

Placeholders

In this lesson, we'll be looking at SASS placeholders.

WE'LL COVER THE FOLLOWING ^

- For Example:
- Why use Placeholders?

For Example:

In SASS a **placeholder** looks and acts a lot like a class selector, only it starts with a % and it's not included in the CSS output.

Our %placeholder selector contains some width and height declarations:

```
%placeholder {  
  width: 100%;  
  height: 100%;  
}  
body {  
  @extend %placeholder;  
}  
p {  
  @extend %placeholder;  
}
```

Note that we've used the @extend directive, which (as we saw previously) allows one selector to inherit styles of another selector.

This outputs to CSS as follows:




```
body, p {  
  width: 100%;  
  height: 100%;  
}
```

Simple and as expected!

However, the preprocessor will skip `%placeholder` and it won't be included in the final CSS file.

Output
HTML
CSS (SCSS)

```
%placeholder {  
  width: 100%;  
  height: 100%;  
}  
body {  
  @extend %placeholder;  
}  
p {  
  @extend %placeholder;  
}
```



Why use Placeholders?

Placeholder selectors are mainly useful when writing a SASS library where each style rule is optional.

Typically, when working on your own project, it's often better to just extend a class selector instead. But it's good to know as it could come in quite handy if you start working on larger-scale projects.

Next, we'll sum up the features of SASS, before moving on to learn how to structure our projects!