TCS170D: Web Design

Lecture 3: Other CSS Considerations

3 Ways to include CSS

- External Style sheets

- Embedded Styles

- Inline Styles

External Style Sheets

- Text files with a .css extension

- Include in your page using a <link> tag in the head of a document

```html
<link href="main.css" rel="stylesheet" type="text/css" media="screen">
```

- href→ location of css file

- rel→tells the browser it is a stylesheet

- type→ tells the browser it contains CSS (can be omitted in HTML5)

- media→ tells the browser which medium to apply styles for (e.g. "screen", "print")

External Style Sheets

- Can easily be applied to multiple pages, so best for site-wide styles.

Embedded Styles

Go inside of a <style> element on a page. Only apply to elements on that page:

```
<html>
<head>
<style>
  body {font-family: Arial; font-size: 100%;}
  h1, h1 {font-weight: normal; color: red;}
  p {font-size: .9em; margin-bottom: 1.2em;}
  .pullquote {background: yellow; padding:10px}
</style>
</head>
<body>
```

Inline Styles

- Goes inside of the style attribute of an HTML element

- No selector needed because the element it applies to is obvious

```
<html>
<head>
</head>
<body>
<div>
<p style="font-size: .9em; margin-bottom: 1.2em;">
...</p>
</div>
</body>
</html>
```

- Inline style use in static websites is highly discouraged because it is hard to maintain.

The Cascade

- With so many different selectors and ways to include CSS, many different CSS rules can target the same HTML elements.

- You might call this a 'Cascade' of styling rules.

- If two identical selectors specify conflicting style rules, the last rule applied wins.
  - The browser goes through a page sequentially, applying any CSS rules it finds as it goes.

Inheritance

- Child elements inherit attributes given to their parents.

Child Rule Overwrite Parent Rules

Inheritance Leads to Efficient Code

```
h1 {font-family:Arial; color: #333;}
h2 {font-family:Arial; color: #333;}
h3 {font-family:Arial; color: #333;}
p {font-family:Arial; color: #333;}
li {font-family:Arial; color: #333;}
address {font-family:Arial; color: #333;}
label {font-family:Arial; color: #333;}
```

```
body {font-family: Arial; color: #333;}
```

Specificity

- Each selector gets a score. IDs are worth 100, classes, 10, elements 1
- The selector with the higher specificity score has its rules apply to the element
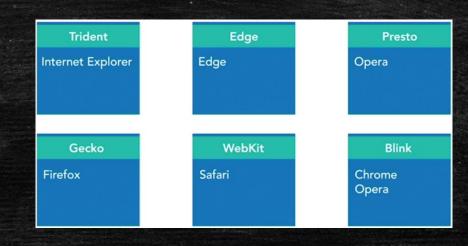
Selector	ID	classes	elements	specificity
body	0	0	1	1
#mainContent	1	0	0	100
quote	0	1	0	10
div p	0	0	2	2
#sidebar p	1	0	1	101

Browser Rendering Difficulties

- Browsers have different rendering engines, which parse CSS somewhat differently. As a result, pages appear differently in different browsers.

```css
background: -webkit-linear-gradient(red, yellow); /* For Safari 5.1 to 6.0 */
background: -o-linear-gradient(red, yellow); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(red, yellow); /* For Firefox 3.6 to 15 */
background: linear-gradient(red, yellow); /* Standard syntax */
```

- Test in different browsers

- Caniuse.com

Trident	Edge	Presto
Internet Explorer	Edge	Opera

Gecko	WebKit	Blink
Firefox	Safari	Chrome Opera

CSS Specifications

- Written by browser developers, not authors

- Pretty confusing, but can be useful

- https://www.w3.org/TR/css-fonts-3/

- Index, Property Indexes are good peruse

Font family property:

Name:	**font-size**
Value:	<absolute-size> | <relative-size> | <length> | <percentage>
Initial:	medium
Applies to:	all elements
Inherited:	yes
Percentages:	refer to parent element's font size
Media:	visual
Computed value:	absolute length
Animatable:	as length

Backus Normal Form (BNF)

- '|' read as OR

- Inside '<>' is computed value

Value: <absolute-size> | <relative-size> | <length> | <percentage>

- Brackets '[ ]' group terms together, and denote that whatever is inside of it is optional

- https://www.w3.org/TR/css-syntax-3/