CSS worksheet

JMC 105 | Drake University

1. Download the css-site.zip file from the class blog and expand the files. You'll notice that you have an images folder with three images inside and an index.html file with all of the content already there.

2. In your source code editor (TextWrangler or Notepad++), create a new file and save it as **styles.css**. This will be the stylesheet for our website. This is an **external stylesheet**, meaning it can apply to multiple pages in our website. If we had a website with 1,000 pages, and one day we woke up and decided all of the headlines in our website should be orange, we could make that change one time in the stylesheet, and it would apply to our entire site (translation: external stylesheets rule!).

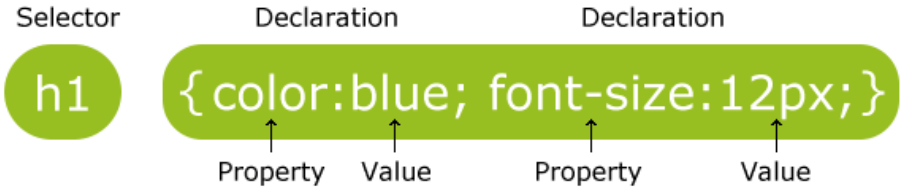
3. Open the index.html file both in your source editor and in your web browser (Firefox). It shouldn't look like much in your browser, but that will change soon.

I've already set up the index.html file to link to the stylesheet. Here's the code that does that:

```
<link rel="stylesheet" type="text/css" href="styles.css" />
```

4. Most of the work we will do today will be in our styles.css file. But we will have to add some code to our index.html file as well. First, a reminder of what our CSS code looks like.

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style - in this case h1.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

5. For the sake of keeping our code looking nice and neat, we generally write it a little different than what you see above. To get things started, let's apply some styles to our body tag. We do that by writing the following in our styles.css file:

```
body {
    font-family:Georgia;
    background-color:#CCFFCC;
    }
```

If you refresh your index.html page in Firefox, you will notice that all type on the page now is in the font Georgia and that the page has a nice mint green background.

I used tabs to indent the declarations, and I put one on each line. The tabs don't do anything. They just make it easier to see what I am working on. Also, notice the semi-colon at the end of each line. That's important. If your styles are not working, it usually means you are missing a semi-colon somewhere in your styles.

6. Now let's switch from a background color to a background image. You can leave the background color in, but add the line below that links to a background image:

```
body {
    font-family:Georgia;
    background-color:#CCFFCC;
    background-image:url('images/stripes.png');
    }
```

If you open that stripes.png file, you'll notice that it's just a 10-pixel wide file, but it repeats over and over to create a background image.

7. Next we will style our headlines. We can style both our H1s and our H2s at the same time by separating them by a comma in the selector. Press return a couple times and put this code below the body styles:

```
h1,h2 {
    color:#663300;
    font-family:Arial;
    line-height:70%;
    }
```

8. If we want to be a little more specific with h2 styles, we still can do that. Add the following code:

```
h2 {
    font-size:18px;
    line-height:100%;
    }
```

9. Next style the paragraphs. Here we are using a font stack. This is telling the browser to use helvetica if possible. It the computer does not have helvetica, then it is saying to use arial. And if the computer does not have arial, then it is saying to just use the default sans-serif font.

```
p {
    font-family:Helvetica,Arial,sans-serif;
    font-size:14px;
    }
```

10. Next we style our unordered list, and most importantly get rid of the default indent that all lists have.

```
ul {
    font-family:Helvetica, Arial;
    font-size:12px;
    margin-left: 1em;
    padding-left: 0em;
    line-height:16px;
    }
```

This is a good time to look at what is known as the CSS box model (right). If you think of all CSS elements as rectangles, then each of those rectangles has content. Outside the content is what is known as padding. Then comes the border. Then comes the margin. We will get into this more later in this worksheet.

11. Next style the image. In this case, we're just giving it a border. You can define all three elements of the border (width, style and color) all in one line like so:

```
img {
    border: 2px solid #663300;
    }
```

Now change "border" to "border-left" to see that you can set all four borders separately. The others are border-top, border-right and border-bottom.

12. Next we style our links, which use the "a" tag. By default, links are blue and underlined. And once you have visited the link, it becomes purple and underlined.

But links also have two more properties to set. They can look one way while the mouse is hovering over them, and another way while you are actually clicking on them. The code below shows how to set all four states (I'll present it different to save space). The words to the right of each line are notes. Anything between /* and */ will not show up as actual CSS code. This is how we leave notes for someone else who might be looking at our code.

```
a:link {color:#2e5aba; text-decoration:none;}      /* unvisited link */
a:visited {color:#2e5aba; text-decoration:none;}  /* visited link */
a:hover {color:#000099; text-decoration:underline;}  /* mouse over link */
a:active {color:#000000; text-decoration:none;}  /* selected link */
```

Refresh your browser to see how the link in our first paragraph looks when you hover and while you are clicking on it.

13. Sometimes you want to create a style on your page that isn't associated with an already-existing tag. For example, we might want to create a pullquote in our text. To do this, we create a **style class**. Style classes get a period before their name, and you get to make up the name. In your styles.css file, create the following style class:

```
.pullquote {
    font-weight:bold;
    color:#663300;
    padding-left:20px;
    border-left:10px solid #663300;
    }
```

To make that class actually do something, we need to add it to our index.html page. In the second paragraph, add class="pullquote" to the p tag. This style will end wherever the p tag ends - in this case at the end of the sentence.

```
<p class="pullquote">Almost 95 percent (94.9) of 2006 Drake journalism graduates
reported being employed in the field or in graduate school, according to a
recent survey by the university. Of these, 89.5 percent reported having had an
internship while in school.</p>
```

14. Now we will style the links in our top navigation. These are already set up on our index.html file (using **a class="nav"**). We just need to give them the proper style. Notice how the selectors below compare to the ones in step 12.

```
a.nav:link {color:#663300; text-decoration:none;}      /* unvisited link */
a.nav:visited {color:#663300; text-decoration:none;}  /* visited link */
a.nav:hover {color:#663300; text-decoration:underline;}  /* mouse over link */
a.nav:active {color:#663300; text-decoration:none;}  /* selected link */
```

15. You're on your own for this next one. I want you to create a class called **radio** and apply it to the first paragraph under the 94.1 The Dog headline. Your text should be white, your background color should be #663300 and you should have padding of 10px on all four sides.

16. You can see that I have already applied a class of **navigation** to our top navigation near the top of your index.html file. Now we just need to write the CSS to style that. The new thing here is the display:inline, which will take a list and display it as a line of text. This is commonly used to display a list as top navigation on a page. The li before the period indicates this style class can only apply to li tags.

```
li.navigation {
    display:inline;
    font-family: Arial;
    font-size: 16px;
    }
```

17. So far we have only worked with external styles. Let's add an inline style just to see how those work. We'll make the words **94.1 The Dog** yellow and bold in the paragraph you styled in step 15. We do that by using a span tag. The span tag allows you to apply styles in the middle of a paragraph:

```
<p class="radio">The School of Journalism and Mass Communication is also home to
<span style="color:yellow;font-weight:bold">94.1 The Dog</span>, which operates
under the call letters KDRA-LP FM. The station launched in August 2006 after
having existed as an internet station, KDCS Bulldog Radio. 94.1 The Dog is
broadcast at 80 watts from a tower atop Meredith Hall, the home of Drake's
SJMC.</p>
```

---

18. For the second part of this lesson, we will be using CSS to design the layout of our actual page. We will center our content, give it a fixed width and create two columns.

First of all, we will center our content. At the bottom of the styles.css file, type the following (notice that I put a note at the top to help future me know what I was doing with this code:

```
/* =====DIVIDES THE PAGE INTO TWO COLUMNS====== */

div#content {
    width: 600px;
    height: 1200px;
    margin: 0 auto;
    padding: 20px;
    background-color: #FFFFFF;
    border: 20px solid #663300;
    }
```

Before I explain to you exactly what this code is doing, let's apply it to our index.html page. This is a **style ID**, which can only be used once per page (as opposed to classes which can be used multiple times). We add them to our page using the div tag. This first one is going to wrap all of our content. Since all of our content goes between the body tags, we open our div right after we open the body tag. Like this:

```
<body>

<div id="content">
```

And we close the div right before we close the body tag, like this:

```
</div>

</body>
```

So what is this code doing? If you save your styles.css and your index.html, then refresh your page in Firefox, you will see.

The width and height declarations are simply setting a width and height for the box. The margin: 0 auto is setting a margin of 0 at the top and bottom and centering (auto) the content on the left and right.

The padding of 20px goes inside the border (remember from our box model that padding goes inside the border and margin outside).

The background color of our box is white. And finally, we are giving it a 20px solid border using the color #663300.

19. Now we want to divide our content into two columns. To do that, create the following on your styles.css file:

```
div#left {
    float: left;
    width: 370px;
    padding-left: 20px;
    }

div#rail {
    float: right;
    background-color: #CCCCCC;
    width: 175px;
    padding-left:15px;
    margin-top:15px;
    border: 2px dashed #663300;
    }
```

The div#left will wrap the majority of our content, the sections on Meredith Hall and the section on 94.1 The Dog. We open that div below our image and above our h1 on Meredith Hall:

```
<img src="images/meredith.jpg" width="600" height="350">

<div id="left">

<h1>Meredith Hall</h1>
```

We close it before our h2 on Meredith Hall hours, which also happens to be where we open our div#rail. We close our div#rail after the Meredith Hall hours list:

```
campus even when not broadcasting on the frequency.</p>


</div>

<div id="rail"><h2>Meredith Hall hours</h2>
<ul>
<li>Monday: 7 a.m. - 10 p.m.</li>
<li>Tuesday: 7 a.m. - 10 p.m.</li>
<li>Wednesday: 7 a.m. - 10 p.m.</li>
<li>Thursday: 7 a.m. - 10 p.m.</li>
<li>Friday: 7 a.m. - 10 p.m.</li>
<li>Saturday: CLOSED</li>
<li>Sunday: 2 p.m. - 10 p.m.</li>

</div>
```

The new thing we introduced in step 19 are the **float:left** and **float:right**, which basically just send our content as far as it can go to the left and right. Perfect when we want two columns in our design.

20. We still have one line of type that will be the footer at the bottom of our page. Add the text below to your styles.css file (the new thing here is **clear:both**, which tells this content to go below all content on both the left and right of the page).

```
div#footer {
    clear: both;
    text-align:center;
    padding: 5px;
    border-top:5px solid #663300;
    }
```

In your index.html file, wrap the line "Official page of Drake University" in the proper div to make this footer work.

21. By now, your page should be looking like an actual web page (let's hope). Some of these design choices you probably wouldn't do on a real page, but we did them in the name of learning.

Now let's do one more thing and be done. A current trend in web design is to have floating Facebook and Twitter logos on your page that link to your Facebook and Twitter pages. Here's how we can create those. Somewhere in your styles.css file, write the following code:

```
img#thumbnail {
    position: fixed;
    right: 15px;
    top: 15px;
    border: 0;
    }
```

The new thing here is fixed positioning. Even when you scroll, images fixed to a position stay in that position on the browser.

In your index.html file, add the following (really you can put this anywhere, but we will put it at the very bottom before we close our body tag just to avoid any confusion:

```
<a href="http://www.facebook.com" border="0"><img id="thumbnail"
src="images/facebook-logo.png" width="100" /></a>

</body>
```

You should see a floating Facebook logo when you reload your page. If you scroll down the page, the logo should stay in the same place.

That concludes our CSS lesson. If everything worked OK, you should have a page that looks like the one on the next page. If you have any questions, don't hesitate to ask Chris. Email him at chris.snider@drake.edu.

Upload your entire **css-site folder** to your site. If you do this correctly, I should see your page when I go to www.yourdomain.com/css-site. Notice I don't need to put .html at the end because this is an index file inside a folder called css-site.

Meredith Hall

Meredith Hall is home to the School of Journalism and Mass Communication at Drake University. It houses two lecture halls, three computer labs, conference rooms, The E.T Meredith Center for Magazine Studies, professor's offices, the dean's office, radio and television production studios and the SJMC Reading Room in addition to student publication offices.

Almost 95 percent (94.9) of 2006 Drake journalism graduates reported being employed in the field or in graduate school, according to a recent survey by the university. Of these, 89.5 percent reported having had an internship while in school.

The SJMC's magazine program has achieved national prominence. The Accrediting Council on Education in Journalism and Mass Communication (ACEJMC) team that visited in 1999 called the SJMC a "real standout" and termed Drake's Magazines program the strongest undergraduate sequence in the country. Drake student magazines THiNK and 515 won 2007 Pacemaker awards in Washington, D.C.

94.1 The Dog

The School of Journalism and Mass Communication is also home to **94.1 The Dog**, which operates under the call letters KDRA-LP FM. The station launched in August 2006 after having existed as an internet station, KDCS Bulldog Radio. 94.1 The Dog is broadcast at 80 watts from a tower atop Meredith Hall, the home of Drake's SJMC.

An agreement with the Federal Communications Commission (FCC) allows Drake to utilize the frequency from 4 p.m. to 4 a.m. weekdays and all day Saturday, while Grand View University controls the frequency the rest of the week under the call letters KGVC-LP. Drake students schedule 24 hours of programming under "The Dog," broadcasting online and on channel 12 on closed-circuit television on campus even when not broadcasting on the frequency.

Meredith Hall hours

- Monday: 7 a.m. - 10 p.m.
- Tuesday: 7 a.m. - 10 p.m.
- Wednesday: 7 a.m. - 10 p.m.
- Thursday: 7 a.m. - 10 p.m.
- Friday: 7 a.m. - 10 p.m.
- Saturday: CLOSED
- Sunday: 2 p.m. - 10 p.m.