

# TP5: compléments sur la propagation des styles CSS

Les propriétés CSS se voient appliquées au document HTML suivant 3 principes : la mise en **cascade**, l'**héritage**, et la notion de **spécificité**.

## 1 Styles en cascade

Ce terme désigne simplement le fait que en cas de spécifications contradictoires, c'est celle qui est définie en **dernier** qui "gagne".

Par exemple, avec les styles suivants :

```
h2 { color:blue;}
h2 { color:red;}
```

Alors le code suivant apparaîtra en rouge.

```
<h2>Titre</h2>
```

On ne rencontrera en pratique jamais ce cas de façon directe, mais une page web fait fréquemment appel à plusieurs feuilles de style :

```
<head>
  <link rel="stylesheet" href="styles1.css" type="text/css">
  <link rel="stylesheet" href="styles2.css" type="text/css">
</head>
```

Si ces fichiers contiennent des propriétés contradictoires pour le même sélecteur, alors c'est celle qui est en dernier qui s'applique.

## 2 Héritage

Ce terme désigne le fait qu'un élément inséré dans un autre va "récupérer" les propriétés de l'élément parent. Par exemple, avec les styles suivants :

```
body { color: red; }
p { color: blue; }
span { font-weight:bold; }
```

et le code HTML :

```
<p>Ceci est du texte <span>en gras</span>.</p>
```

Alors les mots "en gras" apparaîtront en bleu : le contenu de la balise **span** hérite de la couleur bleue attribuée à l'élément **p**.

## 3 Spécificité

Dans le cas d'une partie de code qui pourrait "matcher" plusieurs sélecteurs, alors le navigateur va appliquer le style du sélecteur le plus spécifique. Par exemple, soit le style suivant :

```
p { color: blue; }
p.c1 { color: red; }
```

Avec cette spécification, tous les paragraphes (balise **p**) seront bleu, **sauf** ceux de la classe **c1**. Ceci sera **indépendant** de l'ordre dans lequel on donne ces spécifications : on aurait le même résultat avec :

```
p.c1 { color: red; }
p { color: blue; }
```

```
<p>Ceci sera bleu</p>
<p class="c1">Ceci sera rouge</p>
```

Les sélecteurs sont évalués de la façon suivante, du moins spécifique au plus spécifique :

	Type de sélecteur	Exemple de sélecteur
1	sélecteur universel	<b>*</b>
2	balise	<b>p</b>
3	classes et pseudo-classes	<b>p.c1, a:visited</b>
4	id	<b>p#id1</b>
5	style défini localement	

En cas d'égalité de niveau de spécificité, alors c'est la règle de "cascade" qui s'applique (le dernier "gagne").

Attention, les sélecteurs descendants amènent un degré de spécification supplémentaire. Par exemple, avec les styles suivants, et indépendamment de l'ordre dans lequel ces deux styles sont donnés :

```
div p { color: red; }
p { color: blue; }
```

On aura :

```
<p>Ceci sera bleu</p>
<div>
  <p>Ceci sera rouge</p>
</div>
```

Autre exemple, avec les styles suivants :

```
p.c1 { color: red; }
p#id1 { color: yellow; }
p { color: blue; }
p span { color: green; }
```

Alors on aura, indépendamment de l'ordre dans lequel ces styles sont donnés :

```
<p>Ceci sera bleu</p>
<p class="c1">Ceci sera rouge</p>
<p id="id1">Ceci sera jaune</p>
<p>Ceci <span>sera vert</span></p>
<p style="color:black;">Ceci sera noir</p>
```

## 4 Classes multiples

Il est possible d'avoir des balises dotées de plusieurs classes. Par exemple

```
<p class="c1 c2">texte</p>
```

Ici, le texte va se voir attribuer à la fois les styles du sélecteur **p.c1** et du sélecteur **p.c2**.

L'intérêt est de pouvoir factoriser les styles dans des spécifications séparées qui pourront être réutilisées dans des contextes différents. Par exemple, on pourra avoir une classe qui concerne le style du texte (police, couleur, etc.) et une classe qui concerne les bordures.

```
p.texte1 { font-family:Helvetica; }
p.texte2 { font-family:Times; }
p.bord1 { border: 1px solid black; }
p.bord2 { border: 2px solid red; }
```

```
<p class="texte2 bord2">un texte en Times, bord rouge</p>
<p class="texte2 bord1">un texte en Times, bord noir</p>
<p class="texte1">un texte en Helvetica</p>
```