

# CSS- 09- f- Mixins

Les mixins permettent de créer des extraits de CSS pouvant être réutilisés à différents endroits grâce à @include.

Par exemple, il est possible de se créer une mixin rendant une image responsive.

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

Plutôt que de répéter ces trois lignes de code CSS à chaque fois à chaque image doit être responsive, il serait possible d'appeler simplement cette mixin.

Par exemple:

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

- [mixin](#)

## Arguments

Les mixins acceptent de recevoir des arguments. Ces derniers sont ensuite disponibles dans le corps de la mixin, rendant ainsi leur usage plus flexible.

Par exemple, afin d'ajoutez un max-width variant selon différents scénarios, il n'est pas nécessaire de créer plusieurs mixins. L'ajout d'un argument permettra de réutiliser la même.

```
@mixin responsive-img($max) {  
  display: block;  
  width: 100%;  
  height: auto;  
  max-width: $max; // 600px  
}  
  
.article-img {  
  @include responsive-img(600px);  
}
```

Si un argument sensé être fournis à un mixin est omis, une erreur se produira au moment de la compilation du code CSS.

## Valeur par défaut

Afin d'éviter une erreur lorsqu'un argument est omis, il est possible de fournir une valeur par défaut.

Par exemple:

```
@mixin responsive-img($max: 100%) {  
  display: block;  
  width: 100%;  
  height: auto;  
  max-width: $max; // 100%  
}  
  
.article-img {  
  @include responsive-img;  
}
```

## Spécifier l'argument

Les arguments sont généralement passés à une mixin selon l'ordre dans lequel ils ont été déclarés dans celle-ci. Cependant, il est possible de spécifier à quel argument une valeur est passée. Ceci permet, d'omettre certains arguments ou simplement de les écrire dans un ordre différent.

Par exemple:

```
@mixin responsive-img($display: block, $max: 100%) {  
  display: $display;  
  width: 100%;  
  height: auto;  
  max-width: $max; // 300px  
}  
  
.article-img {  
  @include responsive-img($max: 300px);  
}
```

Ainsi, le fait de nommer \$max dans l'appel de la mixin permet de spécifier que la valeur 300px n'est pas pour \$display, mais bien \$max, même si ce dernier est le 2<sup>e</sup> argument.

- [arguments](#)