



Intro to Sass

Part II

Download Class 2 Files:

erindepew.github.io/gdi-sass-intro/pdfs/class2

Multiple Arguments

```
@mixin gradient($color1, $color2) {  
  background-image: linear-gradient($color1, $color2, $color1);  
}
```

Default Arguments

```
@mixin gradient($color1: #fff, $color2: #666) {  
  background-image: linear-gradient($color1, $color2, $color1);  
}
```

Interpolation

```
@mixin rotate($degree, $position) {  
  transform: rotate(#{$degree}deg);  
  transform-origin: $position;  
}
```

```
$type: 'large';  
  
.button--#{ $type } {  
  height: 100px;  
}
```

```
$sidebar-width: 250px;  
  
.main-content {  
  width: calc(100% - #{$sidebar-width});  
}
```

@extend

```
%headline {  
  font-size: 2em;  
  font-weight: bold;  
}  
.lead-story-headline {  
  @extend %headline;  
  text-decoration: underline;  
  text-transform: uppercase;  
}
```

Questions?

Let's Develop It

- Create new mixins using the following:
 - Multiple arguments
 - Default arguments
 - Interpolation
 - @extend to include common styles
- Use these mix-ins in your styles.scss file
- Compile to CSS and refresh your index page to see your changes

Sprites

Sprite image:



CSS for icon:

```
.icon {  
display: block;  
background: no-repeat url(images/youtube_sprite.png) -395px -114px;  
background-size: auto;  
width: 18px;  
height: 18px; }
```


Sprite Mixins

If you set up your Photoshop sprite to use grid lines, you can use Sass to easily position your sprite image without lots of trial and error.

```
@mixin sprite-position($x:0, $y:0) {  
  $gridSize: -32px;  
  $offsetX:  $x * $gridSize;  
  $offsetY:  $y * $gridSize;  
  background-position: $offsetX $offsetY;  
}  
@mixin sprite-image($file, $x:0, $y:0) {  
  background-image: url("../images/icons/#{$file}");  
  @include sprite-position($x, $y);  
}
```

Media Queries

@media blocks do not need to be declared at the root level of the stylesheet, which helps keep them in sync with the elements they're modifying.

```
body {  
  article {  
    p {  
      font-size: 100%;  
      color: black;  
      padding: 10px;  
  
      @media (max-width: 768px) {  
        font-size: 150%; // use larger text for smaller screens  
      }  
    }  
  }  
}
```

Media Queries

Compiled CSS:

```
body article p {  
  font-size: 100%;  
  color: black;  
  padding: 10px;  
}  
  
@media (max-width: 768px) {  
  body article p {  
    font-size: 150%;  
  }  
}
```

Break Time!

Stand up and stretch - we'll resume in 5 minutes



Conditionals

@if, @else if, and @else

```
@mixin opacity($value: 0.5) {  
  @if $value == transparent {  
    opacity: 0;  
  } @else if $value == opaque {  
    opacity: 1;  
  } @else {  
    opacity: $value;  
  }  
}  
@include opacity(transparent);
```

Conditionals

Rewrite existing mixins to use conditionals

```
@mixin arrow($direction: right) {  
  @if $direction == right {  
    //right arrow styles  
  }  
  @else if $direction == left {  
    // left arrow styles  
  }  
}
```

For Loops

Sass code:

```
@for $i from 1 through 3 {  
  .column-#{ $i } { width: 2em * $i; }  
}
```

CSS output:

```
.column-1 {  
  width: 2em;  
}  
.column-2 {  
  width: 4em;  
}  
.column-3 {  
  width: 6em;  
}
```

For Loops

- Use loops to make columns that fit within the layout or add links to the footer nav.
- Experiment with math in your loop's styles - change the width, padding, and even the font-size

```
@for $i from 1 through 3 {  
  .column-#{ $i } { width: 2em * $i; }  
}
```




Loops through a list of items and creates styles for each item in the list

Sass code:

```
@each $icon in youtube, twitter, facebook {  
  .icon-$icon {  
    background-image: url('#{ $icon }.png');  
  }  
}
```

CSS output:

```
.icon-youtube {  
  background: url('youtube.png');  
}  
.icon-twitter {  
  background: url('twitter.png');  
}  
.icon-facebook {  
  background: url('facebook.png');  
}
```

@each

- Create background image styles for each article
- Try [Subtle Patterns](#) for images to use
- Write these background styles with @each and a list
- Bonus: style the social media icons using @each and @if logic

```
@each $woman in ada, grace, frances, barbara, anita, maria {  
  .#{$woman}-bg {  
    background-image: url('images/#{$woman}.png');  
  }  
}
```

Advanced Functions

- This is an example of the `index()` function, but there are many other functions you can use.
- [Sass function documentation](#)

```
$elements: project-covers, user-tooltip, sorting-bar, modals, navigation;  
  
.user-tooltip {  
  z-index: index($elements, user-tooltip);  
}
```

Advanced Functions

This is an even more advanced example combining a custom function, lists, variables, and error handling.

```
@function z($list, $element) {  
  
  $z-index: index($list, $element);  
  
  @if $z-index {  
    @return $z-index;  
  }  
  
  @warn 'There is no item "#{element}" in this list; choose one of: #{ $list }';  
  @return null;  
}
```

