



Mixin and Include in SCSS

The syntax for variable declaration is

- Mixin is a concept in SCSS that CSS code can be reused throughout the code.
- Mixin is a keyword that is used to declare the reusability code
- Include is a keyword that is used to call the Mixin code.

```
@mixin name_of_mixin
{
    ...
    css code
    ...
}

-----

element
{
    @include name_of_mixin
    ...
}
```



Example:

```
1
2 <div class='dark_mode'>
3   <h2>Hai</h2>
4 </div>
5 <div class = 'lite_mode'>
6   <h2>Hello</h2>
7 </div>
8 <div class = 'dark_mode' >
9   <h2>Thanks</h2>
10 </div>
11
```

```
1 @mixin lite_mode
2 {
3   color:#000000;
4   background-color:#E2DFD2;
5 }
6 @mixin dark_mode
7 {
8   color:#ffffff;
9   background-color:#000000;
10 }
11 .dark_mode{
12   @include dark_mode
13 }
14 .lite_mode{
15   @include lite_mode
16 }
```

Hai

Hello

Thanks



Arguments in mixin

We can also pass arguments to mixin. If you define values for arguments it is called optional argument. Meaning if the value is not passed from include it can take the default value. Let's see all this by this example

```
@mixin( $arg1,$arg2,$arg3:default_value)
```

```
1
2 ▾ <div class='dark_mode'>
3 ▾   <h2>Hai</h2>
4   </div>
5 ▾ <div class = 'lite_mode'>
6 ▾   <h2>Hello</h2>
7   </div>
8 ▾ <div class = 'dark_mode' >
9 ▾   <h2>Thanks</h2>
0   </div>
1
```

```
1 @mixin set_mode($text_color,$bg_color,$align:left)
2 ▾ {
3   background-color:$bg_color;
4   color:$text-color;
5   text-align:$align;
6 }
7 ▾ .dark_mode{
8   @include set_mode(#ffffff,#000000)
9 }
10 ▾ .lite_mode{
11   @include set_mode(#000000,#E2DFD2,center)
12 }
13
14
15
```



Hai

Hello

Thanks

- From output you can understand the difference between normal arguments and optional arguments.
- Normal argument, we have to pass value. It is compulsory.
- In default argument, it is optional if we did not pass it will take default value. So default value is compulsory for optional argument
- In our case for dark mode we did not passed any alignment details, so it takes default left alignment.
- But in lite mode we passed center as argument value so it takes that value.

Now you have one logical question. Shall I have to pass argument in same order as declared in mixin.

Answer is Yes.



- Then difficulty arises now. Think you have 10 to 20 arguments in mixin. How can I remember the exact order?
- Shall I have to refer declaration each time and pass in that order?



Keywords arguments

Answer is Big No. 😊

We can pass arguments in any order by using the concept keyword argument.

@include(\$arg_name : value)

```
1
2 ▾ <div class='dark_mode'>
3 ▾   <h2>Hai</h2>
4 ▾ </div>
5 ▾ <div class = 'lite_mode'>
6 ▾   <h2>Hello</h2>
7 ▾ </div>
8 ▾ <div class = 'dark_mode' >
9 ▾   <h2>Thanks</h2>
10 ▾ </div>
11
```

```
1 @mixin set_mode($text_color,$bg_color,$align:left)
2 ▾ {
3   background-color:$bg_color;
4   color:$text_color;
5   text-align:$align;
6 ▾ }
7 ▾ .dark_mode{
8   @include
9   set_mode($bg_color:#000000,$text_color:#ffffff)
10 ▾ }
11 ▾ .lite_mode{
12   @include set_mode(#000000,#E2DFD2,center)
13 ▾ }
```

- Here in highlighted area you can see I passed bg_color in first and text_color in second, instead of vice versa order.
- But it produce the same output. How?
- Because we passed argument in front. So it map to correct One