THE COMPLETE GUIDE

CSS TO SAAS

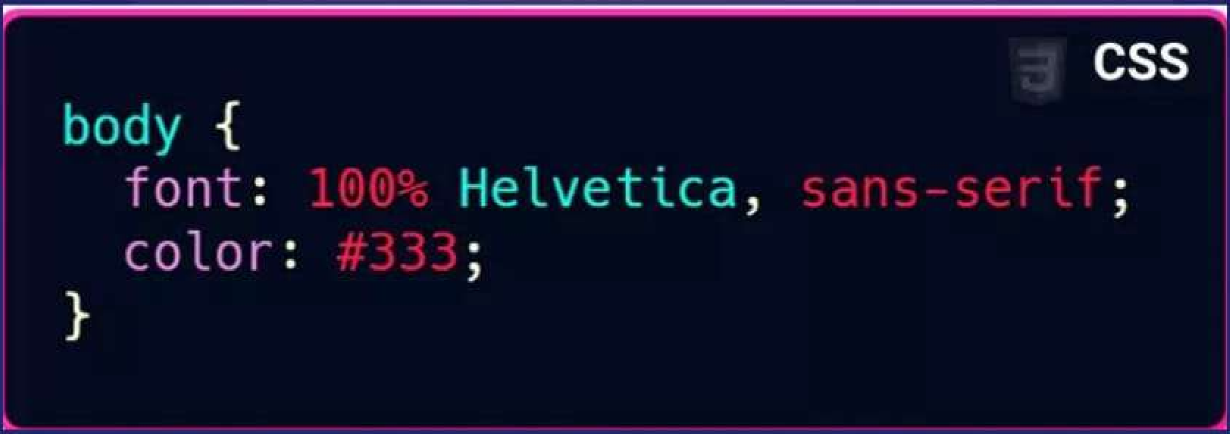
WHAT IS SASS?

SASS stands for Syntactically Awesome Stylesheet. Sass is an extension to CSS, it is a CSS pre-processor & is completely compatible with all versions of CSS.

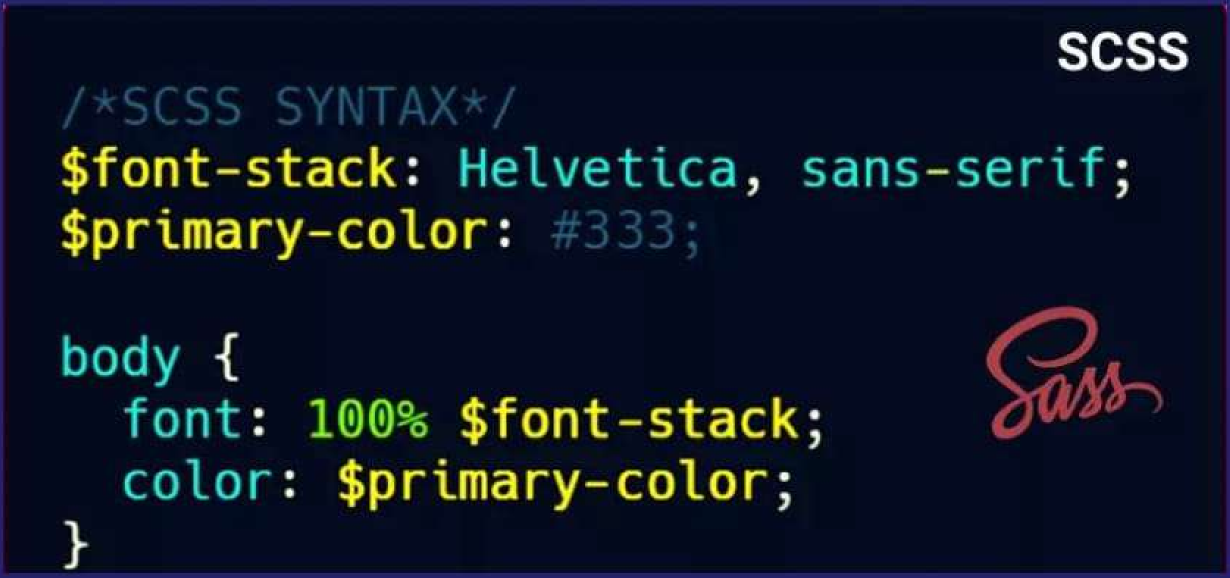
Sass is free to download and use.

Variables →

VARIABLES

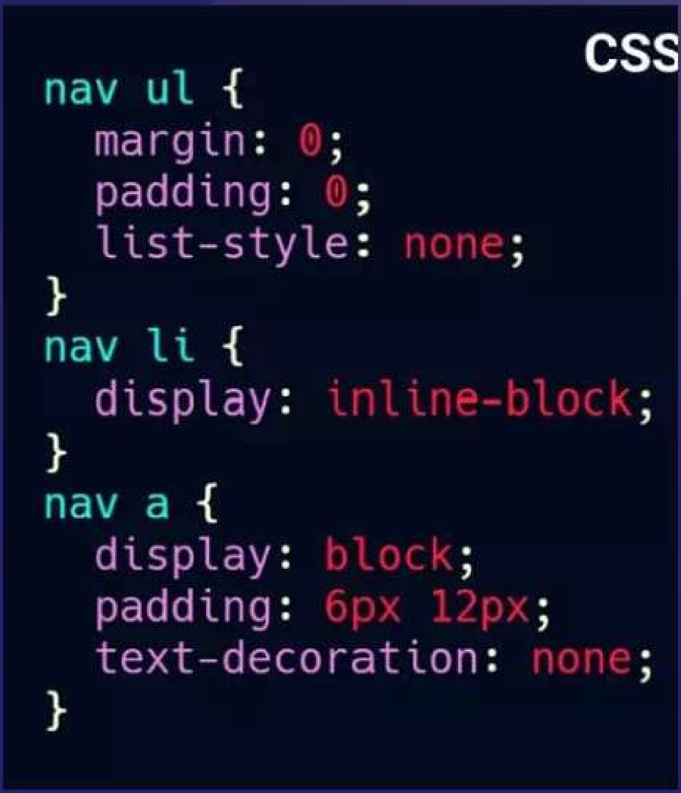


```css
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```



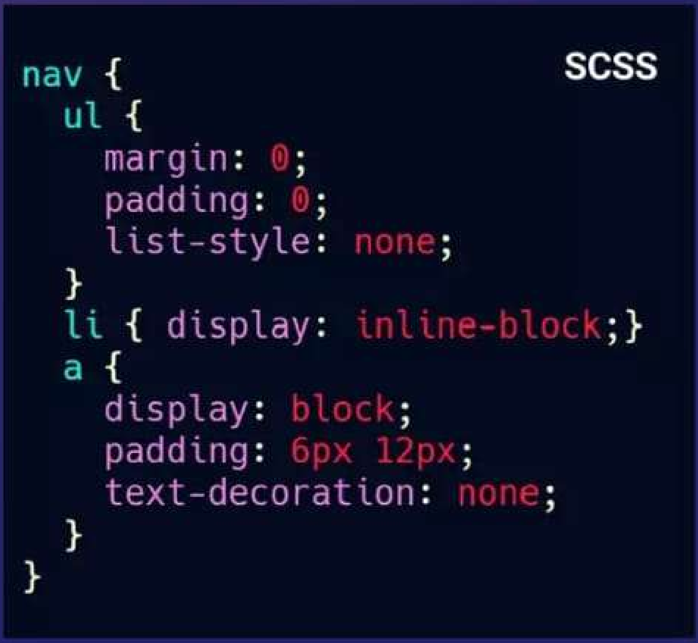
```scss
/*SCSS SYNTAX*/
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

NESTING



CSS

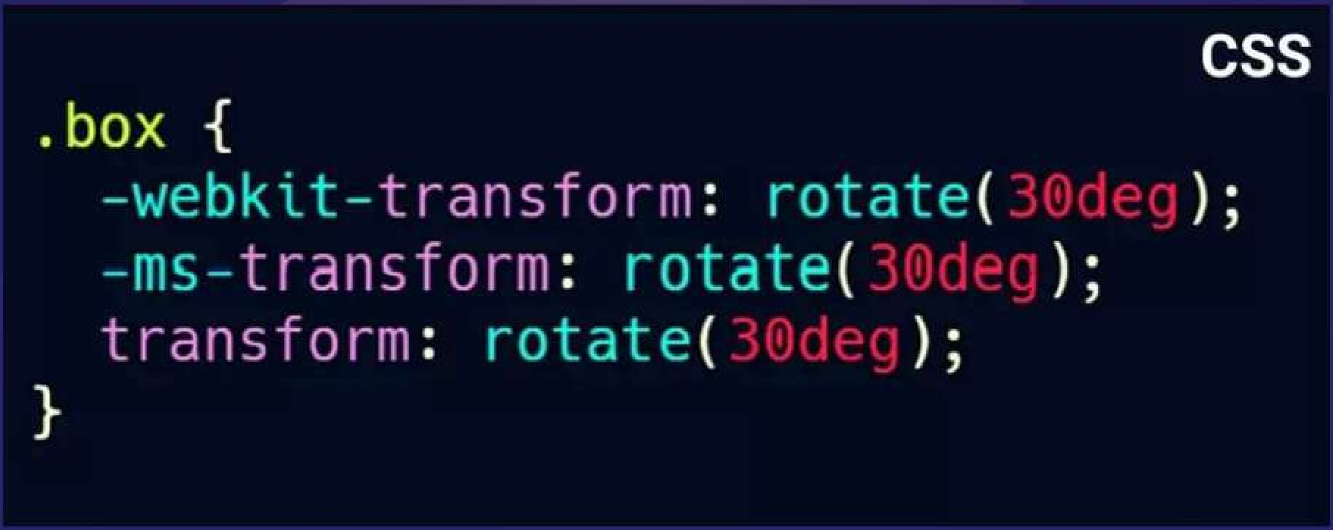
```css
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```



SCSS

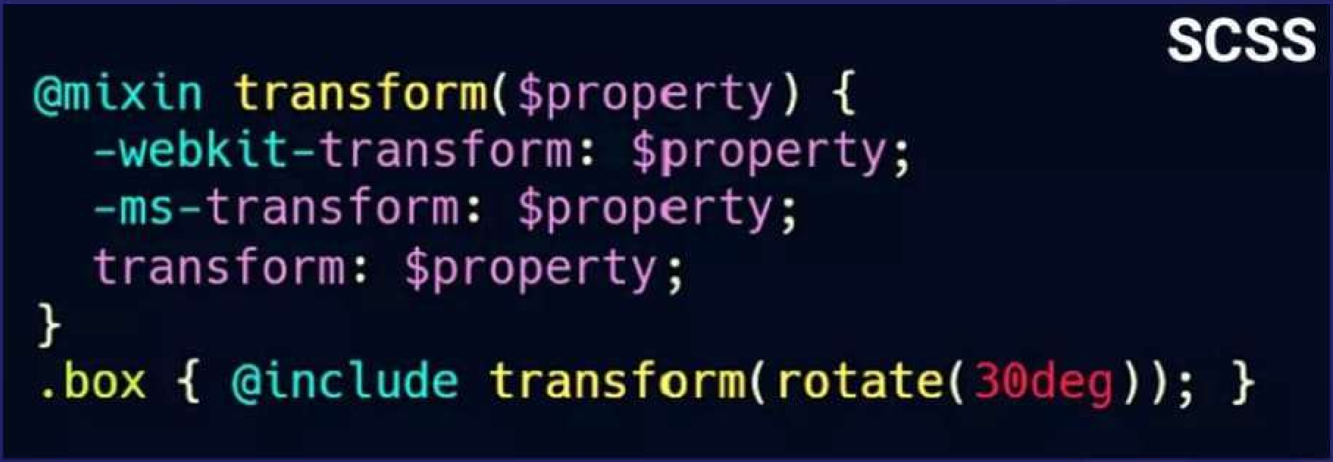
```scss
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li { display: inline-block;}
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

MIXINS



CSS

```css
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}
```

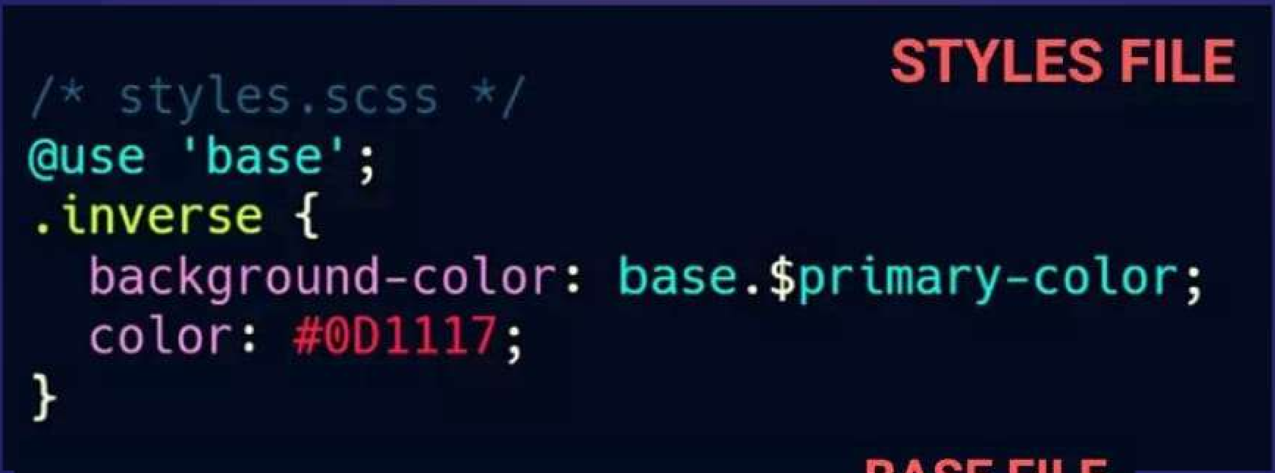


SCSS

```scss
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}
.box { @include transform(rotate(30deg)); }
```

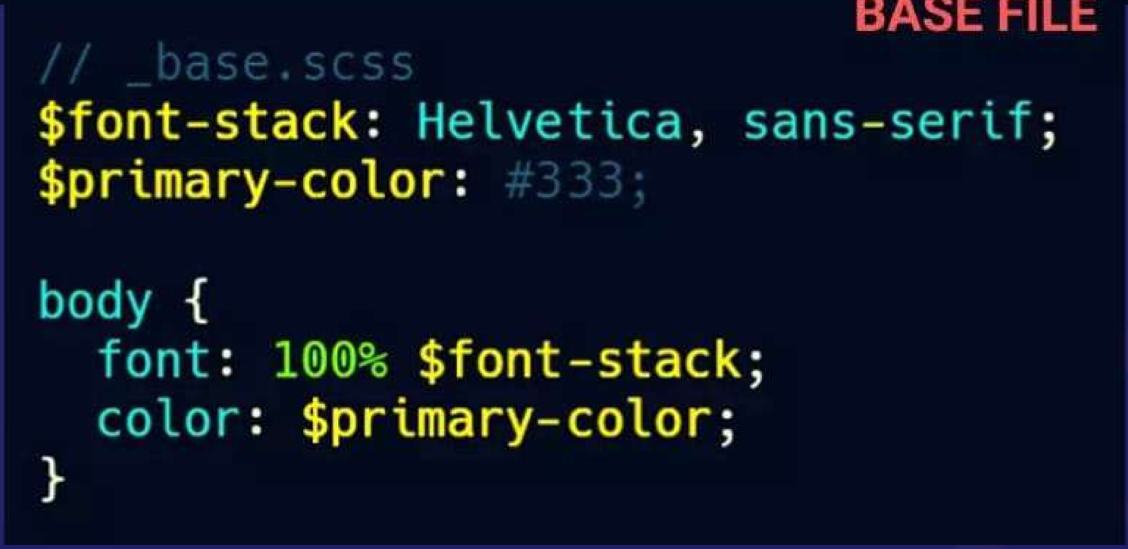
MODULES

You don't have to write all your Sass in a single file. You can split it up however you want with the @use rule.



STYLES FILE

```scss
/* styles.scss */
@use 'base';
.inverse {
  background-color: base.$primary-color;
  color: #0D1117;
}
```

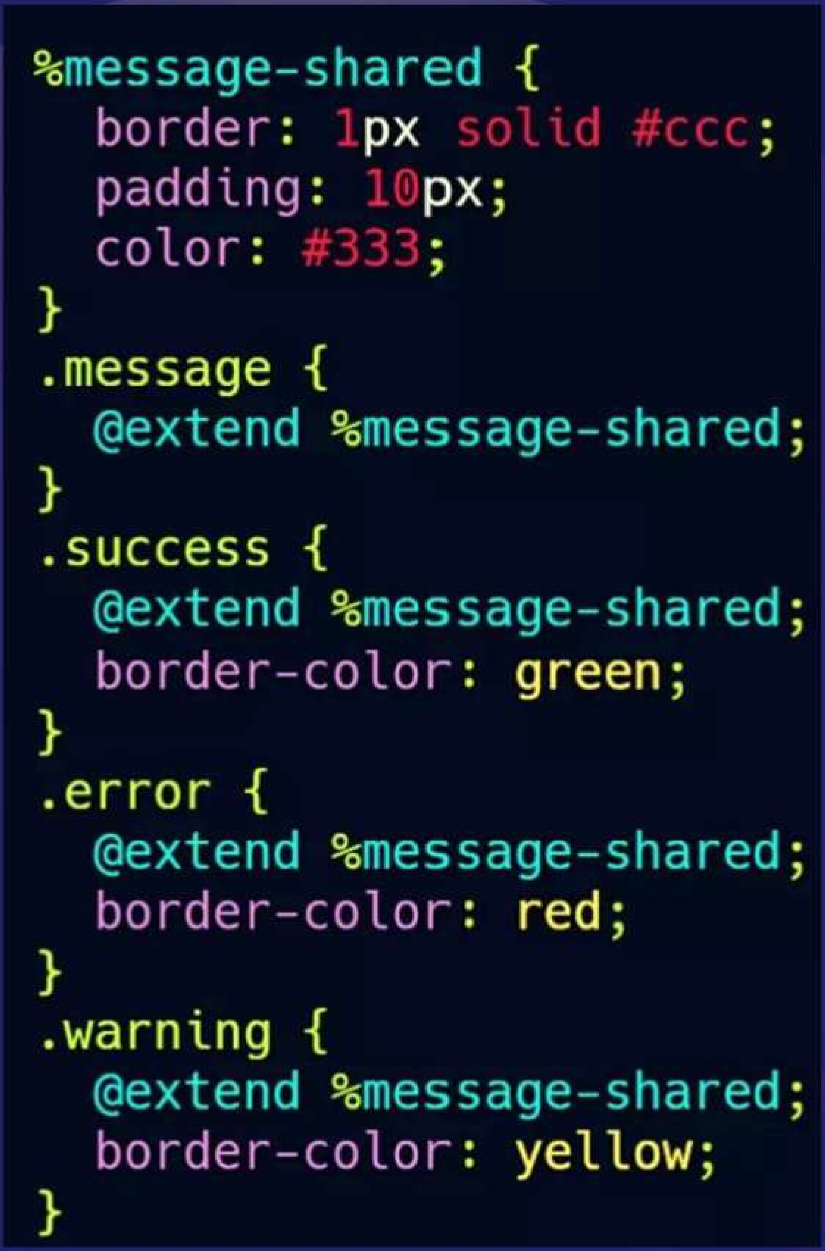


BASE FILE

```scss
// _base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

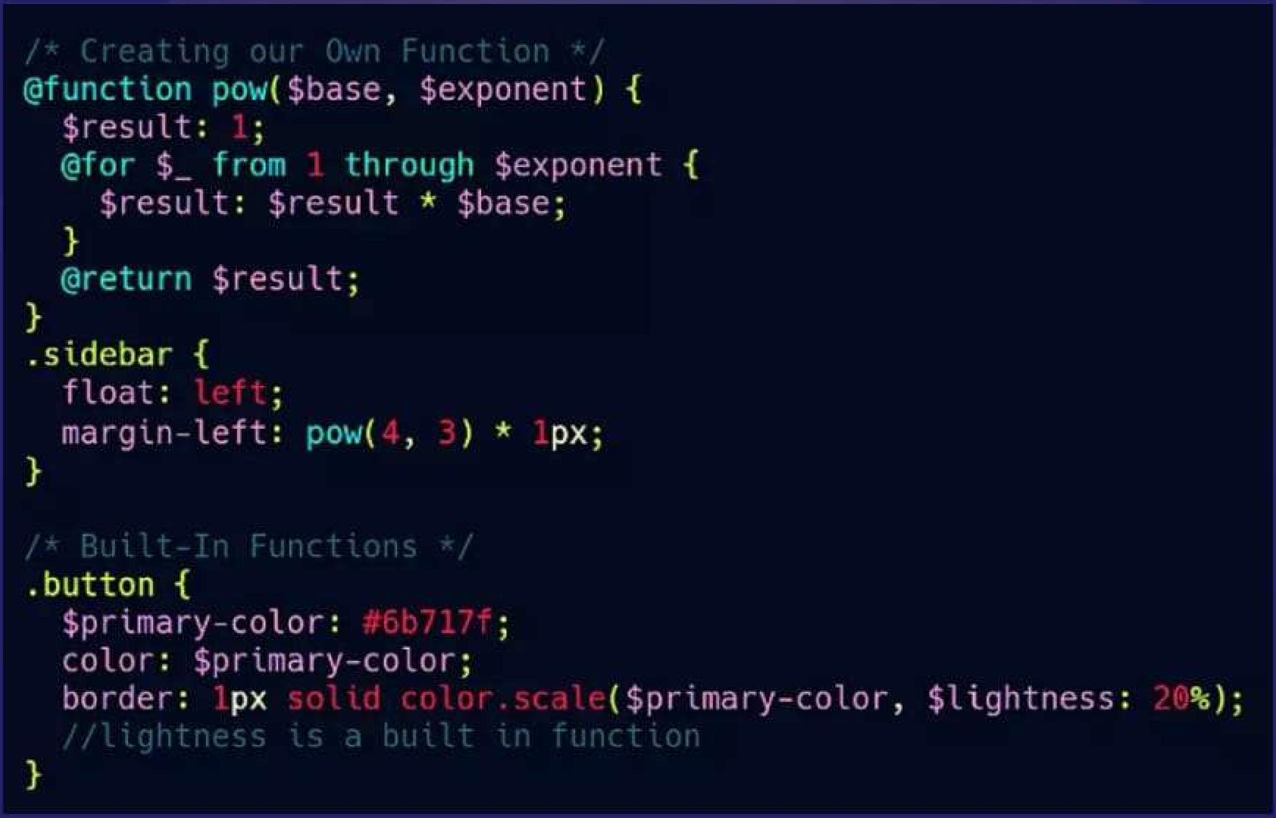
INHERITANCE

Now this is something really useful



```
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}
.message {
  @extend %message-shared;
}
.success {
  @extend %message-shared;
  border-color: green;
}
.error {
  @extend %message-shared;
  border-color: red;
}
.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

FUNCTIONS & FOR LOOP



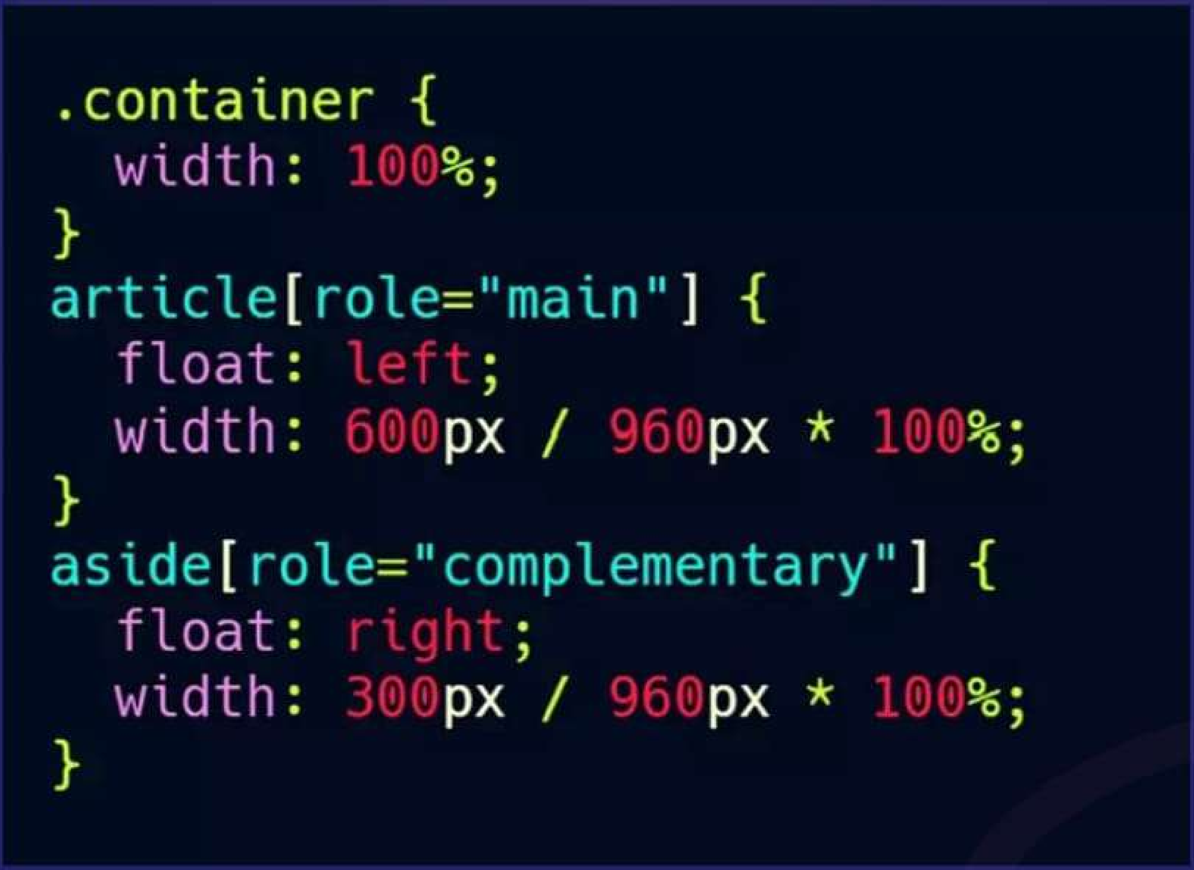
```
/* Creating our Own Function */
@function pow($base, $exponent) {
  $result: 1;
  @for $_ from 1 through $exponent {
    $result: $result * $base;
  }
  @return $result;
}
.sidebar {
  float: left;
  margin-left: pow(4, 3) * 1px;
}

/* Built-In Functions */
.button {
  $primary-color: #6b717f;
  color: $primary-color;
  border: 1px solid color.scale($primary-color, $lightness: 20%);
  //lightness is a built in function
}
```

Similar to lightness, SASS has so many built-in functions which are very useful in large applications.

Similarly to @for, SASS have @each, @while, @if & @else flow controls.

OPERATORS

Doing math in your CSS is very helpful. Sass has a handful of standard math operators like :- +, -, *,/, and %.



```
.container {
  width: 100%;
}
article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}
aside[role="complementary"] {
  float: right;
  width: 300px / 960px * 100%;
}
```