

CSS 09- d- Extend

La règle `@extend` permet de définir une classe ayant les mêmes règles de style qu'une autre classe spécifiquement.

Par exemple, une classe de message pourrait ressembler à ceci:

```
.msg {  
  border-radius: 10px;  
  padding: 10px;  
  background-color: blue;  
}
```

Si d'autres types de messages doivent ensuite être créés et qu'ils doivent partager les mêmes styles de bases que `.msg`, il serait possible d'étendre cette classe plutôt que de réécrire à chaque fois le même code.

Par exemple:

```
.msg-error {  
  @extend .msg;  
  background-color: red;  
}  
  
.msg-success {  
  @extend .msg;  
  background-color: green;  
}
```

Ce qui générerait le code suivant:

```
.msg, .msg-success, .msg-error {  
  border-radius: 10px;  
  padding: 10px;  
  background-color: blue;  
}  
  
.msg-error {  
  background-color: red;  
}  
  
.msg-success {  
  background-color: green;  
}
```

Les trois messages partageraient donc le même border-radius et le même padding tandis que la couleur de fond serait écrasée pour chaque variation.

[@extend](#)

Placeholder

Parfois il est pratique de définir des règles qui peuvent être étendues sans pour autant que ces règles existent en soi, d'où l'utilité des %placeholder. Ces règles commencent avec un % et ne génèrent aucun code à moins d'être étendues.

Par exemple, l'exemple précédent pourrait être réécrit avec un placeholder afin d'éviter que la couleur de fond bleue soit écrasée par les variantes de messages:

```
%msg {  
  border-radius: 10px;  
  padding: 10px;  
}  
  
.msg {  
  @extend %msg;  
  background-color: blue;  
}  
  
.msg-error {  
  @extend %msg;  
  background-color: red;  
}  
  
.msg-success {  
  @extend %msg;  
  background-color: green;  
}
```

Ce qui générerait le code suivant:

```
.msg-success, .msg-error, .msg {  
  border-radius: 10px;  
  padding: 10px;  
}  
  
.msg { background-color: blue; }  
.msg-error { background-color: red; }  
.msg-success { background-color: green; }
```

Les placeholders sont particulièrement utile en ce qui à trait aux règles de styles que nous utilisons régulièrement.

Par exemple, je sais que je centre régulièrement des éléments via transform. Plutôt que de taper à chaque fois:

```
.element {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```

Je me crée donc régulièrement un placeholder similaire à ceci:

```
%centered {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```

Et je l'utilise sur mes éléments que je désire centrer ainsi:

```
.element {  
  @extend %centered;  
}
```

[placeholder](#)

Limitations

Une règle de style ne peut être étendue que si elle se trouve dans le même contexte que sa définition. Autrement dit, une règle de style définie à la base d'un fichier ne peut pas être étendue à l'intérieur d'une media query à moins d'être défini à nouveau dans ce contexte.