Interpolation

In this lesson, we'll learn how to use Interpolation to make our code more dynamic.

WE'LL COVER THE FOLLOWING

Definition

Interpolation is essentially a code insertion. It allows us to interpolate SASS expressions into our code. We can use it to use a selector or property name, quoted or unquoted strings etc, as variables.

Syntax

To interpolate an expression we need to wrap the expression using `#{ }`.

```
#{$variable_name}
```

Let's see an example that shows how we could use interpolation with a mixin:

```
@mixin interpolation($editable, $val, $val2, $prop1, $prop2)
{
    background-#{$editable}: $val;
    position: $val2;
    #{$prop1}: 0px;
    #{$prop2}: 0px;
}


.block1{
    @include interpolation("image", url("img.png"), absolute, top, right);
}
```

```
.block2{
    @include interpolation("color", lightgray, absolute, top, left);
}
```

This will compile in CSS as follows:

```
.block1 {
    background-image: url("img.png");
    position: absolute;
    top: 0px;
    right: 0px;
}

.block2 {
   background-color: lightgray;
   position: absolute;
   top: 0px;
   left: 0px;
}
```

As you can see, it's quite easy to use this to create dynamically reusable code!

Main reasons to use Interpolation

- We can use dynamically created names as a property name, a variable name or for other similar purposes.
- We can create highly reusable code!

Next, we'll learn how to use placeholders in SASS.