

Intro to CSS

CS/IT 490 WD, Fall 2013

Last update 2013-09-05

Written by Rachel J. Morris
Creative Commons Attribution-NonCommercial-ShareAlike
3.0 Unported License

Breakdown

- What is CSS?
- Firebug / Chrome Inspect
- Where you can write CSS
- Corresponding elements to styles
- Inheriting styles
- Specificity
- Overriding styles

What is CSS?

- CSS stands for Cascading Style Sheets.
- A means for keeping page structure (HTML) separate from style (CSS).
- Allows you to style webpage elements based on *type of element*, *class of element*, or *id of element*.

Firebug / Chrome Inspect

- Make sure to download **Firebug** for Firefox if that is your primary browser.
- Google Chrome's inspect element tool is pretty similar.
- Remember that Safari lets you change the user agent (i.e., to test for ios devices)

Where you can write CSS

- You can add CSS styling into one individual element.
- You can add CSS rules within one specific webpage file.
- You can add CSS rules into an external .css file and link it to multiple webpages.
 - This is the best way to work with CSS.

Where you can write CSS

- Styling an individual element (inline)
 - Might be useful for letting people style their blog-posts; it's a one-time style that doesn't need to be applied to other areas of the webpage.

```
<ul>
  <li>Grocery shopping</li>
  <li style="color: #ff0000;">Pay bills</li>
  <li>Walk dog</li>
</ul>
```

To Do List

- Grocery shopping
- Pay bills
- Walk dog

Where you can write CSS

- Adding rules in the .html page
 - If you wanted common styles across multiple pages, this wouldn't work well.

```
<head>
  <title>Sample</title>

  <style type="text/css">
    .urgent { color: #ff0000; }
  </style>
</head>
```

Css rule for
class 'urgent'

```
<body>

  <h2>To Do List</h2>

  <ul>
    <li>Grocery shopping</li>
    <li class="urgent">Pay bills</li>
    <li>Walk dog</li>
  </ul>

</body>
```

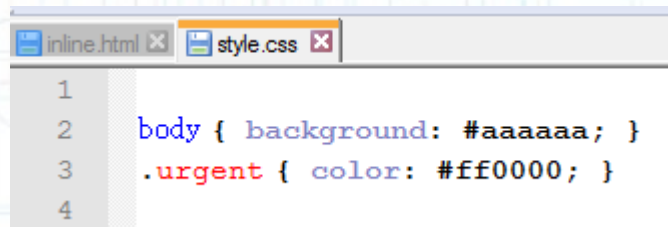
Element
given special
class

To Do List

- Grocery shopping
- Pay bills
- Walk dog

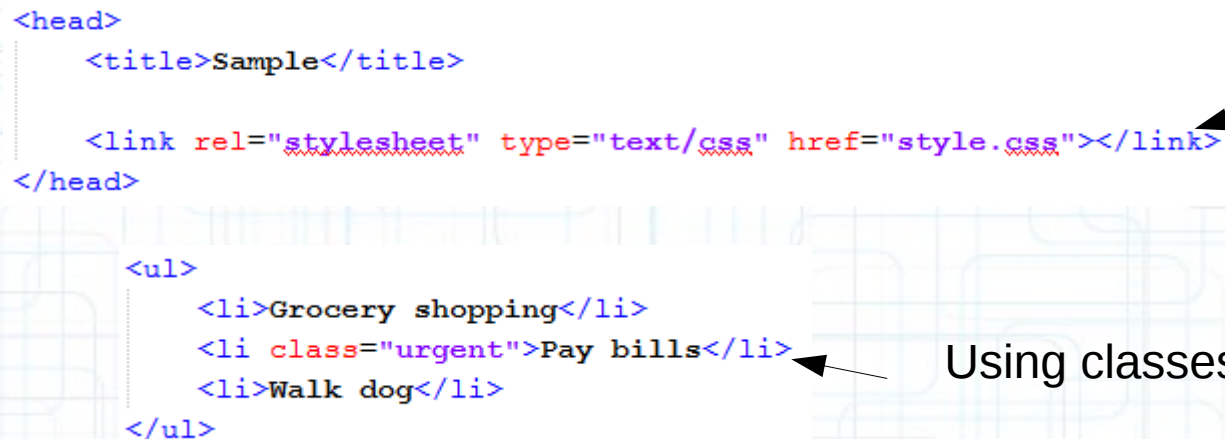
Where you can write CSS

- Adding rules in .css file
 - You can add rules in a .css file and link to them in the .html file.



```
1
2 body { background: #aaaaaa; }
3 .urgent { color: #ff0000; }
4
```

Creating styles



```
<head>
  <title>Sample</title>

  <link rel="stylesheet" type="text/css" href="style.css"></link>
</head>

<ul>
  <li>Grocery shopping</li>
  <li class="urgent">Pay bills</li>
  <li>Walk dog</li>
</ul>
```

Linking the
.css file into
the .html file

Using classes

Where you can write CSS

- Adding rules in .css file
 - You can add rules in a .css file and link to them in the .html file.

To Do List

- Grocery shopping
- Pay bills
- Walk dog

Corresponding elements to styles

- You can style an element in three ways:
 - Style all of a type of element - Just use the element name

```
p { color: #0000ff; font-size: 40px; }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Praesent et diam eget libero egestas mattis sit amet vitae augue.

Corresponding elements to styles

- You can style an element in three ways:
- Style all items that belong to a class

```
2  p { color: #0000ff; font-size: 40px; }
3
4  .news { color: #ff0000; }
5  p.news { font-size: 20px; font-style: italic; }
6  li.news { text-decoration: underline; }
```

Praesent et diam eget libero egestas
mattis sit amet vitae augue.

Praesent et diam eget libero egestas mattis

- Bullet 1
- Bullet 2
- Bullet 3

Corresponding elements to styles

- You can style an element in three ways:
- Style all items that belong to a class
- Notice that the class name goes after a period “.”
- You can also combine ***elementType.className*** to make a style only apply to certain elements of that class type.

```
2  p { color: #0000ff; font-size: 40px; }
3
4  .news { color: #ff0000; }
5  p.news { font-size: 20px; font-style: italic; }
6  li.news { text-decoration: underline; }
```

Corresponding elements to styles

- You can style an element in three ways:
- Style one specific element, with a specific ID
- To specify ID, use the # hashtag/pound-sign. These correspond to the **id** attribute of an element

```
#pageError { background: #ffaana; }  
<div id="pageError">
```

ERROR: Could not find "important.js"

```
h2#about { color: #00ff00; }  
<h2 id="about">ABOUT</h2>
```

ABOUT

Soandso Inc. is the best company ever.

```
h2#jobs { color: #ff00ff; }  
<h2 id="jobs">JOBS</h2>
```

Soandso Inc. is not currently hiring.

Inheriting styles

Styles can also be inherited. If you set up a style for <p>

```
p { color: #0000ff; font-style: italic; }
```

Then set up a style for p.news (elements of type *p* whose class is *news*)

```
p.news { color: #ff0000; text-decoration: underline; }
```

Then it inherits styles from both. If two styles clash, the one belonging to the ***more specific*** rules wins out.

Normal paragraph

News Paragraph

News paragraph inherits the *italics* from its parent, but its color is overridden by the p.news style.

Specificity

- Rules can be overridden in CSS.
Whichever rule is *more specific* wins out.

```
a { font-size: 20px; }  
nav ul li a { font-size: 30px; }
```

```
<p>Paragraph</p>
```

Paragraph

```
<p><a href="asdf.html">Normal Link</a></p>
```

Normal Link

```
<nav>
```

```
<ul>
```

```
<li><a href="index.html">Home</a></li>
```

```
<li><a href="about.html">About</a></li>
```

```
<li><a href="pizza.html">Pizza</a></li>
```

```
</ul>
```

```
</nav>
```

- [Home](#)
- [About](#)
- [Pizza](#)

Specificity

- Rules can be overridden in CSS. Whichever rule is *more specific* wins out.

```
a { font-size: 20px; }
```

← All links

```
nav ul li a { font-size: 30px; }
```

← Links contained within an
Unordered List within a Nav

- You can force adherence with !important, but it is not good practice.

```
li a { color: #ff0000 !important; }
```

Overriding styles

- Many elements have default styles. You can override these.

Normal paragraph

```
p { margin: 0; }
```

→ No margin on Paragraph

```
hr { margin: 0; }
```

→ No margin on Paragraph or Horizontal Rule

Multiple Classes

- Elements can also have multiple classes. Just separate each class name by a space:

```
<div class="content-wrapper full-page highlight">  
</div>
```

Duplicating CSS

- You can also list several rules, separated by commas, with one style definition { }:

```
p, h1, h2, h3 { font-style: italic; }
```

```
p.classOne, p.classTwo { font-weight: bold; }
```

Pseudoclasses

- You can also add on pseudoclasses that may describe different states an element may be in.
- For example:
 - hover (mouse hover)
 - first-child (First child of some type)
 - before (Insert style before an element)
 - after (Insert style after an element)

Pseudoclasses

- Links are most common:

```
a,  
a:active { color: #000000; }  
  
a:hover { text-decoration: underline; }  
  
a:visited { color: #ff00ff; }
```

Duplicate style for a
and a:active

Hovering over a link

A link that has
already been visited

Summary

- Style an element:
`h2 { font-size: 30px; }`
- Style a class:
`p.news { font-style: italic; }`
`nav .link { color: #ff0000; }`
- Style an id:
`#header-about { background: #000000; }`
`ul#main-list li { list-style-type: none; }`

References

CSS Reference, W3Schools.com
<http://www.w3schools.com/cssref/>