

SASS, Les directives

Mixins, Extend, les boucles, etc... sont faites pour rendre le code plus réutilisable.

Les mixins

pro: évite la répétition con: duplique le code généré.

Les mixins permettent de créer des blocs de code CSS réutilisables dans différents endroits grâce à `@mixins` et `@include`

Exemple sans arguments: (dans ce cas précis, il est peut être préférable d'utiliser un placeholder)

```
@mixin responsive-img {  
  display: block;  
  width: 100%;  
  height: auto;  
}
```

```
.image {  
  @include responsive-img;  
}
```

Exemple avec arguments

```
@mixin border-radius( $radius ) {  
  border-radius: $radius;  
}
```

```
.box {  
  @include border-radius(10px);  
}
```

La directive @extend

Elle permet d'hériter des styles d'un autre sélecteur, rendant le code plus DRY (Don't Repeat Yourself).

pro: Réduit la taille des fichiers CSS con: Ne fonctionne qu'avec les sélecteurs existants.

Exemple:

```
.button-basic {  
  padding: 10px;  
  font-size: 14px;
```

```
}

.button-primary {
  @extend .button-basic;
  background-color: blue;
}

.button-secondary {
  @extend .button-basic;
  background-color: gray;
}
```

Les boucles @for

Elles permettent de générer du code répétitif en boucle, utile pour des classes ou des grilles.

```
@for $i from 1 through 5 {
  .box-#{ $i } {
    width: #{20 * $i}px;
    height: #{20 * $i}px;
    background-color: lighten(blue, $i * 5%);
  }
}

.box-1 {
  width: 20px;
  height: 20px;
  background-color: #0000ff;
}

.box-2 {
  width: 40px;
  height: 40px;
  background-color: #3333ff;
}
etc..
```

Conclusion

@mixins : Pour réutiliser facilement des blocs CSS. @extend : Pour partager des styles entre sélecteurs sans duplication. @for : Pour générer dynamiquement du code CSS. Ces outils rendent SASS puissant et indispensable pour des projets CSS complexes.