

# Sass

## Walk through

### Part 1 – making it run

- npm i sass
- create folders scss, css
- create a file main.scss

```
.success {  
color: green;  
}  
.success-dark {  
color: darken(green,10%);  
}
```

- See the generated CSS

### Part 2 – use partials

- Add HTML File and link to the CSS
- Add a file \_buttons.scss
- @import "buttons"
- Note that CSS imports makes multiple, delayed network request, but scss is compiling to single css main.css file.

### Part 3 – understand nesting

```
.toy{  
  .txt {  
    color :green;  
  }  
  
  img {  
    transition:all .8s ease-in-out;  
    max-width:100px;  
    &:hover{transform:rotate(90deg);}  
  }  
  
  &-container{background-color:#dbbebe;}  
  &-container:hover{background-color:lightcoral}  
  
  &::before {  
    content:url('../img/best.png');  
  }  
}
```

```
}
```

## Part 4 – Use Variables

Variables allow us to update repeated values in one place. They can be block scoped, overridden, and even interpolated.

```
$color-text-default:#424954;
$color-text-light:#EBEBEB;

$color-primary:#8465D9;
$color-danger:#E83043;

$font-family-primary:    Roboto, 'Open Sans',Helvetica Neue,sans-serif;
$font-family-secondary:Georgia,TimesNew Roman,serif;

// Interpolation:
$puki: puki;
.#{ $puki }:after {content:"#{ $puki }";}
```

## Part 5 – Play with Mixins

Mixins are reusable chunks of code that are included, similar to calling a function:

```
@mixin make-box($bcolor){
    padding:1.4em;
    border-radius:0.8em;
    border:1px solid $bcolor;
}

.item-box {
    @include make-box(gray)
}
.item-sale-box {
    @include make-box(tomato)
}
```

Play with it

## Part 6 – Loops

Writing similar classes with minor variations, like utility classes, can be a pain to write and update, this is where @for loops become handy:

```
@for $i from 1 through 10{
  .col-#{ $i }{
    width:100% / $i;
  }
}

////////////////////////////////////

@for $i from 1 through 10{
  $value:0.5*$i;

  [class~="mt-#{ $value }"]{
    margin-top:#{ $value }rem;
  }
}
```

Each loops are also supported:

```
$sales: regular, sale, super-sale;
@each $sale in $sales{
  .#{ $sale }-icon {
    background-image :url("../images/#{ $sale }.gif");
  }
}
```