

Chapitre 4

CSS3

1. Les trois styles de base possibles

À ce niveau du livre, il a été vu qu'il était possible de créer des styles de trois façons. Il faut retenir que quelle que soit la méthode utilisée, style de balise, de classe ou d'ID, les paramètres que nous pourrions modifier (couleur, position, police...) sont tous accessibles.

Cela signifie que pour changer la taille de la police, le type de style, que ce soit un style de balise, un style de classe ou un style d'ID, n'a aucune importance. Il va donc falloir définir quand utiliser telle ou telle méthode.

1.1 Le style de balise

Avec cette méthode, c'est le style de la balise HTML qui va être modifié, sans rien avoir à ajouter dans le code HTML. La « connexion » entre le CSS et la balise HTML se fait automatiquement. C'est très pratique.

Il est très difficile de changer le style d'une balise. Prenons par exemple la balise `<p>` qui permet d'afficher un paragraphe. Il y a de fortes chances que sur une page un paragraphe ait une certaine apparence et sur une autre page une apparence complètement différente. Dans ce cas, changer le style de la balise `<p>` ne serait pas une bonne solution.

Il existe par contre une balise qui aura quasi obligatoirement son style bien à elle : la balise `<body>`. Puisque toute notre page est contenue dans cette balise `<body>`, cela signifie que tout ce qui sera défini pour cette balise le sera pour toute notre page. Il est donc possible d'y définir le style principal de notre page pour la couleur, le choix de la police ou l'arrière-plan.

```
body {  
    font-family: Arial, sans-serif;  
    font-size: 12px;  
    background-color: rgb(150,150,150);  
}
```

Dans cet exemple, il y a trois éléments qui définissent le style : les deux premiers sont pour le texte, le dernier pour la couleur du fond de la page. Le détail de ces trois lignes sera vu plus loin. Ce qu'il faut retenir ici, c'est que puisque tout est affiché dans la balise `<body>` de notre page, les styles qui affectent la balise `<body>` affecteront également toute la page.

Il est donc souvent utile de créer un style de balise pour toute la page grâce à la balise `<body>` qui va finalement donner le ton du site.

Un autre cas avec les formulaires : ils permettent de renseigner des informations, de saisir un identifiant, du texte, et utilisent principalement la balise `<input>`. Le chapitre Les formulaires est entièrement consacré à ce sujet. La balise `<input>` a une propriété `type` qui peut avoir la valeur `text`, `password`, `button`, `date`, `range`, `number` et bien d'autres. Autrement dit, la création d'un style qui va modifier la balise `<input>` :

```
input {  
    ...  
}
```

va permettre de faire en sorte que tous les champs `<input>` de notre formulaire aient le même aspect, la même police, la même bordure...

Pour finir sur le style de balise, qui n'est pas celui qui sera le plus utilisé, il y a la balise `<h1>` (h pour *head*, l'en-tête d'un paragraphe) :

```
<h1>Un gros titre</h1>
```

Cette balise est utilisée pour des titres et va de `<h1>`, qui écrit très gros, à `<h6>`, qui écrit petit.

Écriture avec la balise h1

la balise h3

et la h6

Si notre site a régulièrement besoin d'un style pour un grand titre et d'un style pour un sous-titre, il serait possible, niveau HTML, d'avoir :

```
<h1>Les aventures de Toto</h1>
<h2>Le réveil de Toto</h2>
...
<h2>Toto prend son petit déjeuner</h2>
...
<h2>Toto va à la plage</h2>
```

Et niveau CSS cela donnerait :

```
h1 {
  color: rgb(85,97,153);
  font-size: 22px;
  margin:10px 0 5px 0;
  font-variant:small-caps;
}
h2 {
  color: rgb(65,77,130);
  font-size: 16px;
  margin:5px 0 3px 0;
}
```

Le style pour la balise `<h1>` va donc modifier, entre autres, la couleur : `rgb(85,97,153)` ; correspond au bleu, la dernière des trois valeurs étant bien plus grande que les deux autres. La propriété `color` affecte le texte, de même que `font-size`, qui donne aux caractères une taille de 22 pixels.

`margin` est un des termes CSS les plus utilisés. Il définit l'espace autour d'un élément (à l'extérieur de celui-ci). Ici, `margin` va permettre de définir l'espace autour du titre selon les quatre valeurs : `10px 0 5px 0` ;.

La première, 10px, correspond à la marge au-dessus du titre. Cela aurait pu également être écrit de cette façon :

```
margin-top:10px;
```

L'écriture ci-dessus convient très bien s'il n'y a qu'une seule marge à définir, mais s'il faut s'occuper des marges haute (top), droite (right), basse (bottom) et gauche (left), il est plus rapide et plus économique (en octets) de l'écrire sur une seule ligne. La première valeur est celle de la marge haute, et ensuite, en tournant dans le sens des aiguilles d'une montre, nous retrouvons les valeurs des marges suivantes, à savoir le côté droit, puis le bas, puis enfin la gauche.

Une écriture détaillée donne ceci :

```
margin-top:10px;  
margin-right:0;  
margin-bottom:5px;  
margin-left:0;
```

Remarque

Le fait d'écrire « 0 » et non « 0px » n'a pas d'importance. Il y a différentes unités de mesure en CSS, mais si la valeur indiquée est « 0 », cette valeur est nulle quelle que soit l'unité. Il n'est pas obligatoire dans ce cas de préciser l'unité.

Pour finir avec les styles, H1 sera composé de petites majuscules grâce à la propriété `font-variant`, ce qui est plutôt adapté pour un titre.

```
font-variant:small-caps;
```

H2 utilise les mêmes propriétés de style avec des valeurs différentes. Les valeurs pour la couleur de H2 étant plus faibles que celles de H1, la couleur sera plus foncée.

Il n'y a pas beaucoup d'autres cas où le style de balise est utilisé seul, à part pour la balise `<a>` (pour *anchor*, ancre), qui est en fait la balise qui crée les liens et qui est abordée à la section Pour les liens.

Ce style pourra par contre être utilisé en combinaison avec d'autres styles (voir la section Combinaison des trois méthodes).

1.2 Le style de classe

Le style de classe est un peu à l'opposé du style de balise, car il ne s'applique pas tout seul et est énormément utilisé.

Pour ce qui est de son utilisation, côté CSS il faut définir le style en faisant précéder son nom d'un point. Ici, le style de classe `.cl cliclic` :

```
.cl cliclic{  
    cursor: pointer;  
}
```

Et son utilisation dans le code HTML :

```
<p class='cl cliclic' onClick='niveau(2) '>Accéder au niveau 2</p>
```

La propriété `class` permet de donner le nom de la classe CSS qui doit être appliquée. La classe `cl cliclic` va changer le pointeur de la souris et afficher l'icône de la main à la place, qui est normalement affichée quand le pointeur de la souris est au-dessus d'un lien.

Remarque

Lorsqu'un style de classe est créé dans du CSS, son nom est précédé d'un point. Lorsqu'il est utilisé en HTML, le style s'écrit sans le point !

Dans cet exemple, grâce au style de classe `.cl cliclic`, l'utilisateur sera averti qu'il peut cliquer sur le texte « Accéder au niveau 2 ». Le fait de cliquer déclenchera l'événement `onClick` qui lancera la fonction `niveau(2)` ;

Le style de classe a également un autre intérêt. Lorsqu'il est utilisé dans le code HTML, il est possible d'affecter plusieurs styles de classe à une même balise HTML.

Par exemple, le style `.bordureBleu` :

```
.bordureBleu {  
    border: 1px #338 solid;  
}
```

qu'on peut appliquer au paragraphe en écrivant :

```
<p class='cl clic clic bordureBleu' onClick='niveau(2) '>Accéder au  
niveau 2</p>
```

Deux styles sont maintenant appliqués à la balise `<p>` : un premier qui change la forme du pointeur de la souris et un second qui ajoute une bordure bleue.

■ Remarque

Noter que les deux styles sont simplement séparés d'un espace dans la partie HTML : `class='cl clic clic bordureBleu'`.

1.3 Le style d'ID

Le dernier style a comme particularité de référencer un **élément unique**. C'est ce qui le différencie principalement des deux autres. Il est utilisé pour référencer des éléments précis. Puisqu'avec le langage HTML tout est défini dans des balises imbriquées, il est fréquent de donner un ID à la balise principale. Cela peut être appliqué au bloc contenant le menu, au pied de page, ou à une zone qui peut apparaître ou disparaître.

Pour la partie CSS, l'ID est précédé d'un dièse #.

Prenons l'exemple d'une zone servant à afficher une légende :

```
#legende {  
    width: 180px;  
    background-color: rgba(255,255,255,0.2);  
}
```

`#legende` est défini pour avoir une largeur de 180 pixels et une couleur blanche quasi transparente, le rouge, le vert et le bleu étant à leur valeur maximum (255) et l'opacité étant proche de zéro (0.2). Le bloc de la légende sera donc de la même couleur que le fond de la page, mais un peu plus claire.

Pour le code HTML :

```
<div id="legende"></div>
```

Il est correct de se dire qu'au lieu d'utiliser un style d'ID avec #legende, il aurait été possible d'utiliser un style de classe en écrivant .legende, et naturellement en remplaçant id='legende' dans le code HTML par class='legende'. Pour le navigateur, il n'y a aucune différence, le fonctionnement de la page est identique. C'est la raison pour laquelle utiliser le style d'ID est surtout une habitude à prendre, que ce soit pour un élément unique ou pour un élément principal.

Cela va permettre de créer des styles qui vont avoir pour référence le style #legende. Si #legende est le style d'une zone, cette zone va contenir du code HTML et pourra être le point de départ de tous les styles des éléments contenus dans la légende.

Par exemple imaginons que la légende contienne une liste :

```
<div id="legende">
  <ul>
    <li>
      
      <div class="lbl">Habitation</div>
    </li>
    <li>
      
      <div class="lbl">Voiture</div>
    </li>
  </ul>
</div>
```

Cette liste (liste à puces) contient deux items de liste . Chaque item contient une image et un bloc <div> pour écrire un libellé.

Voici le résultat :



Si aucun style n'est appliqué au couple , l'affichage par défaut sera celui d'une liste dont chaque item sera précédé par un petit cercle, la puce. Il faut donc pouvoir modifier le style de la balise .