

# Mixins

In this lesson, we'll be looking at the SASS mixin syntax.

## WE'LL COVER THE FOLLOWING ^

- Definition
- Example

## Definition #

Another powerful feature of Sass are **mixins**. Using mixins, you can group together multiple CSS declarations for reuse throughout your project.

## Example #

Say we want to create a mixin to hold the vendor prefixes for a transform property.

in SASS, we'd code it like so:

```
@mixin transform {  
  -webkit-transform: rotate(180deg);  
  -ms-transform: rotate(180deg);  
  transform: rotate(180deg);  
}
```

To add the mixin into our code, we then use the `@include` directive, like so:

```
.navbar {  
  background-color: orangered;  
  padding: 1rem;  
  ul {  
    list-style: none;  
  }  
  li {
```

```
text-align: center;
margin: 1rem;
@include transform;
}
}
```

*Note:* Don't worry too much about where to put your mixin code for now. This will be explained in the next chapter. For now, just take note of the general idea!

All the code in the transform mixin will now be applied to the `li` element. You can also pass values into your mixins to make them even more flexible.

Instead of adding a specified value, add a name (using a variable, like `property`) to represent the value like so:

```
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}
```

Now we can pass in whatever value we like, whenever we call the mixin:

```
@include transform (rotate(20deg));
```

In our next lesson, we'll take a look at the function syntax.