



Franck CELLIER

Formateur en Informatique

Technicien Qualité d'Organisme de Formation

<http://cellierfranck.alwaysdata.net>



MS LANGAGE & DEVELOPPEMENT POUR LE WEB

L'HÉRITAGE DANS LE CSS

A. LE STYLE D'UN ÉLÉMENT HTML PEUT ÊTRE HÉRITÉ D'UN ÉLÉMENT "PARENT".

Certains styles, tels que le type de caractère, sa couleur, l'alignement du texte, etc., se transmettent automatiquement des blocs HTML "parents" vers leurs "enfants". D'autres styles tels que la bordure, et les marges ([margin](#) et [padding](#)) ne se transmettent pas.

Exemple :

```
<div style="font-family:Times; padding-left:30px;">
```

Cette première phrase sera écrite en Times.

```
<p>
```

Ce paragraphe sera également écrit en Times car sa propriété "font-family" sera héritée de son parent <div>. Par contre, ce paragraphe ne sera pas décalé sur la droite par rapport à la première phrase, car la propriété padding ne se transmet pas par héritage.

```
</p>
```

```
</div>
```

Dans cet exemple, le bloc <div> est **parent**, du paragraphe <p> (il le contient). On pourra également dire que <p> est **enfant** de <div> (il est contenu par <div>)

En théorie, il est possible de "forcer" l'héritage d'une propriété non-héritable en précisant, par exemple : <p style="padding:inherit" >. Malheureusement, Internet Explorer ne supporte pas l'attribut **inherit**, même dans sa version 7. Nous devons donc nous passer de cette possibilité et préciser explicitement les propriétés non-héritables que nous souhaitons appliquer aux éléments enfants :

Exemple :

```
<div style="font-family:Times; padding-left:30px;">
```

Cette première phrase sera écrite en Times.

```
<p style="padding-left:30px" >
Ce paragraphe sera également écrit en Times car sa propriété "font-family" sera héritée de son parent
</div>. Ce paragraphe sera également décalé sur la droite par rapport à la première phrase, car nous
avons répété le "padding" qui ne se transmet pas par héritage. </p>

</div>
```

B. DES STYLES DÉFINIS EN DERNIER REMPLACENT LES STYLES DÉFINIS AUPARAVANT.

Si vous définissez une propriété de style et que vous redéfinissez plus tard la même propriété pour le même type d'objet, c'est la dernière définition qui sera prise en compte par les navigateurs.

Exemple :

CSS

Cette définition de style est à placer dans la section <head> de votre page ou dans une feuille de styles séparée.

```
p {
  color:green;
}
```

HTML

```
<p>Ce texte sera en vert.</p>
```

```
<p style="color:blue" >Ce texte sera en bleu parce que la propriété "color:blue" est la dernière à avoir
été définie pour ce paragraphe.</p>
```

Cette règle du "dernier-gagnant" s'applique aussi dans le cas où vous avez une combinaison de feuilles de styles séparées + définitions de style dans l'en-tête de la page (<head>).

C. LES STYLES SE COMBINENT ENTRE EUX, QUELLE QUE SOIT LA FAÇON DONT ON LES A DÉFINIS.

Les styles qui s'appliquent à un élément peuvent être une combinaison de n'importe quelles listes de définitions :

- Une ou plusieurs feuilles de styles externes,
- Les définitions placées dans l'en-tête de la page,
- Les définitions placées directement dans la balise de l'élément HTML,
- Les définitions héritées des blocs-parents.

Voici un exemple complexe combinant ces possibilités :

Document "mes-styles.css"

```
.ma-classe {  
  border:2px dashed red;  
}  
#mon-ID {  
  font-size:large;  
  color:white;  
}  
body {  
  font-weight:bold;  
}
```

Document "ma-page.html"

```
<head>  
<link href="mes-styles.css" media="screen" rel="stylesheet" type="text/css" />  
<style type="text/css">  
  .ma-classe {  
    background-color:orange;  
  }  
</style>  
</head>  
  
<body>  
<div class="ma-classe" id="mon-ID" style="font-style:italic;">
```

Exemple de texte.

```
</div>  
</body>
```

Dans cet exemple, la classe "**ma-classe**" s'est vu attribuer un cadre rouge dans la feuille de style séparée et un fond orange dans l'en-tête de la page. Le bloc **<div>** comportant le texte d'exemple récupère tous les styles de la classe "**ma-classe**", plus les styles définis dans "**mon-ID**" auxquels on ajoute le style "**italic**" défini directement dans la balise.

De plus, ce paragraphe hérite du style **font-weight:bold** qui a été attribué à son parent **<body>**.

D. UNE DÉFINITION DE STYLE PEUT PRÉCISER LA FAÇON DONT LE STYLE S'APPLIQUE : À QUELS ÉLÉMENTS, À QUELLES CLASSES, À quels ID, dans quel contexte.

Pour préciser les conditions dans lesquelles un style va s'appliquer, vous avez plusieurs possibilités :

- Préciser le type d'élément HTML
- Préciser la classe
- Préciser l'ID
- Préciser le "*type modifier*" (ou *pseudo-classe*)

Dans un premier temps nous avons examiné plusieurs exemples qui démontrent comment définir un style pour un type d'élément (tel que `<p>`, par exemple), pour une classe ou pour un ID.

Il nous reste donc à aborder la notion de "*type modifier*".

Le principe du "*type modifier*" et de préciser dans quel contexte un style va s'appliquer ou non à un élément HTML. Les "*type modifiers*" sont très peu nombreux. Nous nous contenterons d'examiner ceux qui sont reconnus par tous les navigateurs. Ils s'appliquent tous les quatre à la balise `<a>` qui sert à créer un lien :

- `a:hover` : permet de préciser l'apparence d'un lien lorsque l'utilisateur survole le lien avec sa souris.
- `a:active` : permet de préciser l'apparence d'un lien lorsque l'utilisateur clique sur ce lien.
- `a:visited` : permet de préciser l'apparence d'un lien lorsque l'utilisateur a déjà cliqué sur ce lien.
- `a:link` permet de restreindre la définition de styles aux "vrais" liens, car la balise `<a>` peut également être utilisée pour positionner des ancres à l'intérieur d'une page.

Exemple :

CSS

Cette définition de style est à placer dans la section `<head>` de votre page ou dans une feuille de styles séparée.

```
a {color:brown;font-weight:bold;}
a:hover {color:red;}
a:active {color:blue;}
a:visited {color:purple;}
```

HTML

```
<a href="recherche.php"> Exemple de lien </a>
```

NOTE : les "*type modifiers*" sont également appelés "*pseudo-classes*".

5- On peut attribuer un style à un emboîtement particulier d'éléments HTML

Cette dernière règle offre une très grande souplesse dans la définition des styles.

Vous pouvez par exemple souhaiter afficher les liens situés à gauche de votre page en vert et ceux de droite en rouge. Voici comment vous pouvez procéder :

CSS

Propriétés communes aux deux classes.

```
.bloc-gauche, .bloc-droite{  
  position:absolute;  
  top:0px;  
  width:60px;  
  border:1px solid orange;  
}
```

Propriétés propres à chaque classe.

```
.bloc-gauche {left:0px;}  
.bloc-droite {right:0px;}
```

Couleur des liens pour chaque classe.

```
.bloc-gauche a {color:green; left:0px;}  
.bloc-droite a {color:red; right:0px;}
```

HTML

Utilisation des classes dans votre code HTML.

```
<div class="bloc-gauche">  
  <a href="page1.php">Lien 1 </a><br>  
  <a href="page2.php">Lien 2 </a><br>  
  <a href="page3.php">Lien 3 </a><br>  
</div>
```

```
<div class="bloc-droite">  
  <a href="page1.php">Lien 1 </a><br>  
  <a href="page2.php">Lien 2 </a><br>  
  <a href="page3.php">Lien 3 </a><br>  
</div>
```

Résultat :

[Lien 1](#)

[Lien 2](#)

[Lien 3](#)

[Lien 1](#)

[Lien 2](#)

[Lien 3](#)

Notez la formulation

.bloc-gauche, .bloc-droite{}

qui permet d'attribuer aux deux classes des propriétés de styles identiques, sans avoir à les déclarer deux fois.

On peut aussi définir un style particulier pour un emboîtement précis d'**éléments HTML**.

Exemple 1 - emboîtement de <p> dans <div> :

CSS

```
div p{font-weight:bold};
```

HTML

```
<p>Ce paragraphe n'est pas en gras</p>
```

```
<div>
```

```
  <p>Ce paragraphe est en gras car il est à l'intérieur d'un bloc div</p>
```

```
</div>
```

Exemple 2 - emboîtement de <p> dans 2 <div> successifs :

CSS

```
div div p{font-weight:bold};
```

HTML

```
<div>
```

```
  <div>
```

```
    Ce texte n'est pas en gras car il n'est pas dans un paragraphe.
```

```
    <p>Ce paragraphe est en gras car il est à l'intérieur de deux blocs div emboîtés, ce qui correspond bien à notre définition de style</p>
```

```
  </div>
```

```
</div>
```

Exemple 3 - emboîtement d'une classe dans un div:

CSS

```
.mon_style{font-weight:bold};
```

```
<-- Emboîtement
```

```
  div .mon_style{font-style:italic};
```

HTML

```
<p class="mon_style">Ce paragraphe est en gras</p>
```

```
<div>
```

```
  <p class="mon_style">Ce paragraphe est en gras et en italique</p>
```

```
</div>
```

```
<p>Ce paragraphe est écriture normale</p>
```

```
<div>Ce paragraphe est écriture normale</div>
```

Attention : Ne confondez pas la déclaration suivante :

```
div .mon_style{font-style:italic}; (avec un espace)
```

Avec celle-ci :

```
div.mon_style{font-style:italic}; (sans espace)
```

Ou avec celle là :

```
div, .mon_style{font-style:italic}; (avec une virgule)
```

Dans le premier cas, tout élément ayant les particularités suivantes :

1. - l'élément à la classe "**mon_style**"
2. - l'élément est emboîté dans un élément **<div>**

sera en italique (dans notre exemple, on constate que cela fonctionne avec l'élément **<p>**).

Dans le deuxième cas, seuls les éléments **<div>** ayant pour classe "**mon_style**" seront en italique (il ne s'agit donc pas d'emboîtement mais de la définition du contexte dans lequel "**mon_style**" doit s'appliquer).

Dans le troisième cas, tous les éléments **<div>** et tous les éléments ayant pour classe "**mon_style**" seront en italique. Il s'agit d'une **déclaration commune de propriété** .