

# Programming with Sass

# Why Program in Sass?

- » Create re-usable components
- » Let the computer do the boring work
- » Do things that wouldn't be feasible in raw CSS

```
div:nth-child(1) { background: hsl( 0deg, 0.5, 0.5); }
div:nth-child(2) { background: hsl( 60deg, 0.5, 0.5); }
div:nth-child(3) { background: hsl(120deg, 0.5, 0.5); }
div:nth-child(4) { background: hsl(180deg, 0.5, 0.5); }
div:nth-child(5) { background: hsl(240deg, 0.5, 0.5); }
div:nth-child(6) { background: hsl(300deg, 0.5, 0.5); }
```

```
$total: 6;
@for $i from 0 to $total {
  div:nth-child(#{ $i + 1 }) {
    $hue: $i * (360deg / $total);
    background: hsl($hue, 0.5, 0.5);
  }
}
```

```
@mixin color-palette(  
    $base-color, $total, $step  
) {  
    @for $i from 0 to $total {  
        &:nth-child(#{ $i + 1 }) {  
            $hue-offset: $i * $step;  
            background: adjust-hue($base-color, $hue-offset);  
        }  
    }  
}
```

```
div {  
    @include color-palette(hsl(0, 0.5, 0.5), 6, 60deg);  
}
```

# API Library

# Google Web Fonts

Can specify

- >> Fonts

- >> Variants

- >> Subsets

# Design an API

- >> Customisable parameters
- >> Concise
- >> Compatible with Ruby Sass and Libsass



# Libsass Compatibility

// doesn't work with libsass

```
@include web-fonts("Open Sans");
```

// doesn't work with libsass

```
@import url(web-fonts-url("Open Sans"));
```

// works with libsass!

```
$url: web-fonts-url("Open Sans");
```

```
@import url($url);
```

# The API

```
// import a single font
```

```
@import url(web-fonts-url("Open Sans"));
```

```
// specify variants for a font
```

```
@import url(web-fonts-url( ("Open Sans": "bold") ));
```

```
// multiple variants
```

```
@import url(web-fonts-url( ("Open Sans": ("500", "600 italic")) ));
```

```
// multiple fonts
```

```
@import url(web-fonts-url(  
    "Open Sans",  
    ("Ubuntu": ("400", italic))  
));
```

# URL Encoding

```
@function url-encode($string) {  
  $replacements: (  
    " ": "%20",  
    " ": "%20",  
    "!": "%21", "#": "%23", "$": "%24", "&": "%26",  
    "'": "%27", "(": "%28", ")": "%29", "*": "%2A",  
    "+": "%2B", ",": "%2C", "/": "%2F", ":": "%3A",  
    ";": "%3B", "=": "%3D", "?": "%3F", "@": "%40",  
    "[": "%5B", "]": "%5D", " ": "%20");  
  
  @each $from, $to in $replacements {  
    $string: str-replace($string, $from, $to);  
  }  
  
  @return $string;  
}
```

# String replacement

```
@function str-replace($string, $find, $replace, $all: true) {  
  $index: str-index($string, $find);  
  @if $index {  
    $before: str-slice($string, 1, $index - 1);  
    $after: str-slice($string, $index + str-length($find), str-length($string));  
    $string: $before + $replace + $after;  
  
    @if $all and not str-index($find, $replace) {  
      $string: str-replace($string, $find, $replace);  
    }  
  }  
  @return $string;  
}
```

# No Module System

Imports use file paths

```
@import "bower_components/sass-web-fonts/web-fonts";
```

All functions are public

```
@function wf-str-replace...
```

# Modules in Sass 4

<https://github.com/sass/proposal.module-system>

@use module loading system

@use sass-web-fonts;

Functions whose names start with - or \_ are private.

@function -str-replace...

# Unit Testing

# Bootcamp (*no longer maintained*)

<https://github.com/thejameskyle/bootcamp>

```
@include describe("Math Power") {  
  @include it("should expect positive values to be calculated correctly") {  
    @include should( expect( power( 10, 2) ), to( equal( 100 ) ));  
    @include should( expect( power(  2, 2) ), to( equal(   4 ) ));  
    @include should( expect( power(0.5, 2) ), to( equal( 0.25 ) ));  
  }  
}
```



# git diff

From <https://github.com/sass-mq/sass-mq>

test/

test-1.css

test-1.scss

test-2.css

test-2.scss

Compile, then run `git diff` and check for changes.

# SassUnit

<https://github.com/alyssais/sassunit>

- » Same structure as `git diff` approach
- » Built on an existing testing library
- » Multiple Sass implementations at once (soon!)

# Sass Web Fonts

<https://github.com/alyssais/sass-web-fonts>

# Use @ statements

- >> @debug: print arbitrary output to the terminal
- >> @error: halt execution
- >> @warn: print warning to the Terminal

# sass -i

```
~ $ sass -i
```

```
>> 3px + 1pt
```

```
3.25pt
```

```
>> adjust-hue(red, 40deg)
```

```
#ffaa00
```

```
>>
```

# Extending Sass

- >> Hook into the Sass compiler
- >> Write in Ruby (Sass) or C (libsass)
- >> Not portable