

LES FEUILLES DE STYLE.

Patrick Poulingeas.

25 mars 2005.

1. Introduction.

Les feuilles de style (CSS : *Cascading Style Sheets*) servent à contrôler la mise en page des différents éléments qui composent un document HTML. Avec les feuilles de style, on peut – entre autre - définir la couleur d'une police, la taille de celle-ci, le positionnement d'un objet (image, texte, etc.), ou bien encore l'espacement entre les paragraphes.

L'objectif principal des CSS est de permettre une séparation rigoureuse entre le contenu et la forme d'un document Web. Cette séparation permet notamment d'avoir différentes possibilités de présentation (affichage avec un navigateur ou impression) et de réutiliser une mise en forme pour différentes pages.

Il y a deux normes pour les CSS : CSS niveau 1 (introduit en 1996 et revu en 1999) et CSS niveau 2 (dont la dernière spécification date de 2004 et s'étend sur 349 pages). L'un des apports de CSS2 est le fait de pouvoir travailler au niveau du document dans sa globalité et non plus uniquement au niveau des balises. Ainsi, on pourra décider qu'un bloc sera affiché en haut à droite de l'écran de manière fixe, même quand l'internaute fera défiler la page HTML pour la parcourir quand elle dépasse la hauteur de l'écran. Internet Explorer 6.0 supporte CSS1 et partiellement CSS2 ; Firefox 1.0 supporte CSS1 et CSS2.

Les feuilles de style permettent de créer un affichage particulier pour une ou plusieurs balises. Le terme « **cascading** » apparaissant dans l'acronyme CSS provient du fait qu'une balise incluse à l'intérieur d'une autre, soit hérite par défaut du style de sa balise parente, soit a la possibilité de modifier ce style. Pour déterminer quel style s'applique exactement à une balise, le navigateur doit donc parcourir l'arbre des éléments composant la page HTML en examinant chacune des indications de style spécifiquement associées à certains éléments.

Afin de vérifier la syntaxe d'une feuille de style, le W3C met à la disposition du webdesigner un validateur (disponible à l'URL suivante : <http://jigsaw.w3.org/css-validator/>).

2. Un exemple.

Supposons que nous désirions qu'un titre de taille 1 ait une couleur bleue. Il y a trois façons d'obtenir cet effet avec les CSS.

2.1. Utilisation de l'attribut style.

On spécifie ici directement le style qui ne sera valable que pour l'élément h1 courant :

```
<h1 style="color: blue">
```

Introduction.

```
</h1>
```

Résultat obtenu :

Introduction.

Un style est constitué de différentes propriétés (ici, la propriété « color ») auxquelles on affecte des valeurs (ici, le nom de couleur « blue »).

Les noms des propriétés et les valeurs ne sont pas sensibles à la casse.

On sépare les différentes propriétés d'un style par un point-virgule.

Exemple :

```
<p style="color: red; font-family: Arial; font-style: italic">
  Un essai.<br />
  Une tentative.<br />
</p>
```

Affichage obtenu :

Un essai.
Une tentative.

2.2. Feuille de style intégrée à la page HTML.

On va définir un style par défaut pour tous les éléments h1 du document. On pourra toutefois utiliser un autre style en recourant à l'attribut style pour un élément h1 particulier (d'où le qualificatif de feuilles de style **en cascade**).

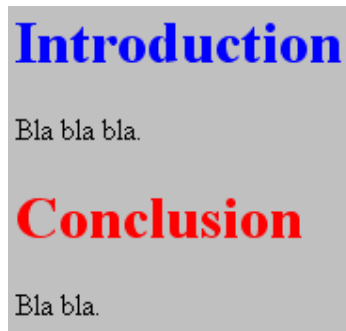
Pour insérer une feuille de style dans une page HTML, on ajoute à l'intérieur de l'élément head la structure suivante :

```
<style type="text/css">
  <!-- définition du style -->
</style>
```

Exemple d'utilisation :

```
<!DOCTYPE ...>
<html>
  <head>
    <title>Un titre significatif</title>
    <style type="text/css">
      h1 { color: blue }
    </style>
  </head>
  <body>
    <h1>Introduction</h1>
    <p>Bla bla bla.</p>
    <h1 style="color: red">Conclusion</h1>
    <p>Bla bla.</p>
  </body>
</html>
```

Affichage obtenu :



Remarques :

- « `h1 { color : blue }` » est appelée une règle. « `h1` » est le sélecteur de la règle, « `color : blue` » la déclaration de la règle.
- Il est possible de regrouper des balises pour leur donner un même style. On les sépare alors par des virgules. Exemple :
`h1, h2, h3 { font-style : italic }`

2.3. Les feuilles de style externes.

Il s'agit de la solution à privilégier puisqu'elle permet :

- de ne pas mélanger du contenu HTML avec la définition d'un style,
- de pouvoir réutiliser simplement un style pour différentes pages.

Pour employer cette méthode, on ajoute une balise `<link>` dans l'élément `head` de la page HTML.

Exemple :

```
<link rel="stylesheet" type="text/css" href="monStyle.css">
```

L'attribut `rel` indique le type de lien (ici : une feuille de style).

L'attribut `type` sert à spécifier le type (MIME) de la feuille de style.

L'attribut `href` a pour valeur une URI référençant la feuille de style.

Remarque : Le fichier `monStyle.css` ne doit pas contenir de code HTML.

Exemple d'utilisation :

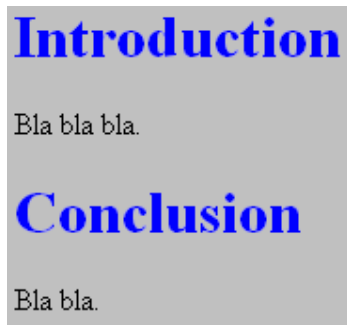
```
<!DOCTYPE ...>
<html>
  <head>
    <title>Un titre significatif</title>
    <link rel="stylesheet" type="text/css" href="monStyle.css">
  </head>
  <body>
    <h1>Introduction</h1>
    <p>Bla bla bla.</p>
    <h1>Conclusion</h1>
    <p>Bla bla.</p>
```

```
</body>
</html>
```

On a le fichier monStyle.css suivant :

```
/* Commentaire */
h1 { color: blue }
```

Résultat obtenu :



Remarques :

- Il est possible d'avoir plusieurs balises <link> pour un même document HTML. Ceci permet de combiner des définitions de style.
- Les commentaires CSS ont la même forme que les commentaires du C.

3. Classes et identificateurs.

Nous avons vu précédemment que l'on peut associer un style (constitué d'une ou plusieurs propriétés) à un élément.

Exemple :

```
p { font-size : 12pt; color : white }
```

Ce style s'appliquera alors par défaut à tous les paragraphes du document.

Pour distinguer différents types de paragraphes et leur assigner différents styles, on dispose de la notion de classe. On commence par définir dans la feuille de style différentes classes pour une balise. On positionne ensuite l'attribut class dans une balise du document HTML pour lui associer le style de la classe indiquée. Les balises n'ayant pas d'attribut class auront le style par défaut déterminé par le navigateur.

Exemple :

```
<!DOCTYPE ...>
<html>
  <head>
    <title>Un titre significatif</title>
    <style type="text/css">
      p.important { font-size: 1cm; font-weight: bold;
                    color: red }
      p.futile { font-size: 5mm; color: blue }
    </style>
  </head>
```

```

<body>
  <p class="important">Lisez Login:</p>
  <p class="futile">Bla bla.</p>
  <p>Normal</p>
</body>
</html>

```

Résultat obtenu :



Si l'on souhaite affecter un style à tous les éléments d'une classe, on dispose de la notation :

```
.maClasse { color : green }
```

Exemple :

```

<!DOCTYPE ...>
<html>
  <head>
    <title>Un titre expressif</title>
    <style type="text/css">
      .important { font-size: 0.6cm; font-weight: bold;
                  color: red }
    </style>
  </head>
  <body>
    <h1 class="important">Un message important :</h1>
    <p class="important"><em>Lisez Linux Magazine</em></p>
  </body>
</html>

```

Résultat obtenu :



Pour limiter un effet de style à une seule balise de la page HTML, on a recours à un identifiant.

Exemple :

```
<!DOCTYPE ...>
<html>
  <head>
    <title>Un titre significatif</title>
    <style type="text/css">
      #beau { text-align: right; color: green }
    </style>
  </head>
  <body>
    <p id="beau">Lisez Linux Magazine</p>
    <p>Normal</p>
    <!--
      <p id="beau">Bla bla</p>    → Interdit puisque un
      <h2 id="beau">1</p>        identifiant doit être unique.
                                   → Interdit pour la même
                                   raison.
    -->
  </body>
</html>
```

Affichage obtenu :



Remarque : L'interdiction qui est mentionnée en commentaire est souvent ignorée par les navigateurs. Elle ne l'est pas par le validateur du W3C.

4. Les éléments div et span.

La balise <div> sert à faire un regroupement dans un document HTML.

Exemple d'utilisation :

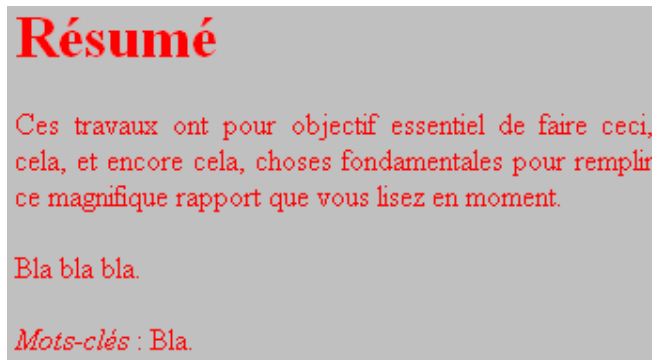
```
<!DOCTYPE ...>
<html>
  <head>
    <title>Théâtre</title>
    <style type="text/css">
      div.resume { text-align : justify;
                    color: red }
    </style>
  </head>
  <body>
    <div class="resume">
      <h1>Résumé</h1>
```

```

        <p>
            Ces travaux ont pour objectif essentiel de faire
            ceci, cela, et encore cela, choses fondamentales
            pour remplir ce magnifique rapport que vous lisez
            en moment.
        </p>
        <p>
            Bla bla bla.
        <p>
            <em>Mots-cl&eacute;s</em> : Bla.
    </div>
</body>
</html>

```

Affichage obtenu :



Les blocs constitués par les éléments div peuvent être positionnés au pixel près.

Exemple :

```

<div id="bloc1" style="position: absolute; left: 10; top: 10;
    color : green; font-size: 0.7cm ">
    <p>
        Contenu du bloc 1
    </p>
</div>
<div id="bloc2" style="position: absolute; left: 8; top: 9;
    color : red; font-size: 0.5cm ">
    <p>
        *** Le bloc 2 ***
    </p>
</div>

```

Résultat obtenu :



L'élément span est similaire à l'élément div. On utilise habituellement div pour des blocs de texte entiers et span pour des petites portions de texte (mots ou lettres).

5. L'attribut media.

Il s'agit d'un attribut de la balise link. Avec cet attribut, on demande au navigateur de charger une feuille de style différente en fonction du médium de sortie. On peut ainsi écrire :

```
<link rel="stylesheet" type="text/css" href="normal.css" media="screen">
```

```
<link rel="stylesheet" type="text/css" href="impression.css" media="print">
```

On aura des feuilles de style différentes pour un affichage écran et pour une impression. Avec l'exemple ci-dessus, le navigateur affichera la page en prenant en compte la feuille de style « normal.css ». Par contre, en sélectionnant « Fichier/Aperçu avant impression » sous Firefox ou Internet Explorer, vous observerez que c'est « impression.css » qui sera employée pour la mise en forme du document avant l'impression.

Cette utilisation des feuilles de style est à privilégier sur des solutions où l'on met à la disposition de l'internaute une page « normale » pour l'affichage et une page « spéciale » pour l'impression.

D'autres types de media ont une valeur standard : projection, handheld, etc.

6. Quelques propriétés et valeurs.

Propriété	Valeurs possibles	Description
font-size	small, medium, large. Taille définie en pixels (px), en cm (cm), en mm (mm), en pourcentage (%)	Taille de la police
font-family	serif, sans-serif, cursive, ... Nom de la police (Arial, Verdana, etc.)	Type de police
font-weight	normal, bold, bolder, lighter	Poids de la police
font-style	normal, italic, oblique	Style de la police

Propriété	Valeurs possibles	Description
color	Nom ou valeur hexadécimale de la couleur.	Couleur de la police
background-image	URL du fichier. Exemple : url(marble.jpg)	Image d'arrière plan d'un élément
background-repeat	repeat, no-repeat, repeat-x, repeat-y	Répétition de l'image d'arrière-plan
background-color	Nom ou valeur hexadécimale de la couleur.	Couleur d'arrière-plan d'un élément

Propriété	Valeurs possibles	Description
text-align	left, right, center, justify	Alignement du texte
text-indent	Valeur en pixels ou en pourcentage	Retrait de la première ligne
text-decoration	none, underline, overline	Décoration du texte
text-transform	none, capitalize, uppercase, lowercase	Casse du texte

7. Les liens hypertextes.

On dispose de facilités pour attribuer un style aux différents liens hypertextes d'une page HTML.

Exemple :

```
a:link { color: red }      /* Style pour des liens non visités */
a:visited { color: blue } /* Style pour des liens déjà visités */
a:hover {color: green }   /* Style lorsque le pointeur de la souris survole le
                           lien */
```

Utilisation :

```
<!DOCTYPE ...>
<html>
  <head>
    <title>Titre</title>
    <style type="text/css">
      a:link { color: red}
      a:visited { color : blue }
      a:hover {color : green }
    </style>
  </head>
  <body>
    <a href="asuivre.html">
      A suivre...
    </a>
  </body>
</html>
```

« link », « visited », etc. sont appelées des pseudo-classes.