

# Les feuilles de style

## Objectifs

- Comprendre la structure d'une feuille de style,
- Créer des feuilles de style simples,
- Appréhender la notion de cascade.

## Syntaxe d'un élément de style

La définition générale d'un élément de style est la suivante :

`balise { propriété de style: valeur; propriété de style: valeur }`

Ce qui se lit : à la balise déterminée, appliquer la première propriété de style ainsi que la propriété de style suivante.

### Exemple

`h2 {font family: Arial; color: #999999}`

Ce qui se lit : à la balise de titre h2, appliquer la propriété de police de valeur Arial, et la couleur désignée.

- Les feuilles de style s'ajoutent principalement à des balises, par exemple **p**, **body**, mais aussi à des éléments **a:link** pour un lien (non-visité).
- Les propriétés de style sont entourées par des accolades "{".
- Le couple "propriété de style et valeur" est séparé par un double point (:).
- Chaque couple "**propriété de style et valeur**" est séparé par un point-virgule (;).
- Il n'y a pas de limite pour le nombre de couples "**propriétés de style et valeur**".
- L'espace entre propriétés de style et valeur n'est pas obligatoire mais aide fortement à la lisibilité du code source.
- Bien que les styles ne soient pas sensibles aux majuscules et minuscules (case insensitive), l'usage veut que les feuilles de style soient notées en minuscules.
- Pour la lisibilité toujours, les styles peuvent être écrits sur plusieurs lignes :  
`h2 {font family: Arial;  
color: #999999}`
- Plusieurs valeurs peuvent être attribuées à une même propriété. Dans ce cas, les différentes valeurs sont séparées par des virgules.  
`h3 {font-family: Arial, Helvetica, sans-serif}`
- Un même style peut être attribué à plusieurs balises (séparées par des virgules).  
`h1, h2, h3 {font-family: Arial; font-style: italic}`

## Choix des polices de caractères

Il existe 5 familles de polices de caractères dites « génériques » :

- Serif
- Sans-serif
- Monospace
- Cursive
- Fantasy

Il n'y a pas de grosses différences avec les familles. Il vaut mieux donc se contenter des familles **serif** et **sans-serif**, les sans-serif étant les plus lisibles à l'écran.

Les familles de polices se déclarent à l'aide de l'attribut **font-family**, à déclarer dans le **body** pour que toute la page en bénéficie (c'est la notion d'héritage). On finit toujours la liste de fontes déclarées par sa famille générique à laquelle elle appartient. Ainsi, si la police déclarée n'existe pas sur l'ordinateur de votre visiteur, le navigateur affichera la police appartenant à la même famille.

Dans le cas où la police n'est pas disponible sur l'ordinateur de votre visiteur, il y a possibilité de la télécharger.

`<link href='http://fonts.googleapis.com/css?family=Doppio+One' rel='stylesheet' type='text/css'>`

Pour un large choix de polices, voir le site <http://www.google.com/fonts>

## Exercice 1 – Feuille de style interne

Les feuilles de style dites **internes** s'appliquent à tout un document HTML mais à ce seul document. Il est ainsi possible d'incorporer les feuilles de style internes dans l'en-tête, entre les balises <head>.

Effet de style qui enlève le soulignement à tous les liens d'un document HTML. Cet effet s'obtient par le code :

```
1 <html>
2 <head>
3 <style>
4 <style type="text/css">
5 a {text-decoration:none}
6 </style>
7 </head>
8 <body>
9 <a href="test.htm">Ceci est bien un lien</a>
10 </body>
11 </html>
```

## Exercice 2 – Feuille de style externe

Les feuilles de style externes (externes car regroupées dans un fichier séparé), sont idéales pour donner un style commun à toute une série de pages. Plutôt que de devoir mettre dans chaque page les lignes de code du style, il suffit de les encoder une seule fois et elles pourront servir à toutes les pages qui y font référence.

Pour appeler ce document; cela se fait par la balise <link> qui se positionne dans l'en-tête du document HTML <head>.

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

- La balise **<link>** avertit le navigateur qu'il faudra réaliser un lien.
- L'attribut **rel=stylesheet** précise qu'il y trouvera une feuille de style externe. Remarquez que c'est stylesheet au singulier donc sans s.
- L'attribut **type="text/css"** précise que l'information est du texte et du genre Cascading Style Sheets (css).
- L'attribut classique de lien **href="destination"** donne le chemin d'accès et le nom du fichier à lier. Ici le fichier style.css est supposé être dans le même dossier que les pages Web.

1. Créer la feuille de style nommée **style1.css** :

```
1 body {
2   color: #000000;
3   font-size: 12px;
4   font-family: verdana, sans-serif;
5   background-color: #e8f6e9
6 }
7
8 p {
9   color: #333333;
10  font-size: 16px;
11  font-weight: bold;
12  text-align: justify
13 }
```

2. Créer le document HTML **feuille1.html** appelant la feuille de style.

```
1 <html>
2 <head>
3   <title>CSS externe</title>
4   <link rel="stylesheet" type="text/css" href="style1.css">
5 </head>
6 <body>
7   <p>Ceci est un paragraphe</p>
8 </body>
9 </html>
```

## Exercice 3 –Style local

Les styles sont insérés n'importe où dans la page en affectant un attribut style à une balise. À utiliser avec modération cependant, pour modifier quelques mots ou quelques lignes, une portion de texte, un paragraphe ou une cellule de tableau, puisqu'on se détourne ainsi de la véritable finalité du CSS, soit la séparation du contenu et de la mise en forme.

```

1  <html>
2  <head>
3    <title>Titre de la page</title>
4  </head>
5  <body>
6    <p style="color:#ff0000;font-weight:bold">
7      Le texte du paragraphe sera rouge (#ff0000) et gras (bold).
8    </p>
9  </body>
10 </html>

```

## Exercice 4 –Les éléments class et id

Une feuille de style est appliquée à une balise donnée. Les éléments **class** et **id** permettent, eux, de donner des effets de style différents à une même balise.

La définition d'un style devient :

```
balise.nom_de_classe { propriété de style: valeur }
```

Comme la mention de la balise est facultative, on peut aussi indiquer :

```
.nom_de_classe { propriété de style: valeur }
```

Pour appeler l'effet de style dans le document, ajoutez le nom de la classe à la balise :

```
<balise class="nom_de_classe"> .... </balise>.
```

**Exemple :**

1. Mettre un type de paragraphe en gras et en rouge désigné par le nom « important » et un autre type de paragraphe en italique nommé « remarque ».

```

1  <html>
2  <head>
3    <title>CSS</title>
4    <style type="text/css">
5      .important {font-weight: bold; font-color: red;}
6      .remarque {font-style:italic;}
7    </style>
8  </head>
9  <body>
10   <p class="important">Feuilles de style en cascade</p>
11   <p class="remarque">Cascading Style Sheets</p>
12 </body>
13 </html>

```

2. Ajouter un type de paragraphe souligné et vert désigné par le nom « attention ».

Il existe aussi l'élément **id** dont la fonction est similaire à l'élément class à ceci près. Comme la convention nom/point/nom est utilisée aussi en JavaScript, il a fallu trouver une autre convention d'écriture pour utiliser les feuilles de style avec du JavaScript.

Ce sont les id, aussi appelés les identifiants. Les id fonctionnent exactement comme les class. La syntaxe est :

```
#nom_de_id { propriété de style: valeur }
```

```
<balise id="nom_de_id"> .... </balise>
```

Un seul appel à #nom\_de\_id par page pourra être effectué.

Si vous pensez utiliser des feuilles de style, mais sans utiliser des scripts, oubliez au plus vite id et utilisez exclusivement les éléments class.

## Exercice 5 –Les balises span de div

Ces balises introduisent dans la conception de pages Web ce que l'on pourrait comparer aux zones de texte pour ceux qui utilisent les outils bureautiques ou aux calques pour les infographistes.

Ces zones ou boîtes ainsi déterminées ne prennent de sens que par les effets de style qui leur sont appliquées. Ces zones peuvent s'inclure dans le texte de la page mais aussi en relief par rapport à la page grâce à la propriété **z-index** des feuilles de style.

- La balise **<div>** crée une boîte ou un élément de type bloc. Ces éléments servent à distinguer des parties entières de texte comme les paragraphes **<p>**, les titres **<h>**, les listes **<ol>** ou **<ul>**, les citations **<blockquote>**, etc. Ces éléments sont distingués du reste du contenu par un retour chariot.
- La balise **<span>** est un élément en-ligne qui sert à distinguer une portion de texte incluse dans un élément bloc. Par exemple, quelques mots d'un paragraphe.

```
<span style="propriétés: valeur"> ... </span>  
<div style="propriétés: valeur"> ... </div>  
<span class="nom_de_classe"> ... </span>  
<div class="nom_de_classe"> ... </div>
```

### Exemples

```
1 <html>  
2 <head>  
3 <title>CSS</title>  
4 </head>  
5 <body>  
6 <h2>Nouveau, nouveau...</h2>  
7 <div style="background-color:#99ccff">Un concept nouveau est apparu  
8 dans l'univers de la publication Web, c'est celui des feuilles de style CSS  
9 </div>  
10 </body>  
11 </html>
```

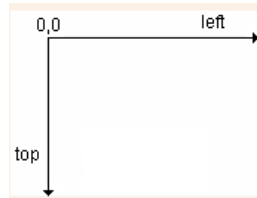
```
1 <html>  
2 <head>  
3 <title>CSS</title>  
4 </head>  
5 <body>  
6 <h2>Nouveau, nouveau...</h2>  
7 <p>Un concept nouveau est apparu dans l'univers de la publication Web, c'est  
8 celui des <span style="background-color:#99ccFF;">feuilles de style  
9 CSS</span>.  
10 </p>  
11 </body>  
12 </html>
```

Ce concept de boîte est dans certaines circonstances (avec ou sans doctype) interprété différemment par Explorer par Chrome et par Firefox. Sans entrer dans le détail, ce cas fournit un bel exemple de différences d'affichage d'un même code par des navigateurs différents.

## Exercice 6 –Le positionnement avec span et div

Il est possible de positionner, au pixel près, du texte ou une image avec les feuilles de style. Le positionnement peut se réaliser en position absolue ou en position relative.

La **position absolue** {position: absolute} se détermine par rapport au coin supérieur gauche de la fenêtre du navigateur. Les coordonnées du point initial sont top = 0 et left = 0. Les coordonnées d'un point s'expriment en pixels, de haut en bas pour top et de gauche à droite pour left.



La **position relative** {position: relative} se détermine par rapport à la position normale ou originale de l'élément dans la page, par exemple un élément du code Html.

Le positionnement absolu sera le plus souvent utilisé car il est plus facile et plus sûr. Mais il est important de ne pas oublier que le visiteur de votre site n'a peut-être pas la même résolution d'écran que vous. Ce qui tombe pile avec une résolution 800x600, risque de se retrouver n'importe où avec une résolution 1024x768.

Créer un effet d'ombrage en superposant, mais de façon légèrement décalée, un texte en rouge et son ombre (le même texte) en gris (créez deux zones en position absolue sous forme de <span> de 200 pixels de large et de 55 pixels de haut).

## Exercice 7 - Entrainement

1. Ecrire la syntaxe générale de la déclaration d'un style.
2. Ecrire le sélecteur et le style donnant une couleur rouge et un fond noir aux éléments <h1>et <h3>.
3. Ecrire le sélecteur afin que tous les éléments de la page soient écrits en vert.
4. Ecrire une classe qui définit un fond jaune et appliquez-la aux éléments <h1>et <p>.
5. Ecrire une classe spécifique à un élément <code>afin que son texte soit bleu.
6. Ecrire les classes correspondant aux styles « fondgris », « textevert » et « textejaune ». Appliquez la première et la troisième à un paragraphe, puis la première et la deuxième à une division <div>.
7. Ecrire le sélecteur afin que l'élément dont l'attribut id vaut « menu » ait un fond rouge.

## Exercice 8 - Synthèse

Réaliser la page HTML ci-dessous (le texte seulement) en utilisant la Doctype Strict et l'encodage UTF-8.

Créer une feuille de style externe pour obtenir le résultat ci-dessous.



## Annexes

### Les unités de mesure

Le HTML n'utilisait que le pixel ou le pourcentage. Les feuilles de style permettent d'utiliser d'autres unités de mesure.

px	en pixel
%	en pourcentage
in	en pouce (1 inch = 2,54 centimètres)
cm	en centimètre
mm	en millimètre
pt	en point (1 point = 1/72 inches)
pc	en pica (1 pica = 12 points)

### La codification des couleurs

Outre la codification par nom et en valeur hexadécimale, les feuilles de style introduisent d'autres façons d'encoder les couleurs. Soit :

- par un nom fonctionnel,
- par la valeur hexadécimale composée de 6 chiffres précédée d'un dièse # soit #000000. Donc comme en Html mais sans les guillemets !
- par une valeur hexadécimale à 3 chiffres. Chaque chiffre est alors implicitement dupliqué, ainsi #fd3 est équivalent à #ffdd33,
- par la notation fonctionnelle rgb qui prend 3 arguments en l'occurrence 3 nombres entiers compris entre 0 et 255 ou 3 pourcentages entre 0% et 100%, par exemple, color : rgb(255,0,0) ou color : rgb(50%,50%,50%).