

# CSS 09 – b- Imbrication

Par défaut le langage CSS ne permet pas de faire d'imbrications, communément appelé "*nesting*" en anglais. Il est donc parfois nécessaire de répéter un sélecteur plusieurs fois dans son code CSS.

Heureusement, Sass apporte cette fonctionnalité à CSS, ce qui permet d'accélérer l'écriture de feuilles de styles et de simplifier leur lecture.

## Imbrications de base

Afin d'éviter de répéter le sélecteur `.list` à plusieurs reprises, il est possible d'écrire en SCSS:

```
.list {  
  position: relative;  
  
  .item { display: inline-block; }  
  
  .link { color: blue; }  
}
```

Ce qui générera le code CSS suivant :

```
.list { position: relative; }  
  
.list .item { display: inline-block; }  
  
.list .link { color: blue; }
```

## Niveaux d'imbrications

Vous pouvez imbriquer des éléments ainsi à l'infini. Dans l'exemple précédent, nous avons un seul niveau d'imbrication.

Voici maintenant un exemple contenant deux niveaux d'imbrications :

```
.list {  
  position: relative;  
  
  .item {  
    display: inline-block;  
  
    .link { color: blue; }  
  }  
}
```

Ce qui générera le code CSS suivant :

```
list { position: relative; }  
  
.list .item { display: inline-block; }  
  
.list .item .link { color: blue; } // ici
```

Évitez d'abuser des imbrications. Plus de trois niveaux d'imbrication est généralement considéré comme étant une mauvaise pratique et rendra votre code difficile à déboguer.

[nesting](#)

## Sélecteurs avancés

Les imbrications sont compatibles avec tous les [sélecteurs CSS avancés](#).

Par exemple:

```
.list {  
  > .item { display: inline-block; }  
  
  + .logo { display: none; }  
}
```

Générera le code CSS suivant

```
.list > .item { display: inline-block; }  
  
.list + .logo { display: none; }
```

[selector combinator](#)

## Sélecteur de parent

Le sélecteur de parent & est un sélecteur spécial inventé par Sass permettant de faire référence au sélecteur parent courant.

Tout sélecteur imbriqué dans un autre se fait convertir par défaut en enfant du premier.

Autrement dit:

```
.selecteur1 {  
  .selecteur2 { ... }  
}
```

Générera le code CSS suivant :

```
.selecteur1 .selecteur2 { ... }
```

Remarquez **l'espace entre les deux sélecteurs** indiquant que .selecteur2 est enfant de .selecteur1.

Cependant grâce au sélecteur parent il est possible de contourner ce comportement. Voyons quelques usages de ce sélecteur.

## Pseudo-classes

Les imbrications Sass sont compatibles avec les [pseudo-classes](#) lorsqu'un sélecteur parent est utilisé.

Par exemple :

```
.btn {  
  background: blue;  
  
  &:hover { background: lightblue; }  
}
```

Générera le code CSS suivant :

```
.btn { background: blue; }  
  
.btn:hover { background: lightblue; }
```

Remarquez l'utilisation du sélecteur parent &. Celui-ci permet de faire une référence au sélecteur courant, en l'occurrence .btn et de lui rabouter directement, **sans espace**, la pseudo-classe :hover.

## Pseudo-éléments

Les imbrications Sass sont compatibles avec les [pseudo-éléments](#) lorsqu'un sélecteur parent est utilisé.

Par exemple:

```
.btn {  
  background: blue;  
  
  &::before { content: "🔴"; }  
}
```

Générera le code CSS suivant 

```
.btn { background: blue; }
```

```
.btn::before { content: "❌"; }
```

## Renversé

Parfois, il est souhaitable de spécifier que notre élément devrait avoir une apparence différente dans un certain contexte.

Par exemple, lorsqu'un bouton est affiché dans un menu, sa couleur de fond pourrait devoir être grise plutôt que bleue. Le sélecteur parent permet de couvrir ce cas de figure sans avoir à quitter le contexte du sélecteur du bouton en utilisant ce sélecteur comme suffix.

```
.btn {  
  .menu & { background: grey; }  
}
```

Générera le code CSS suivant :

```
.menu .btn { background: grey; }
```