

154-Media Queries video

- go to _page-content.scss
- create a media querie - in this example, anything that is 50 or more rem will turn the font color red in the sidebar

```
.page-content {  
    display: flex;  
    height: $page-content-height;  
    max-width: $site-max-width;  
    margin: 0 auto;  
    padding: $large-space $space;  
}  
  
.page-content__sidebar{  
    display: flex;  
    padding-right: $large-space;  
    width: $page-content-sidebar-width;  
}  
  
@media (min-width: 50rem) {  
    .page-content__sidebar * {  
        color: red;  
    }  
}  
  
.page-content__main{  
    display: flex;  
    width: $page-content-main-width;  
}
```

- now if we use max-width: 50rem, that means that everything that 50rems or below will turn the font color to red in the sidebar

```
@media (max-width: 50rem) {
  .page-content__sidebar * {
    color: red;
  }
}
```

- we can also chain media queries together - in this case we can apply the styles we want between 30 and 50rems

```
@media (max-width: 50rem) and (min-width: 30rem) {
  .page-content__sidebar * {
    color: red;
  }
}
```

- in this case we are just going to be using one media querie using min
- our approach is going to be mobile first, so with sidebar we are going to start with not displaying, and once the size increases we will show it

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

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

```

@media (max-width: 50rem) and (min-width: 30rem) {

    .page-content__sidebar * {

        color: red;

    }

}

.page-content__main{

    display: flex;

    width: $page-content-main-width;

}

```

- the approach next is that we are going to nest our media query inside the page-content__sidebar rule

```

.page-content {

    display: flex;

    height: $page-content-height;

    max-width: $site-max-width;

    margin: 0 auto;

    padding: $large-space $space;

}

.page-content__sidebar{

    display: none;

    padding-right: $large-space;

    width: $page-content-sidebar-width;

    @media (min-width: 50rem){

        display: flex;

    }

}

// @media (max-width: 50rem) and (min-width: 30rem) {

//     .page-content__sidebar * {

//         color: red;

```

```
//    }

// }

.page-content__main{

    display: flex;

    width: $page-content-main-width;

}
```

- now in page-content, we dont want padding in mobile view because we don't want unnecessary space - so we will remove padding from that rule, and applied it our nested media querie

```
.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;

    }

}

// @media (max-width: 50rem) and (min-width: 30rem) {

//     .page-content__sidebar * {

//         color: red;

//     }

// }
```

```
//    }

// }

.page-content__main{

    display: flex;

    width: $page-content-main-width;

}
```

- lets modify the page-content__main so it works well mobile

```
.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;
```

```
}
```

```
}
```

-