CSS

October 19

Review of CSS

- CSS was designed to separate the content of HTML pages from their appearance
- We are primarily dealing with external style sheets in this course
  - External style sheets are nice because you can apply them to multiple web pages easily
  - Changing the external style sheet will change the look of all of your pages

CSS Rules

- Style sheets are made up of rules
- Rules have two main parts:
  - Selector
  - Property: Value;
- Basic format for a rule looks like:

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

Type Selectors

- First selectors we saw were *type selectors*
  - Basically it's the tag we want to apply the rules to
  - Ex.
    - p, body, em, h1, h2, etc.
- Easiest to use

```
p       {
        color: blue;
        font-family: serif;
}
```

- Gets applied to all paragraphs

Class Selector

- We've looked at these a lot
- We define a class with a name
- Classes have the same rule format as all CSS rules, but they are defined using "." and a class name.
- Ex.

```
p.comment        {
    font-size: smaller;
    color: gray;
}
```

Class Selectors, cont.

- To apply a class to a tag we use the class attribute
  - The value of the class attribute will be the name of the class we defined in the CSS rule
- Ex.

   ……..
- We can apply a class as many times as we want in a document

ID Selectors

- IDs perform the same function as classes
- They allow us to define a special case of a tag
- Unlike classes, however, IDs can only be used once per page
- Because they can only be used once, we can use them as fragments or anchors to create a point somewhere on the page to “jump” to

IDs, cont.

- IDs are defined using the “#” and the ID name
- tag#Name { ……}
- Ex.

```
div#sidebar {
        float: left;
        background-color: #FFC;
}
```

- To apply an ID to a tag in the HTML document simply set the id attribute = “Name”
  - …….

Rules without a Tag

- In class you've seen things like:
  - .clearall
  - #sidebar
  - #navbar
- By not specifying the tag we are applying it to, we can apply it to any tag we want
  - Or even different types of tags (provided we aren't applying an ID more than once)
  - Ex. <p class = "clearall">

    <br class = "clearall/>

Span vs. Div

- and are the generic containers
- They do nothing until they have attributes applied to them
- is a block-level tag
  - We can put lots and lots of elements inside div
  - Useful for navigation bars
- is an inline tag
  - We can change part of the document without having to create an entirely new <p> or , etc.
  - A good example is changing part of a line of text to be a different color

- What we've done with
  - Created a floating side bar
  - Created a top navigation bar
  - Created multiple columns
  - Created different division on the page
    - Header, footer, main columns, etc.

Summarized Examples

```
div#sidebar{
    float: left;
    color: #FFC;
    background-color: olive;
    width: 15%;
}
```

Typical sidebar (or column)

```
div.column {
    float: left;
    width = “150px”
}
```

We can use this to have equally wide columns, as many as we want.

- Haven't done a whole lot with span
- But we can use it to change parts of a line of text

```
span.funkyRed {
    color: red;
    font-family: cursive;
    font-size: 250%;
    text-decoration: underline
}
```

- When applied to a line of text we'd get big, underlined, red, cursive-family text

Applying

- I **** love **** html. HTML is almost as much **** fun as a rollercoaster, a ski trip to Aspen, or even backpacking around Europe**** after winning the lottery.

- I love html. HTML is almost as much fun as a rollercoaster, a ski trip to Aspen, or even backpacking around Europe after winning the lottery.

- Span can be used to apply inline style changes
- Can be used with a class, an id or with the style attribute
- Using with a class implies that you'll be making this change to your document a few times in a page
- Using it with an id is probably to make a fragment out of it
  - You can provide a way to jump to the middle of the paragraph

Cool Little CSS Bits

- This is just to show you some of the other things built into CSS
- Today we're going to look at how to change the first line and the first letter of a paragraph
  - We could do this with the tag
  - But if we want it for every paragraph, better way to do this

Changing First Line

- Can be used only with block-level tags,
  - Like <p>, <blockquote>, etc.
- Format of selector is:

  tag:first-line {…..}
- Example:

```
p:first-line {
 font-size: 150%;
 color: navy;
}
```

Using :first-line in HTML

- Let's say we have the following css rule

```
p:first-line     {
    color:red;
}
```

To use it in our HTML we do nothing special:

```
    <p>This is some text which spans more than
          one line</p>
```

The result:

This is some text which spans more than
one line

First-Letter

- first-letter is just like first-line
- Again it can only be applied to block-level elements
- Same rule format:

p:first-letter{……

}

First-letter Example

- Let's say we have the following rule:

```
p:first-letter{
    color:red;
    font-size: 200%;
}
```

And our HTML:

```
    <p>This is the first paragraph in our html…</p>
   <p>And here we have a second paragraph</p>
```

The result would be:

This is the first paragraph in our html…

And here we have a second paragraph

Using first-letter and first-line

- first-letter and first-line can be used to create some nice effects on a page
- Because you can apply it to all paragraphs, quotes, etc. all paragraphs would still look the same
- However, if you only want to use it in a small section, like for a comment section, we can still use it with IDs and classes
- p.comment:first-letter{…..}
- blockquote#Rowling:first-line{ …..}
- Applied just like a regular class or ID
  - <p class = “comment”>
  - <blockquote id = “Rowling”>

Questions?