

# Les feuilles de style

Les feuilles de style (*style sheet*) regroupent des définitions de mises en forme qui peuvent être reprises dans plusieurs pages. C'est un moyen simple pour uniformiser l'aspect d'un site. Le second objectif du recours aux styles est de séparer autant que possible la présentation de la page de son contenu. Le langage de feuille de style (*stylesheet language*) a pour ce faire une syntaxe plus structurée que le HTML et un vocabulaire plus complet.

Il ne faut pas croire que les styles ne sont jamais définis que dans des feuilles de styles. On les déclare souvent aussi dans les entêtes (<head>) des documents HTML ou même parfois directement dans le corps du document (le <body>) comme attributs de certaines balises.

Exemple : Imaginez qu'on veuille que tous les titres importants s'affichent en rouge, il y a quatre manières de s'y prendre :

1. Spécifier, en langage HTML conventionnel, qu'il faut utiliser du rouge.  
Cette indication serait répétée !!! chaque fois qu'on utilise la balise <h1>

```
<h1><font color="red" > ... </font></h1>
```

On a utilisé pour cela la balise <font> qui est maintenant dépréciée.  
Ce n'est plus le rôle du html de s'occuper de la présentation.

2. Utiliser l'**instruction style** en tant qu'attribut de la balise <h1>

```
<h1 style=" color : red " > ... </h1>           = « style en ligne »
```

La syntaxe est différente mais elle n'apporte jusqu'ici aucun avantage puisqu'il n'y a pas moins à écrire et que la présentation reste fortement liée au contenu.

3. Définir le style dans l'entête du document HTML. Il s'applique alors à l'ensemble du document. On parle alors de « **feuille de style interne** » ou « **incorporée** ».

Tous les titres du niveau h1 dans la page auront par défaut ce même style.

```
<html>
<head>
  <title> Le temps des cerises</title>
  <style type="text/css" >
    h1 { color : red }
  </style>
</head>
...
</html>
```

#### 4. Définir le style dans un document à part : « **feuille de style liée** »

Exemple:

La feuille de style "MonStyle.css" Cette même feuille peut dès lors être reprise dans toutes les pages d'un même site pour y créer une certaine homogénéité d'aspect

```
h1 { font-size: 18; color: red }  
h2, h3 { font-family: Comic sans ms; color : green }  
p { font-family : verdana; color : blue }
```

Il ne reste plus qu'à indiquer le lien (<link ...>) à cette feuille de style dans l'entête <head> des documents.

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

Une autre technique : l'importation :

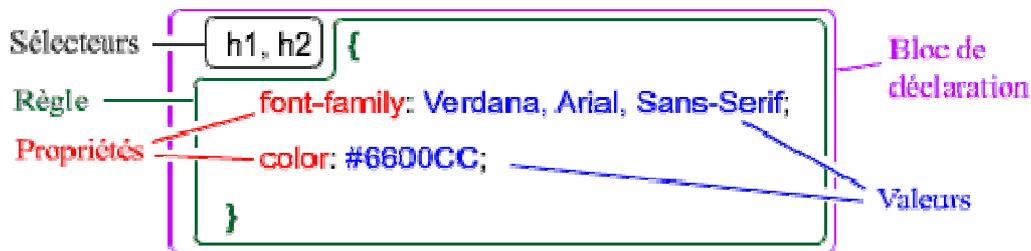
```
@import url( MonStyle.css);
```

La directive @import peut être incluse dans une feuille de style

## Les styles en cascade

La feuille de style est rédigée avec une syntaxe autre que celle du HTML. Le langage utilisé est le "CSS" (*Cascading Style Sheet*), en français : "feuille de style en cascade". Cette expression "en cascade" traduit le fait que les styles peuvent être redéfinis et que de nouvelles règles peuvent masquer celles qui les précèdent. Des règles définies en ligne peuvent recouvrir d'autres définies dans l'entête qui elles-mêmes en remplaçaient d'autres plus globales encore, définies dans une feuille de style.

## Syntaxe CSS



La syntaxe CSS ou plutôt la disposition des éléments de ce langage ressemble à celle des langages de programmation tels que le langage C.

Les instructions, ou plus exactement les **déclarations**, se terminent par un point-virgule. Les blocs de déclarations sont délimités par des accolades et forment ce qu'on appelle des **règles**.

La virgule sert de séparateur pour les composants d'une liste (les éléments auxquels on assigne un même style ou les différentes valeurs qu'on attribue à une même propriété).

```
h1 { font-size: 18; color: red }  
h2, h3 { font-family : Comic sans ms, Verdana, Arial ; color : green }  
#introduction { font-family : Courier new, Helvetica ; color : blue }  
.danger { color: red }  
div.theorie { font-family: Times, Serif; }  
div.exemple { font-family: Arial, Sans-Serif; }
```

Le sélecteur est soit :

- une balise html telle que : p, h1, h2, ... h6, ol, ul, li, div, span, em, strong, table, tr, td, th, ... etc.
- un élément sélectionné par un identificateur "id",
- une classe d'éléments "class"
- ou encore une combinaison de ces 3 types de sélecteurs.

Les **id** et les **class** sont des attributs que l'on applique à des éléments html pour leur assigner des noms.

## Sélecteur *id*

Un sélecteur **id** sert à spécifier un élément unique dans la page.

Exemple :

Pour donner le nom "introduction" à la division d'une page, on encadre cette division par les balises <div> : `<div id="introduction"> ... </div>`.

Il n'y aura pas d'autres éléments portant le même identificateur dans cette page.

Lors de la déclaration des styles, les sélecteurs de type id seront précédés du caractère #.

```
#introduction { font-style : italic ; }
```

## Sélecteur *class*

Plusieurs éléments d'une même classe peuvent coexister dans une même page.

Les classes permettent d'assigner le même style à des éléments regroupés sous des balises différentes.

Exemple :

`<span class="code">...</span>` et plus loin `<strong class="code"> ... </strong>`

Inversement, une même balise peut varier de classe, ce qui permet d'avoir plusieurs styles pour un même élément HTML.

Exemple :

`<div class="theorie">...</div>` et plus bas `<div class="exemple"> ... </div>`

Lors de la déclaration des styles, les sélecteurs de type class seront précédés d'un point .

```
.code { font-family : Courier ; }
```

**Attention !!!** `p.code { ... }` ≠ `p code { ... }` 🙅

`p.code { ... }` = appliquer un style aux paragraphes de class code

`p code { ... }` = appliquer un style qu'aux éléments de class code imbriqué dans un paragraphe, quel que soit le degré de parenté entre l'élément de class code et son ancêtre paragraphe.

Les espaces entre sélecteurs servent à définir ce qu'on appelle des « sélecteurs descendants »

Les noms d'ID ou de CLASS sont sensibles à la casse ! #EnTete est différent de #entete

## Tableau des principales propriétés utilisables dans les styles

Le tableau ci-dessous rassemble une liste de propriétés intéressantes que heureusement il est inutile de mémoriser si vous utilisez un éditeur HTML qui gèrent les feuilles de style tels que les outils gratuits: Visual Studio Express ou Netbeans.

Propriété	Valeurs / Descriptions / Exemples
background-attachment	scroll - <i>l'arrière plan défile avec le texte</i> fixed - <i>l'image de fond reste fixe (sauf avec Netscape)</i> body { background-attachment : fixed }
background-color	<i>Nom d'une couleur ou Code hexa : RRGGBB</i> p { background-color : #C0C0C0 }
background-image	url(image/graphique.gif) body { background-image : url(images/bg73.jpg) }
background-position	top, center, bottom - left, center, right body { background-position : top left }
background-repeat	repeat ( <i>par défaut</i> ) , no-repeat , repeat-x , repeat-y body { background-repeat: no-repeat }
background	<i>regroupement des 5 valeurs présentées ci-dessus</i> body { background: url(images/bg100.jpg) fixed repeat-y }
border-color	<i>Nom d'une couleur ou Code hexa : #RRGGBB</i>
border-left	<i>en pixels ou en pourcents</i>
border-right	"
border_top	"
border-bottom	"
border-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border-width	medium, thick <i>ou en pixels</i>
border-left-width	"
border-right_width	"
border_top_width	"
border_bottom_width	"
color	<i>Nom d'une couleur ou Code hexa : #RRGGBB</i>

font-family	<p><i>Donne par ordre de priorité la liste des polices souhaitées</i>          Arial, Verdana, "Courier new", Helvetica, "Times New Roman", Verdana, ...  <i>avec en dernier recours les noms génériques pour désigner des familles de polices</i>          cursive, serif, sans-serif, fantasy, monospace          body { font-family : Verdana, Arial, Courier, sans-serif }</p>
font-size	<p><i>Taille absolue</i>          xx-small, x-small, small, medium, large, x-large, xx-large  <i>ou taille relative</i> larger, smaller          cite { font-size: smaller }  <i>ou taille exprimée en</i> %, cm, in, pt, px          p { font-size: 12 pt }          blockquote { font-size: 150% }</p>
font-style	<p><i>Droit ou penché (qui se dit indifféremment "italic" ou "oblique")</i>          normal, italic, oblique          h1, h2, h3 { font-style: italic }</p>
font-variant	<p>normal <i>ou</i> small-caps <i>qui signifie</i> PETITES MAJUSCULES          h3 { font-variant : small-caps }</p>
font-weight	<p>normal, bold <i>pour spécifier l'épaisseur des traits</i>          lighter, bolder <i>pour un modif nuancée par rapport à ce qui précède</i>          body { font-weight: normal } /* 400 */  <i>Spécification chiffrée de l'épaisseur ( normal = 400, bold = 700)</i>          100, 200, 300, 400, 500, 600, 700, 800, 900          H1 { font-weight: 700 } /* Gras */</p>
font	<p>Permet de combiner en une ligne les 5 propriétés décrites ci-dessus          p { font : Verdana, sans-serif 14 pt italic }          h1 { font : Arial large small-caps }</p>
height	%, cm, in, pt, px
letter_spacing	%, cm, in, pt, px
line-spacing	%, cm, in, pt, px
list-style	<i>regroupe les propriétés des listes (voir ci-dessous)</i>
list-style-image	<i>remplace les puces par des images</i>
list-style-position	<i>puces à l'intérieur du texte</i> inside, outside
list-style-type	<p><i>type de puces :</i> disc, circle, square          ul { list-style-type: circle }  <i>ou type de numérotation :</i>          decimal, lower-roman, upper-roman, lower-alpha, upper-alpha          ol { list-style-type: upper-roman }          ol { list-style-type: lower-alpha }</p>

margin	en pixel ou en pourcents table { margin : 2em } /* Marge autour des tables de 2 fois la hauteur de la fonte */
margin-left	marges à gauche, à droite, au dessus ou en dessous et données en pixel ou en pourcents p { margin-left : 10 px } /* Marge de 10 pixels à gauche des § */
margin-right	"
margin-top	"
margin-bottom	"
padding-left	en pixel ou en pourcents
padding-right	"
padding-top	"
padding-bottom	"
text_align	left, center, right, justify
text_decoration	blink, none, undeline, overline, line-through
text-indent	<i>retrait d'un bloc de texte spécifié en inches, en cm ou en pixels</i> ...in, ...cm, ...px
text-transform	uppercase, lowercase, none <i>ou capitalize (=La Première Lettre En Majuscule) ,</i>
width	%, cm, in, pt, px
white-space	normal, pre, nowrap
word-spacing	%, cm, in, pt, px