LESS & SASS

Jed Willard

http://go.illinois.edu/less_slides
SASS

- SASS = Syntactically Awesome Stylesheets
- Extension of CSS
- Two syntaxes (SCSS & SASS)
- Must install Ruby & Sass
- http://sass-lang.com/
LESS

● LESS != cool acronym

● Extension of CSS, similar to SASS

● Runs client-side (javascript) or server-side (node.js)

Why use these extensions?

MORE POWER!!!
Powerful Additions

- Variables
- Mixins
- Nested Rules
- Functions & Operations
- Namespaces
Key Areas of Usage

Variables
● Fonts (all aspects: family, size, etc)
● Colors

Mixins
● Repetitive CSS properties (border-radius, borders, etc)

Nested Rules
● Mimics DOM tree, much leaner CSS

Functions, Operations, Namespaces
● Anywhere applicable, create reusable blocks of CSS, font-size operations, responsive grid calculations
Get Started

- Use same software tools you currently use.
- Use your current CSS skills.
- Pick one: LESS or SASS
Recommended Workflow

- Write your CSS the SASS/LESS way during development
- Compile LESS/SASS to CSS
- Link to compiled CSS instead of relying on JS or server-side processing
Variables

@color: #4D926F;

#header {
  color: @color;
}

h2 {
  color: @color;
}

Mixins

.rounded-corners (@radius: 5px) {
    border-radius: @radius;
    -webkit-border-radius: @radius;
    -moz-border-radius: @radius;
}

#header {
    .rounded-corners;
}

#footer {
    .rounded-corners(10px);
}
Nested Rules

#header {
  h1 {
    font-size: 26px;
    font-weight: bold;
  }
}

p {
  font-size: 12px;
  a {
    text-decoration: none;
    &:hover {
      border-width: 1px
    }
  }
}
}
Functions & Operations

@the-border: 1px;
@base-color: #111;
@red:    #842210;

#header {
  color: @base-color * 3;
  border-left: @the-border;
  border-right: @the-border * 2;
}

#footer {
  color: @base-color + #003300;
  border-color: desaturate(@red, 10%);
}
Namespaces

#bundle {
  .button () {
    display: block;
    border: 1px solid black;
    background-color: grey;
    &:hover { background-color: white }
  }
  .tab { ... }
  .citation { ... }
}

#header a {
  color: orange;
  #bundle > .button;
}
Pros

- Added power from extensions
- Lighter CSS
- Efficiency
Cons

- Potentially more processing time
- LESS = client-side usage requires another JS call
- Dumps generated CSS into document head

All of these cons are alleviated by using LESS/SASS only during development.
Examples

- Awesome Demo File
- A more realistic example
- Drupal's Zen theme (7.x-5.x-dev files)
Resources

LESS

SASS

An Introduction to LESS, and Comparison to SASS

SASS/LESS Comparison

Getting Started with SASS and Compass

Compass