155-SCSS Media Query Mixins video

- we can think of a SCSS mixins like a javascript function, we need to define it and then later call it, mixins don't return values but instead they set properties
- we use @mixin, this is syntax specific to SCSS
- here we are going to have a demo just so we can understand it, so go to _page-content.scss

```
@mixin funkyBorder {
    border: 10px solid orange;
}

.page-content {
    display: flex;
    height: $page-content-height;
    max-width: $site-max-width;
    margin: 0 auto;
    @media (min-width: 50rem){
        padding: $large-space $space;
    }
}

.page-content__sidebar{
    display: none;
    padding-right: $large-space;
    width: $page-content-sidebar-width;
    @media (min-width: 50rem){
        display: flex;
    }
}

.page-content__main{
    display: flex;
    width: 100%;
    @media (min-width: 50rem){
```

```
        width: $page-content-main-width;
    }
}
```

- we are going to apply this mixin to .page-content__main, we use @include to call out mixin

```
.page-content__main{
    display: flex;
    width: 100%;
    @include funkyBorder();
    @media (min-width: 50rem){
        width: $page-content-main-width;
    }
}
```

- we can use this anywhere we want to apply it, all we have to do is call it.
- now we are going to setup funkyBoarder to take arguments - define arguments we have to use $ right before the name of the argument

```
@mixin funkyBorder ($borderWidth) {
    border: $borderWidth solid orange;
}
```

- and now lets pass in a value when we call it down below

```
@mixin funkyBorder ($borderWidth) {
    border: $borderWidth solid orange;
}

.page-content {
    display: flex;
    height: $page-content-height;
    max-width: $site-max-width;
    margin: 0 auto;
```

```
    @media (min-width: 50rem){
        padding: $large-space $space;
    }
}

.page-content__sidebar{
    display: none;
    padding-right: $large-space;
    width: $page-content-sidebar-width;
    @media (min-width: 50rem){
        display: flex;
    }
}

.page-content__main{
    display: flex;
    width: 100%;
    @include funkyBorder(20px);
    @media (min-width: 50rem){
        width: $page-content-main-width;
    }
}
```

- now lets pass in another argument, lets do a color this time

```
@mixin funkyBorder ($borderWidth, $borderColor) {
    border: $borderWidth solid $borderColor;
}
```

- and now lets pass the value down below

```
.page-content__main{
    display: flex;
    width: 100%;
```

```
    @include funkyBorder(20px, purple);
    @media (min-width: 50rem){
        width: $page-content-main-width;
    }
}
```

- we can also define a default value to our arguments

```
@mixin funkyBorder ($borderWidth: 10px, $borderColor: orange) {
    border: $borderWidth solid $borderColor;
}
```

- now down below when we call it we are going to overwrite our border default value, but we are going to leave the color alone

```
.page-content__main{
    display: flex;
    width: 100%;
    @include funkyBorder(20px);
    @media (min-width: 50rem){
        width: $page-content-main-width;
    }
}
```

- we can also do selectors inside our mixins

```
@mixin funkyBorder ($borderWidth: 10px, $borderColor: orange) {
    border: $borderWidth solid $borderColor;

    *{
        color: red;
    }
}
```

- right here when we call funkyBorder every font color will be red

- we can also create a block when we call our mixin

```
.page-content__main{
    display: flex;
    width: 100%;
    @include funkyBorder(20px) {
        color: green;
    };
    @media (min-width: 50rem){
        width: $page-content-main-width;
    }
}
```

- but the above example wont' work until we include @content where we defined our mixin

```
@mixin funkyBorder ($borderWidth: 10px, $borderColor: orange) {
    border: $borderWidth solid $borderColor;

    *{
        @content;
    }
}
```

- so now the fonts are green
- we are going to create a mixin for media queries

```
@mixin funkyBorder ($borderWidth: 10px, $borderColor: orange) {
    border: $borderWidth solid $borderColor;

    *{
        @content;
    }
}

@mixin funkyBackground($backgroundColor: green){
```

```
    background-color: $backgroundColor;
}

@mixin desktop {
    @media (min-width: 50rem){
        @content;
    }
}

.page-content {
    display: flex;
    height: $page-content-height;
    max-width: $site-max-width;
    margin: 0 auto;
    @include funkyBackground();
    @media (min-width: 50rem){
        padding: $large-space $space;
    }
}

.page-content__sidebar{
    display: none;
    padding-right: $large-space;
    width: $page-content-sidebar-width;

    @media (min-width: 50rem){
        display: flex;
    }
}

.page-content__main{
    display: flex;
```

```
    width: 100%;
    @include funkyBorder(20px) {
        color: green;
    };
    @media (min-width: 50rem){
        width: $page-content-main-width;
    }
}
```

- and now lets call it and include a block where you can define some properties

```
@mixin funkyBorder ($borderWidth: 10px, $borderColor: orange) {
    border: $borderWidth solid $borderColor;

    *{
        @content;
    }
}

@mixin funkyBackground($backgroundColor: green){
    background-color: $backgroundColor;
}

@mixin desktop {
    @media (min-width: 50rem){
        @content;
    }
}

.page-content {
    display: flex;
    height: $page-content-height;
    max-width: $site-max-width;
```

```
    margin: 0 auto;
    @include funkyBackground();
    @media (min-width: 50rem){
        padding: $large-space $space;
    }
}

.page-content__sidebar{
    display: none;
    padding-right: $large-space;
    width: $page-content-sidebar-width;

    @include desktop{
        display: flex;
    }
}

.page-content__main{
    display: flex;
    width: 100%;
    @include funkyBorder(20px) {
        color: green;
    };
    @include desktop{
        width: $page-content-main-width;
    }
}
```

- now instead of defining our mixins in this file lets create a folder and a file to hold our mixins
- so under imports/client/styles create a folder named mixins - and a file named _media-query-mixins.scss
- cut our mixins from our other file and paste them on this one

```
@mixin funkyBorder ($borderWidth: 10px, $borderColor: orange) {
    border: $borderWidth solid $borderColor;
```

```
    *{
        @content;
    }
}

@mixin funkyBackground($backgroundColor: green){
    background-color: $backgroundColor;
}

@mixin desktop {
    @media (min-width: 50rem){
        @content;
    }
}
```

- next lets take care if the importing by going to _main.scss - make sure to import them after variables and before everything else

```
@import './variables';
@import './mixins/media-query-mixins';
@import './base';
@import './components/boxed-view';
@import './components/button';
@import './components/editor';
@import './components/header';
@import './components/page-content';
@import './components/checkbox';
@import './components/item';
```

- now delete the funkyBorder and background examples
- and commit our changes

```
joses-MacBook-Pro:notes mendoza$ git add .
```

```
joses-MacBook-Pro:notes mendoza$ git commit -m "Setup media queries for page content"

joses-MacBook-Pro:notes mendoza$ git push
```

-