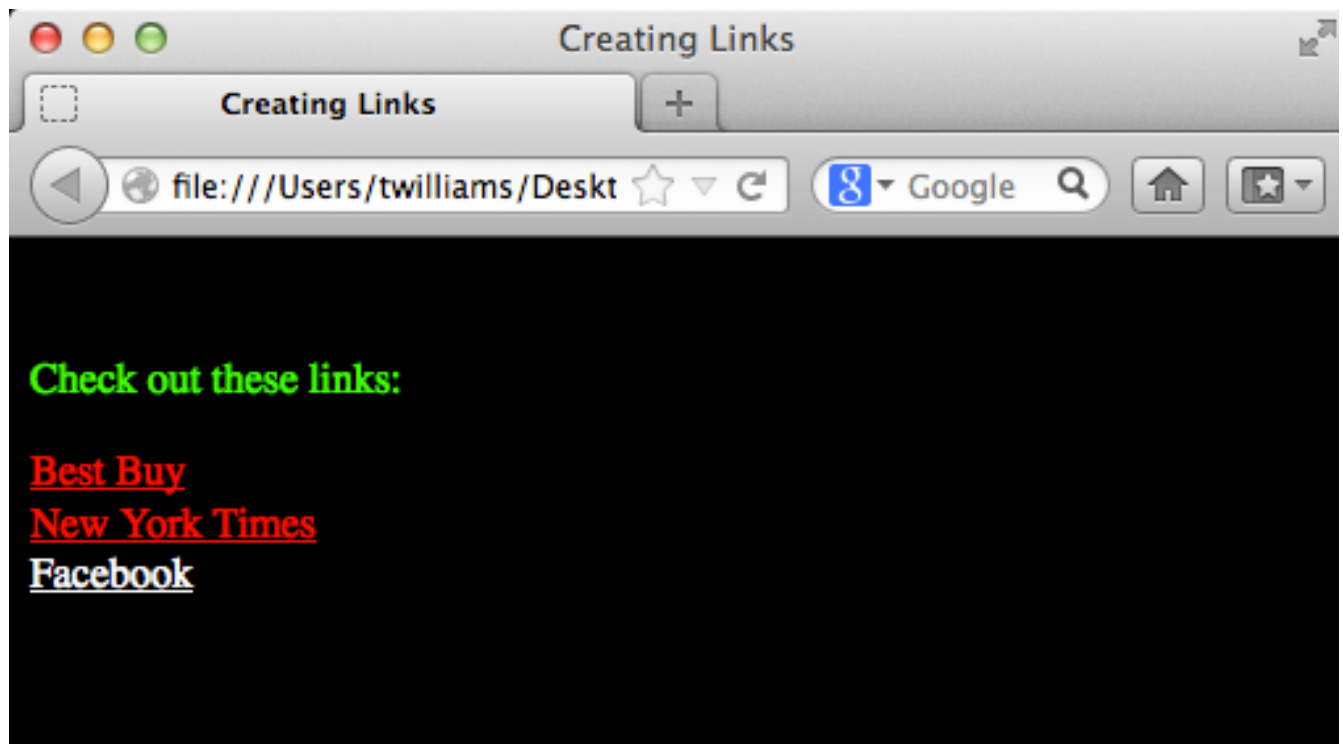
```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Creating Links</title>
</head>

<body bgcolor="#000000" link="#ff0000" vlink="#ffffff" alink="#ffffff"
      text="#00ff00">

<p>Check out these links:</p>

<a href="http://www.bestbuy.com">Best Buy</a> <br>
<a href="http://www.nytimes.com">New York Times</a> <br>
<a href="http://www.facebook.com">Facebook</a> <br>

</body>
</html>
```



I have NOT visited Best Buy or the New York Times sites, so they appear with the LINK color. If I were to click on either of these and then hit the BACK button, they would appear in the VLINK color. I've been to Facebook recently, so that is displayed in the VLINK color.

## Displaying Lists and Horizontal Rules

---

Organizing data into lists makes it easier for people to find the important information on your web page. There are three types of lists we can create in HTML.

### Three Types:

1. Unordered     <ul>, <li> (*bullets*)
2. Ordered       <ol>, <li> (*numbers or letters*)
3. Definition    <dl>, <dt>, <dd> (*words*)

**<ul>....</ul> Unordered List:** Open your template code and save as **lists.html**. We will create lots of different list styles. Each **list item** is enclosed within the **<li>** and **</li>** tags.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Lists</title>
</head>

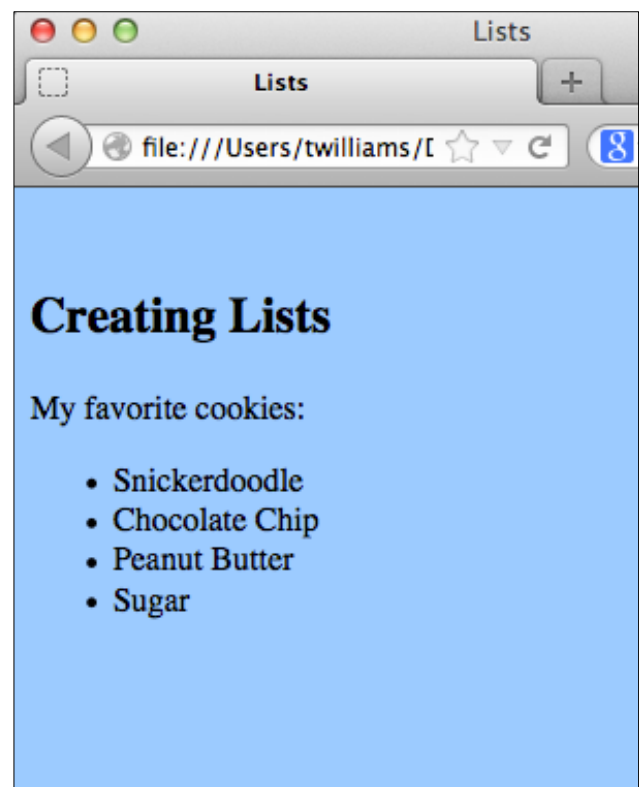
<body bgcolor="#99ccff">

<h2>Creating Lists</h2>

My favorite cookies:
<ul>
  <li>Snickerdoodle</li>
  <li>Chocolate Chip</li>
  <li>Peanut Butter</li>
  <li>Sugar</li>
</ul>

</body>
</html>
```

\* Save and check in browser.



**<hr> Horizontal Rule:** Creates a line across the browser window. We will insert horizontal rules to separate our lists. Later, we will learn how to change the way the line looks.

Attribute for the <ul> tag:

- type="disc" (solid bullet, default)
- type="circle" (open circle)
- type="square" (small square)

Add the <hr> commands and COPY and PASTE the existing list code. Add the "type" attribute.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Lists</title>
</head>

<body bgcolor="#99ccff">

<h2>Creating Lists</h2>

My favorite cookies:
<ul>
  <li>Snickerdoodle</li>
  <li>Chocolate Chip</li>
  <li>Peanut Butter</li>
  <li>Sugar</li>
</ul>

<hr>

My favorite cookies:
<ul type="circle">
  <li>Snickerdoodle</li>
  <li>Chocolate Chip</li>
  <li>Peanut Butter</li>
  <li>Sugar</li>
</ul>

<hr>

</body>
</html>
```



\* SAVE and REFRESH

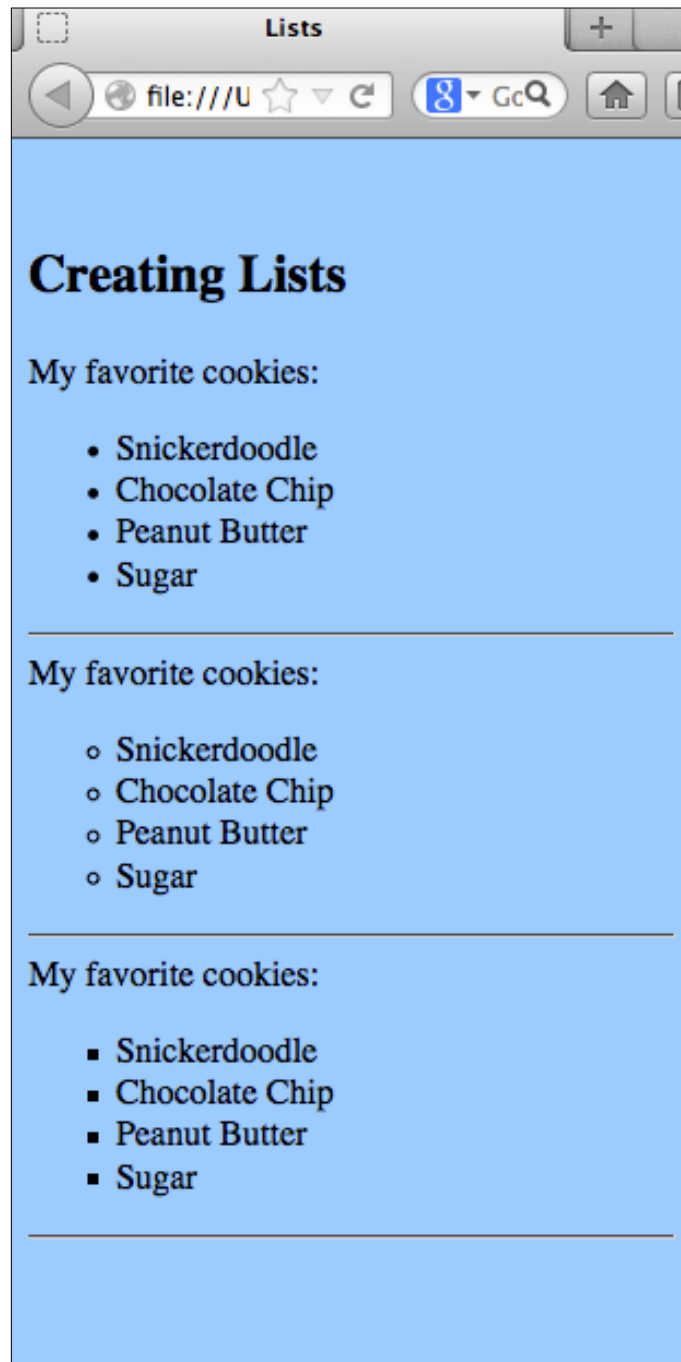
Now copy the code again and try the square setting:

My favorite cookies:

```
<ul type="square">  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ul>
```

```
<hr>
```

\* SAVE and REFRESH



**<ol>...</ol> Ordered List:** These lists can be ordered using numbers, letters, or Roman numerals. Each **list item** is enclosed within the **<li>** and **</li>** tags.

We will keep adding to our existing lists.html code.

**Attributes for <ol>:**

- start="number" (specifies the starting point in the list)
- type="1" (default, numbers)
- type="A" (upper case letters)
- type="a" (lower case letters)
- type="I" (upper case Roman numerals)
- type="i" (lower case Roman numerals)

Try adding the following code:

```
My favorite cookies:  
<ol>  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

```
My favorite cookies:  
<ol type="A">  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

The image shows two browser window screenshots. The top screenshot displays the rendered HTML code from the first code block, showing a numbered list of four cookies: 1. Snickerdoodle, 2. Chocolate Chip, 3. Peanut Butter, and 4. Sugar. The bottom screenshot displays the rendered HTML code from the second code block, showing a lettered list of four cookies: A. Snickerdoodle, B. Chocolate Chip, C. Peanut Butter, and D. Sugar.

\* SAVE and REFRESH

Now try adding the following code:

```
My favorite cookies:  
<ol start="43">  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

```
My favorite cookies:  
<ol type="a" start="7">  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

My favorite cookies:

43. Snickerdoodle
44. Chocolate Chip
45. Peanut Butter
46. Sugar

My favorite cookies:

- g. Snickerdoodle
- h. Chocolate Chip
- i. Peanut Butter
- j. Sugar

\* Ordered lists automatically line up the letters or numbers with a decimal tab (notice how the decimal points/periods all line up?). If your font is too big, or your numbers too large, it can mess up the layout.

\* When using letters, the start attribute must be a number associated with the letter you want to start on - ask yourself, "Where does this number fall in the alphabet?" For example, "g" is the 7th letter in the alphabet, so I used start="7" in my code.



Now try adding the following code for Roman numerals and negative numbers:

```
My favorite cookies:  
<ol type="I" start="5" >  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

```
My favorite cookies:  
<ol start="-9">  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

```
My favorite cookies:  
<ol type="A" start="45" >  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

```
My favorite cookies:  
<ol type="i" start="10">  
  <li>Snickerdoodle</li>  
  <li>Chocolate Chip</li>  
  <li>Peanut Butter</li>  
  <li>Sugar</li>  
</ol>
```

```
<hr>
```

My favorite cookies:

- V. Snickerdoodle
- VI. Chocolate Chip
- VII. Peanut Butter
- VIII. Sugar

My favorite cookies:

- 9. Snickerdoodle
- 8. Chocolate Chip
- 7. Peanut Butter
- 6. Sugar

With only 26 letters in the alphabet, what happens when you start on a number greater than 26? You should see the letters double up, AA, AB, AC,...AZ, BA, BB, BC, ....., BZ, etc.

My favorite cookies:

- AS. Snickerdoodle
- AT. Chocolate Chip
- AU. Peanut Butter
- AV. Sugar

My favorite cookies:

- x. Snickerdoodle
- xi. Chocolate Chip
- xii. Peanut Butter
- xiii. Sugar

**<dl>...</dl> Definition List:** This type of list is good for typing glossaries or notes. The terms are enclosed in <dt>...</dt> tags and the definitions are enclosed in <dd>...</dd> tags, which indent about half an inch to the right.

We can continue adding to our lists.html code. Simply enter this text below the last <hr>.

Computer Related Terms:

```
<dl>
  <dt>Internet</dt>
  <dd>A global computer network.</dd>

  <dt>Network</dt>
  <dd>Computers connected together to shares files
      and devices.</dd>

  <dt>WWW</dt>
  <dd>World Wide Web, part of the Internet.</dd>
</dl>
```

---

### Computer Related Terms:

#### Internet

A global computer network.

#### Network

Computers connected together to shares files and devices.

#### WWW

World Wide Web, part of the Internet.

---

## Formatting Horizontal Rules

**<hr> Horizontal Rule:** the `<hr>` tag displays a thin, shaded line across the browser window. We used it to separate our lists in the previous file. We can modify how it appears by using the following attributes:

- `size="number"` (the height in pixels)
- `color="#000000"` (displays the line with the assigned hexadecimal color)
- `color="colorname"` (we haven't used color names yet, but we'll try it here)
- `width="number"` (width in fixed pixels)
- `width="number%"` (width as percentage of browser window width)
- `noshade="noshade"` (turns off shading, sometimes appears as single word **noshade**)
- `align="left, center, or right"` (by default, most browsers align horizontal rules in the center, but we can modify the alignment)

\* Note: the horizontal rules will appear rounded on the ends in some browsers, and square in others.

Open the template code and save as `horizontal_rules.html` in your folder. As you create this file, save and refresh after each `<hr>` to see what it does. See the final screenshot on the next page.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Horizontal Rules</title>
</head>

<body bgcolor="#ffffcc">

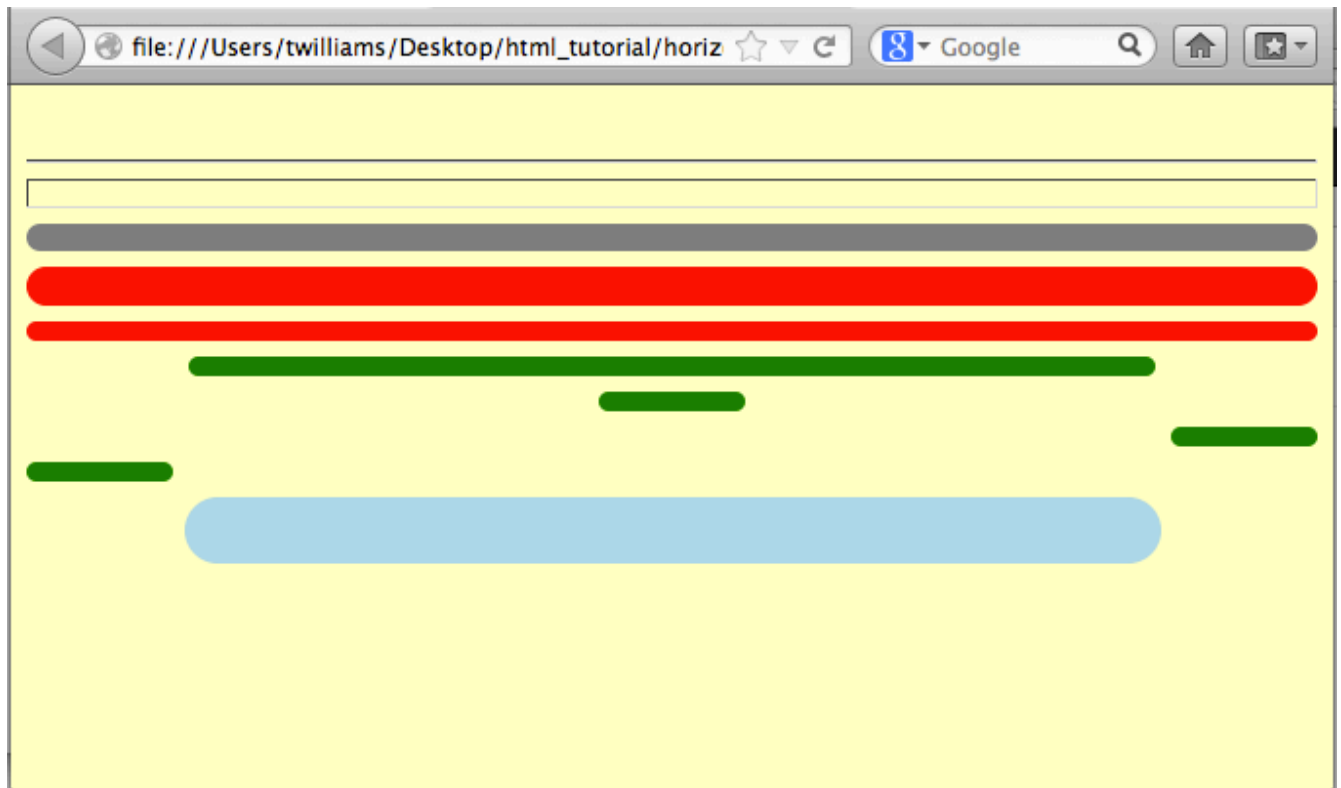
  <hr>
  <hr size="15">
  <hr size="15" noshade="noshade">

  <hr size="20" color="#ff0000">
  <hr size="10" color="red"> (notice the use of a color name and lack of number symbol)

  <hr size="10" color="green" width="75%">
  <hr size="10" color="green" width="75%">
  <hr size="10" color="green" width="75" align="right">
  <hr size="10" color="green" width="75" align="left">

  <hr width="500" size="35" color="lightblue">

</body>
</html>
```



## Using Color Names

---

Whenever you use the bgcolor attribute, you can use a color name instead of hexadecimal code. The problem with this is, let's say you specify bgcolor="blue", you are at the mercy of the browser to determine which shade of blue is displayed. It may not be what you had in mind. There are many websites dedicated to listing all the color names, from classic red, blue, green, to more obscure, like peachpuff, mauve, seagreen, and powderblue. Take a moment to search google for a list. I prefer hexadecimal code because you get what you expect. Many of the colors have "unusual" hexadecimal codes listed beside them that are not in my web safe list. They might work in most browsers, but I won't guarantee it!

This is a site that I often reference:

[http://www.w3schools.com/html/html\\_colornames.asp](http://www.w3schools.com/html/html_colornames.asp)

Well, that's if for this tutorial. We covered a LOT of material! Look for my next tutorial (available in April, 2013) at [www.teacherspayteachers.com](http://www.teacherspayteachers.com). This will focus on inserting images and backgrounds.

## Up Next

---

*"Get Your Geek On with HTML"* Book 2 - Images and Backgrounds




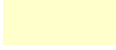




































































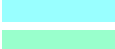



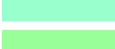



















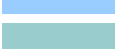














*"Get Your Geek On with HTML"* Book 3 - Tables

































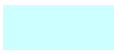



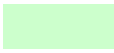



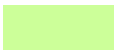

















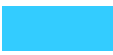












































## Questions?

---

If you have questions about the content of this tutorial, you can reach me at: [twilliams38@gmail.com](mailto:twilliams38@gmail.com). Include "Get your geek on" in the subject line so your message is not overlooked. Thank you!

# HTML Hexadecimal Color Chart

#FFFFFF		#CC66FF		#66FFFF		#3366FF	
#FFFFCC		#CC66CC		#66FFCC		#3366CC	
#FFFF99		#CC6699		#66FF99		#336699	
#FFFF66		#CC6666		#66FF66		#336666	
#FFFF33		#CC6633		#66FF33		#336633	
#FFFF00		#CC6600		#66FF00		#336600	
#FFCCFF		#CC33FF		#66CCFF		#3333FF	
#FFCCCC		#CC33CC		#66CCCC		#3333CC	
#FFCC99		#CC3399		#66CC99		#333399	
#FFCC66		#CC3366		#66CC66		#333366	
#FFCC33		#CC3333		#66CC33		#333333	
#FFCC00		#CC3300		#66CC00		#333300	
#FF99FF		#CC00FF		#6699FF		#3300FF	
#FF99CC		#CC00CC		#6699CC		#3300CC	
#FF9999		#CC0099		#669999		#330099	
#FF9966		#CC0066		#669966		#330066	
#FF9933		#CC0033		#669933		#330033	
#FF9900		#CC0000		#669900		#330000	
#FF66FF		#99FFFF		#6666FF		#00FFFF	
#FF66CC		#99FFCC		#6666CC		#00FFCC	
#FF6699		#99FF99		#666699		#00FF99	
#FF6666		#99FF66		#666666		#00FF66	
#FF6633		#99FF33		#666633		#00FF33	
#FF6600		#99FF00		#666600		#00FF00	
#FF33FF		#99CCFF		#6633FF		#00CCFF	
#FF33CC		#99CCCC		#6633CC		#00CCCC	
#FF3399		#99CC99		#663399		#00CC99	
#FF3366		#99CC66		#663366		#00CC66	

#FF3333		#99CC33		#663333		#00CC33	
#FF3300		#99CC00		#663300		#00CC00	
#FF00FF		#9999FF		#6600FF		#0099FF	
#FF00CC		#9999CC		#6600CC		#0099CC	
#FF0099		#999999		#660099		#009999	
#FF0066		#999966		#660066		#009966	
#FF0033		#999933		#660033		#009933	
#FF0000		#999900		#660000		#009900	
#CCFFFF		#9966FF		#33FFFF		#0066FF	
#CCFFCC		#9966CC		#33FFCC		#0066CC	
#CCFF99		#996699		#33FF99		#006699	
#CCFF66		#996666		#33FF66		#006666	
#CCFF33		#996633		#33FF33		#006633	
#CCFF00		#996600		#33FF00		#006600	
#CCCCFF		#9933FF		#33CCFF		#0033FF	
#CCCCCC		#9933CC		#33CCCC		#0033CC	
#CCCC99		#993399		#33CC99		#003399	
#CCCC66		#993366		#33CC66		#003366	
#CCCC33		#993333		#33CC33		#003333	
#CCCC00		#993300		#33CC00		#003300	
#CC99FF		#9900FF		#3399FF		#0000FF	
#CC99CC		#9900CC		#3399CC		#0000CC	
#CC9999		#990099		#339999		#000099	
#CC9966		#990066		#339966		#000066	
#CC9933		#990033		#339933		#000033	
#CC9900		#990000		#339900		#000000	