CSS frameworks and preprocessors

CSS

CSS Rule

```
body p { text-align: center; }
```

CSS Selector

body p { text-align: center; }

CSS Property

body p { **text-align**: center; }

CSS Value

body p { text-align: **center**; }

CSS Classes

Can be applied to more than one element on a page.

Can assign more than one to an element

```
<p class="class-name another">Test</p>
```

```
.class-name { text-align: center; }
```

CSS Ids

Can be applied to only one element on a page.

Can only have one per element

```
<p id="id-name">Test</p>
```

```
#id-name { text-align: center; }
```

CSS Downsides

- Layouts are hard to get right
- Different browsers render things differently
- No variables
- No good way to have reusable styles

Semantic HTML

"Semantic HTML is the use of HTML markup to reinforce the semantics, or meaning, of the information in webpages rather than merely to define its presentation (look)."

Separation of content and presentation.

Markup should be meaningful.

Semantic HTML

Very, very, very bad:

```
<p color="#FF0000" style="text-align:center" id="left-text">
  Some text
</p>
```

Good:

```
<p id="example">
  Some text
</p>

// style.css
#example {
  color: #FF0000;
  text-align: center;
}
```

please never use Javascript for layout

make sure that your site works without Javascript

Learn about:

- Progressive enhancement
  - http://en.wikipedia.org/wiki/Progressive_enhancement
- Graceful Degradation

Understand and design for the box model in order to have a painless web design experience

CSS Reset Stylesheet

"The goal of a reset stylesheet is to reduce browser inconsistencies in things like default line heights, margins and font sizes of headings, and so on. "

http://meyerweb.com/eric/tools/css/reset/

http://developer.yahoo.com/yui/reset/#start

CSS Frameworks

Two big ones:

- Blueprint
- 960.gs

There are many more

CSS Frameworks

Benefits:

- Easy to quickly design layouts
- Have nice styles for form elements
- Have typographic styles

Downsides:

- Non-semantic HTML
- Are just a starting point
- Still need to work with straight CSS and its shortcomings

CSS Preprocessors

Sass: http://sass-lang.com/

- Built with Ruby, but can be used in any kind of project
- Very mature. Has been through multiple iterations
- Has two syntaxes
  - indented
  - bracketed (CSS superset)

LESS: http://lesscss.org/

- Was originally a Ruby project, now built with Javascript
- Became popular because Sass didn't yet have the bracketed syntax
- Can be used server-side via node.js
- Can be compiled on the client-side, though this requires Javascript

Compass

A CSS metaframework
A CSS authoring framework
A Sass framework

Compass

- Written in Sass
- Makes it possible to use CSS frameworks while leaving markup semantic
- Provides mixins to easily create cross-browser CSS and use CSS3 features like border-radius or text-shadow
- Has mixins for typography
- Much more...

William Melody

hi@williammelody.com

http://github.com/alphabetum

http://www.linkedin.com/in/williammelody