

CSS Tutorial #1

In this tutorial, we are going to

- style the web page we made last class in a similar manner to the Resource Page on Mr. Pelletier's site,
- learn how CSS works, and learn many of the properties you can use to change the appearance of a site.

1. Link your HTML Document to a CSS File

Step 1: Open up your web site project folder (the folder with your HTML file). Create a new file in Notepad++, and save it as "theme.css" in this same folder as your HTML file.

Step 2: Your project folder should now contain an HTML file, a CSS file, and whatever pictures you included in your web page.

Next, open your HTML file in Notepad++ for editing. Make sure that in between the <head> </head> tags, you have included a link tag to connect your HTML file to a CSS file, like so:

```
<head>
  <title> Page Title </title>
  <link rel="stylesheet" type="text/css" href="theme.css" />
</head>
```

2. Begin Styling

Your CSS file will contain a set of rules that your tags will follow, and these rules will determine how your web page looks. You can customize almost any aspect of your web page using CSS.

A typical CSS rule look like the following:

```
selector {
  property: value;
}
```

The selector is the name of an HTML tag, and the property: value; part makes up the rules that will style that tag. You can have many rules for each tag, just make sure each is enclosed in the { } braces.

Step 1 - Colors: Let's start by changing the colors on our web page. First, let's change the background color and text color. Open your CSS file and make your first style rule by typing in the following:

```
body {
  background-color: black;
  color: white;
}
```

The background-color property changes – you guessed it – the color of the background. The color property changes the color of the text. Because you put this property with the body tag, all the text in the page is now white. You can use pretty much any English word that means a color as a value (green, lilac, gold, etc.), but if you want to pick the shade of the color exactly, you can type in it's RGB code. Use html-color-codes.info to select an

exact color. Instead of white, I'm going to use #6E9EFE for my text, so my code would become:

```
body {  
  background-color: black;  
  color: #6E9EFE;  
}
```

Step 2 - Fonts: Time to change the font. Before we do that, you should know that not all fonts are safe to use on the web. If you use some fancy custom font that you have on your computer, it might not show up the same way on somebody else's computer because they don't have that font. Check out this list of [web safe fonts](#) and choose one. I'm going to choose Tahoma. To change all the text to that font, I'll do the following:

```
body {  
  background-color: black;  
  color: #6E9EFE;  
  font-family: tahoma, arial;  
}
```

What's with the Arial afterward? Well, just in case the computer doesn't have tahoma as a font, then I have specified the next best font to use.

Step 3 – Headers: So far our changes have been affecting all of the text. What if we just wanted to play around with our <h1> tags? Let's make a new rule to center the text!

```
body {  
  background-color: black;  
  color: #6E9EFE;  
  font-family: tahoma, arial;  
}  
  
h1 {  
  text-align: center;  
  font-size: 200%  
}
```

This code new style rule will only affect the code in the <h1> tags. Your header should now be centered and double its original size. There are other values for text-align, by the way. You could use **text-align:left;** or **text-align:right;** or **text-align: justify;**. These are more useful for paragraphs, not headers.

Let's make our first style rule and give our page a background. First, find a nice picture to be your background. Go to [interfacelift.com](#) and pick a few nice pictures, and download them at 1280 x 800 resolution. Make sure to save them in the same folder as your CSS and HTML file.

Step 4 – Image Backgrounds: Let's make give our page a picture for a background. First, find a nice picture to be your background. Go to [interfacelift.com](#) and pick a few nice pictures, and download them at 1280 x 800 resolution. Make sure to save them in the same folder as your CSS and HTML file.

Next, add the following rule, replacing “filename.jpg” with the name of your picture:

```
body {
  background-color: black;
  background-image: url(filename.jpg);
  color: #6E9EFE;
  font-family: tahoma, arial;
}
```

Open your HTML file in a web browser to check if your background has changed. Try out a few until you find a background you like. Your text may not show up very well on the background, but we will fix that shortly.

Optionally, you can prevent your picture background from scrolling with the rest of the page. Try adding this rule to your CSS file:

```
body {
  background-attachment: fixed;
  background-color: black;
  background-image: url(filename.jpg);
  color: #6E9EFE;
  font-family: tahoma, arial;
}
```

Refresh your web page just to make sure it worked. When you scroll, the background should not move.

Step 5 – Boxes: Let's put that text in a box so that it is nice and readable. First, we're going to add a new tag to our HTML file. Open your index.html file, and add in a <div> tag, right below the body tag and a </div> tag right before the </body> tag. It should look like this:

```
<html>

  <head>
  ...
</head>

  <body>
    <div>
      ...
    </div>
  </body>

</html>
```

What is this div tag? Div stands for division. What you are doing is creating a division of the page that will be styled separately from everything else. That way we can give a new style (including backgrounds, colors, and fonts) to just one part of the page, while leaving everything else unchanged.

To style your new <div>, we're going to make a new background. Go to <http://www.stripegenerator.com/> and create a nice looking striped background. Try to choose colors that go with your background – you may want to use <http://www.colr.org/> to help you pick the exact colors from your image.

Once you've made your striped background, save it to your project folder as stripe.png.

Next, give your div some style!

```
body {
  background-color: black;
  color: #6E9EFE;
  font-family: tahoma, arial;
}

div {
  background-image: url(stripe.png);
  border-color: white;
  border-style: solid;
  border-width: thick;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  margin-bottom: 40px;
  opacity: 0.7;
  width: 500px;
}

h1 {
  text-align: center;
  text-size: 200%
}
```

Woah, what's happening? Here's a break down:

background-image made your div have a new background.

border-color set the color of the border around the div.

border-style set the type of border around the div.

margin-left and **margin-right** are set to auto, which centers the div

margin-top and **margin-bottom** are how much space are above and below the div.

opacity makes the div see-thru, currently set to 70%

width sets the width of the div

Refresh your page in the browser to see what's happened!

Step 5 – Padding: Our div is looking pretty good, but it would be nice if there was some padding between the border and the text! Can you guess what that property is called? Add the following:

```
div {
  background-image: url(stripe.png);
  border-color: white;
  border-style: solid;
  border-width: thick;
  margin-left: auto;
  margin-right: auto;
  margin-top: 40px;
  margin-bottom: 40px;
  padding: 10px;
  opacity: 0.7;
  width: 500px;
}
```

Step 6 – Links: Styling your links requires 3 sets of rules: 1 set for when your links are just sitting there (called **link**), another set for when your links have been visited (called **visited**), and another for when the mouse is touching them (called **hover**).

Try adding the following rules, and modify the colors to suit your web page.

```
a:link{  
    color: #8FBBFE;  
    font-weight: bold;  
    text-decoration: none;  
}
```

```
a:visited{  
    color: #8FBBFE;  
    font-weight: bold;  
    text-decoration: none;  
}
```

```
a:hover{  
    background-color: #FEC300;  
    color: #001D49;  
    font-weight: bold;  
    text-decoration: none;  
}
```