Mixins in Sass

Mixins allow you to define reusable blocks of CSS styles that can be included or "mixed in" to other CSS rules or selectors. They help in maintaining and organizing your styles by promoting reusability and modularity.

Benefits of mixins are:

- Reusability: You can define a set of CSS properties and values once in a mixin and then include that mixin wherever you need those styles. This reduces redundancy in your code.
- Modularity: Mixins make managing and updating styles easier because changes made to a mixin automatically apply to all the places where it's used.
- Parameters: Mixins can accept parameters, allowing you to customize the included styles for specific elements or situations. This makes them more flexible and versatile.
- Organization: They help keep your codebase organized by grouping related styles together.

Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

Define a Mixin

- To define a mixin, use the **`@mixin`** directive followed by a name for your mixin. You can also include parameters if your mixin needs to accept arguments.

Syntax @mixin

```
@mixin name {
  property: value;
  property: value;
}
```

Example:

```
// Define a mixin without parameters
@mixin myMixin {
    // CSS styles go here
}

// Define a mixin with parameters

@mixin buttonStyles($background, $color) {
    background-color: $background;
    color: $color;
}
```

Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

Use a Mixin

- To use a mixin within a CSS rule, you use the **@include** directive followed by the mixin name. If the mixin accepts parameters, you provide values for those parameters.

Syntax @include

```
Class name {
  @include: mixin name;
}
```

Example:

```
// Using a mixin without parameters
.my-element {
  @include myMixin;
}

// Using a mixin with parameters
.button {
  @include buttonStyles(#3498db, #fff);
}
```

- In this example, the **myMixin** mixin is included in the **.my-element** selector without any parameters. The **buttonStyles** mixin, on the other hand, is included in the **.button** selector with values for the **$background** and **$color** parameters.

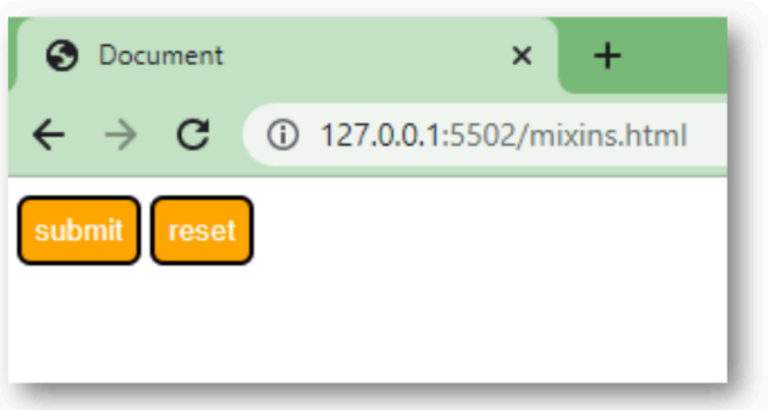
Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

Mixins Example

```html
<body>
  <button class="submit">submit</button>
  <button class="reset">reset</button>
</body>
```

Index.html



Output

```scss
@mixin PrimaryBtn {
  background-color: orange;
  border-radius: 6px;
  border: 2px solid black;
  text-align: center;
  padding: 6px;
  color: white;
}

.submit {
  @include PrimaryBtn();
}
.reset {
  @include PrimaryBtn();
}
```

Index.scss

```css
.submit {
  background-color: orange;
  border-radius: 6px;
  border: 2px solid black;
  text-align: center;
  padding: 6px;
  color: white;
}

.reset {
  background-color: orange;
  border-radius: 6px;
  border: 2px solid black;
  text-align: center;
  padding: 6px;
  color: white;
}
```

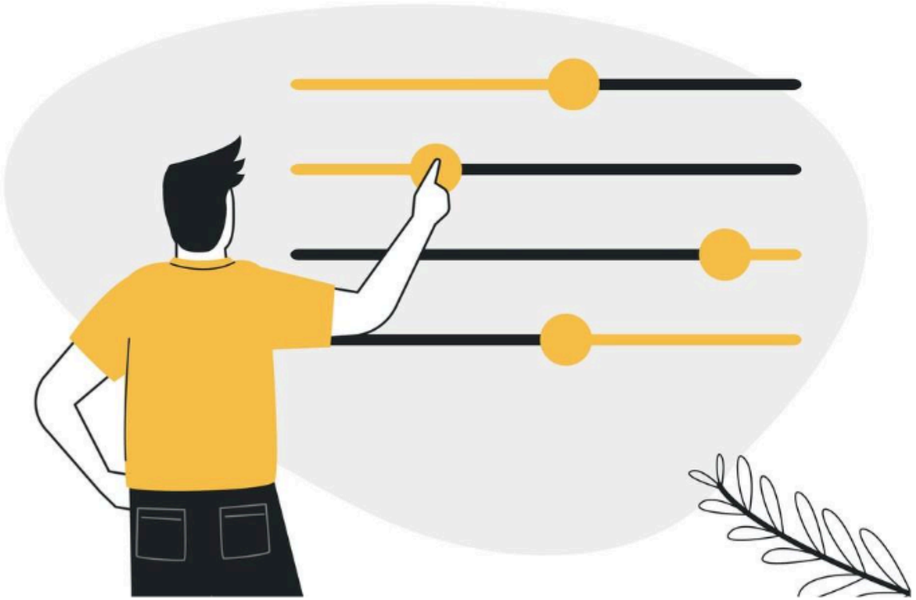
Index.css

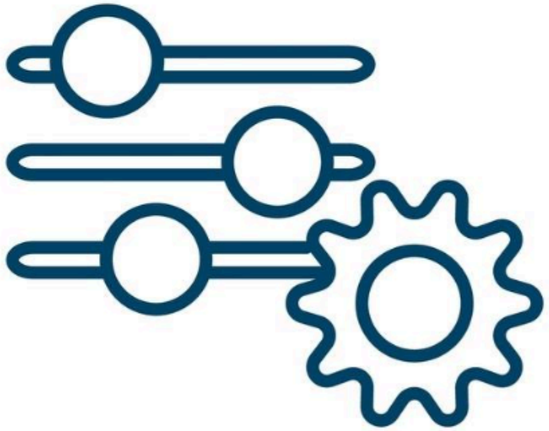
Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

Parameters in Sass

Parameters in Sass enhance code flexibility and reusability. With dynamic mixins and functions, they adapt to various inputs, making stylesheets more versatile and easier to maintain.





PARAMETER

Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

Parameters in Sass (contd.)

- You can also pass parameters to mixin. The parameter is a variable of sass.
- Mixins can act as a function and take parameters.
- We can pass CSS variables as parameters to use them inside the mixin.
- Syntax of passing parameters to mixins is as shown below:

```
@mixin name($arg1,
$arg2,..) {
  property: $arg1;
  property: $arg2;
  ...
}
```

Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

Parameters in Sass (contd.)

In the following example we are passing argument $width,$color,$type,$fontSize. These values of the variables are assigned to properties of elements (p-1, p-2, p-3, p-4).

```
html > s.scss > ...
1   // define variables for the primary colors
2
3   @mixin common_rule($width,$color, $type, $fontSize) {
4     border: $width $color $type ;
5     font-size: $fontSize;
6
7   }
8   #p-1{
9     @include common_rule(2px, yellow,solid, 10px)
10  }
11  #p-2{
12    @include common_rule(5px, rgb(23, 37, 168), solid,15px)
13  }
14  #p-3{
15    @include common_rule(4px, rgb(93, 172, 47),solid, 10px)
16  }
17  #p-4{
18    @include common_rule(3px, rgb(202, 117, 37), solid,10px)
19  }
20
```

Parameters Code Example

Activate Windows
Go to Settings to activate Windows.

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.