In React → node install node-sass @ 4.14.1
Then import "App.scss";

SASS
Version - 3 (SCSS (Sassy CSS))

Variables

```
$primary-color : #272727;

body {
    background: $primary-color;
}
```

Maps

```
$font-weights: (
    "regular": 400,
    "medium": 500,
    "bold": 700
);

body {
    font-weight: map-get ($font-weights, bold);
}
```

Interpolation #{}

In html

```
<div class="main">
  <p class="main-p">....</p>
</div>
```

In SCSS

```
.main {
  color: grey;
  & paragraph {
    font-weight: bold;
  }
}
```

→ In CSS

```
.main {
  color: grey;
}
.main_paragraph {
  font-weight: bold;
}
```

To make it wrap inside parent in CSS

```
.main {
  #{&}_paragraph {
  }
}
```

→

```
.main {
}
.main .main_paragraph {
}
```

Also & can be used for pseudo elements

```
.main_p {
  &:hover {
    color: red;
  }
}
```

→

```
.main_p:hover {
  color: red;
}
```

Contents can

Modularity

- Seperate files can be made and its content can be imported css

```
@import "./resets", "./variables.scss";
```

Functions

```
@function weight($weight-name) {
  @return map-get($font-weights, $weight-name)
}

.main_p {
  font-weight: weight(regular);
}
```

Mixin : To impart certain properties.

```
@mixin theme($light-theme: true) {
  @if $light-theme {
    background: lighten($primary-color, 100%);
    color: darken($text-color, 100%);
  }
}

.light {
  @include theme($light-theme: true);
}
```

Also it is useful with media queries.

```
@mixin mobile {
  @media (max-width: $mobile) {
    @content;
  }
}
```

→ 200px (calibrated by user)

```
.main {
    @include mobile {
        flex-direction: column;
    }
}
```

extend:

Used to inherit all properties.

```
#{&}__paragraph2 {@extend .main__paragraph1;
    font-weight: weight(bold);}
```

Suppose there are 10 properties inherited then we can change the values individually too.

Operations (Add, Sub, multiplication div)

width: 80% - 40%

In normal CSS to do this

calc(80% - 40%)

Advantage of calc → calc(80% - 400px)

different units possible

Loops

```
@for $i from 1 through 4 {
    .menu-nav__item:nth-child(#{$i}) {
        transition-delay: ($i * 0.5) + 0.15s;
    }
}
```

#{$i} → convert numerical $i to string