the Master Course

{CODENATION}

Brown Bag

SASS/SCSS

{CODENATION}

Syntactically

Awesome

Style

Sheets

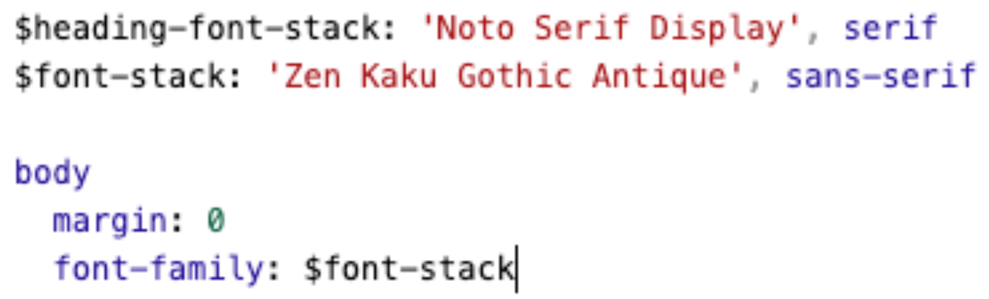
What is SASS?

> CSS Compiler
> **More organised** style sheets
> Lets us use **variables, nested rules** and much more

Then what is SCSS?

> SASS follows strict indentation SCSS does not
> SASS does not use brackets or semicolons whereas SCSS more resembles CSS

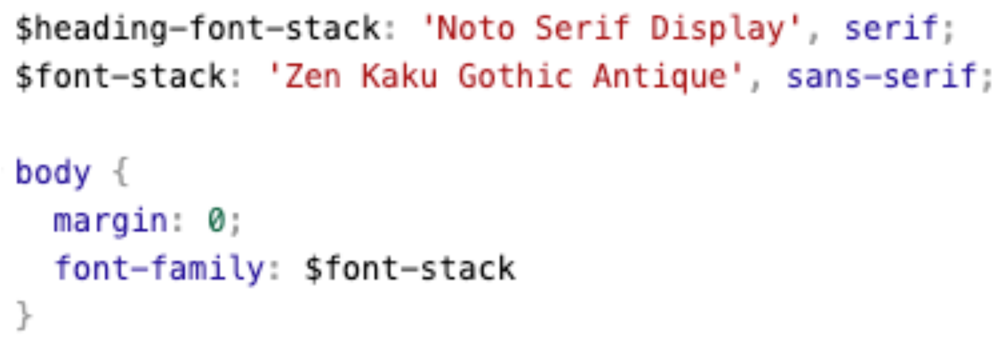
SASS



```sass
$heading-font-stack: 'Noto Serif Display', serif
$font-stack: 'Zen Kaku Gothic Antique', sans-serif

body
  margin: 0
  font-family: $font-stack
```

SCSS



```scss
$heading-font-stack: 'Noto Serif Display', serif;
$font-stack: 'Zen Kaku Gothic Antique', sans-serif;

body {
  margin: 0;
  font-family: $font-stack
}
```

{CODENATION}

How to use SASS

> Install the VS code extension **Live SASS Complier**
> Install SASS in your terminal

```
npm install -g sass
```

> Name files with correct extensions e.g. style.sass style.scss

{ CN }® Install Instructions: https://sass-lang.com/install

Variables

> Just like JS we can **store information** in variables
> Store **any CSS value** you want e.g. colours, fonts etc.
> Use **$name** to create and reference a variable

SASS

```sass
$heading-font-stack: 'Noto Serif Display', serif
$font-stack: 'Zen Kaku Gothic Antique', sans-serif

body
  margin: 0
  font-family: $font-stack
```

SCSS

```scss
$heading-font-stack: 'Noto Serif Display', serif;
$font-stack: 'Zen Kaku Gothic Antique', sans-serif;

body {
  margin: 0;
  font-family: $font-stack
}
```

{CODENATION}

Nesting

> We can **nest selectors** in a way that follows the hierarchy of your HTML
> This makes your CSS **more organised**
> Overly nesting is bad practice as it gets hard to maintain

{CODENATION}

SASS

```sass
section
  h1
    margin: 0
    font-size: 40px
    text-decoration: underline

  h2
    color: gray

  p
    padding: 10px
    font-size: 20px
```

SCSS

```scss
section {
  h1 {
    margin: 0;
    font-size: 40px;
    text-decoration: underline;
  }
  h2 {
    color: gray;
  }
  p {
    padding: 10px;
    font-size: 20px;
  }
}
```

{CODENATION}

Ampersand (&)

> The & **refers to the parent** when nesting selectors

> Allows for quick **DRY** code

{CODENATION}

SASS

```sass
button
  padding: 10px 18px
  background-color: blue
  border-radius: 5px

  &:hover
    background-color: light-blue

  &:active
    background-color: purple
```

SCSS

```scss
button {
  padding: 10px 18px;
  background-color: blue;
  border-radius: 5px;

  &:hover {
    background-color: light-blue;
  }
  &:active {
    background-color: purple;
  }
}
```

{CODENATION}

Mixins

> Mixins are very similar to JS functions
> Allow us to **reuse groups of CSS** declarations
> Keeps your stylesheets very **DRY**
> You can also **pass in values** to make them more flexible e.g. to change colour theme

{CODENATION}

SASS

```sass
$heading-font-stack: 'Noto Serif Display', serif
$font-stack: 'Zen Kaku Gothic Antique', sans-serif

@mixin font($family, $color, $size, $transform)
  color: $color
  font-family: $family
  font-size: $size
  text-transform: $transform

.mainInfo
  padding: 40px

  h1
    margin: 0 0 20px
    @include font($heading-font-stack, black, 50px, uppercase)

  h2
    @include font($heading-font-stack, black, 30px, uppercase)
    margin: 0 0 10px

  p
    @include font($font-stack, #444444, 16px, none)
```

SCSS

```scss
$heading-font-stack: 'Noto Serif Display', serif;
$font-stack: 'Zen Kaku Gothic Antique', sans-serif;

@mixin font($family, $color, $size, $transform) {
  color: $color;
  font-family: $family;
  font-size: $size;
  text-transform: $transform;
}

.mainInfo {
  padding: 40px;

  h1 {
    margin: 0 0 20px;
    @include font($heading-font-stack, black, 50px, uppercase);
  }
  h2 {
    @include font($heading-font-stack, black, 30px, uppercase);
    margin: 0 0 10px;
  }
  p {
    @include font($font-stack, #444444, 16px, none);
  }
}
```

{CODENATION}

Modules

> This lets us **split up our styling** into different files
> You can refer to other files variables, mixins, and functions
> use the **@use 'filename'** you don't need the file extension

SASS

base.sass

```
$font-stack: Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

styles.sass

```
@use 'base'

.inverse
  background-color: base.$primary-color
  color: white
```

SCSS

base.sass

```
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

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

styles.sass

```
@use 'base';

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

{CODENATION}

Extend

> This lets you share a set of properties across multiple selectors

> Helps write **DRY code** quickly

> Lets you **inherit styles** from other selectors

{CODENATION}

SASS

```sass
.border
  border: 2px solid black
  padding: 10px
  margin: 5px

.important
  @extend .border
  border-color: red

.success
  @extend .border
  border-color: green
```

SCSS

```scss
.border {
  border: 2px solid black;
  padding: 10px;
  margin: 5px;
}

.important {
  @extend .border;
  border-color: red;
}

.success {
  @extend .border;
  border-color: green;
}
```

{CODENATION}

More Features

> As with any new language or tool that you use you should **read the documentation**

> https://sass-lang.com/documentation

Now you know **Sassy CSS**