

Chapitre 4

Les CSS : Cascading Style sheets

Chapitre 3 : Les CSS : Cascading Style sheets

Avantages

- dissocier le contenu d'un document de sa forme
- un style, appliqué à toutes les pages d'un site, assure la cohérence graphique de ce site ;
- la maintenance d'un site en est facilitée, un changement de charte graphique ne portera que sur la feuille de style et la modification sera naturellement propagée à toutes les pages ;
- plusieurs mises en forme différentes peuvent être proposées pour un même document sans que celui-ci ne soit répliqué en autant d'exemplaires ;
- le code HTML est lisible et facilement modifiable, sans que l'on ait à penser en même temps au contenu et à son apparence ;

Avantages

- les pages, dégagées des balises de mise en forme, sont plus légères, donc plus rapides à circuler sur un réseau (la feuille de style est, le plus souvent, très légère et ne sera chargée qu'une seule fois pour l'ensemble des documents qui l'utilisent) ;
- les balises donnent une indication sur la nature du contenu et cette indication pourra être utilisée par un moteur de recherche.
- Palier certaines insuffisances du langage Html (contrôle des polices, contrôle de la distance entre les lignes, contrôle des marges et des indentations et ainsi augmenter la créativité des écrivains du Web.

Définition des styles

- **Syntaxe :**

*Sélecteur de balise { propriété de style: valeur;
propriété de style: valeur }*

- **Exemple :**

H3 { font-family: Arial;
font-style: italic }

Remarques :

- On peut attribuer plusieurs valeurs à une même propriété. Dans ce cas, on séparera les différentes valeurs par des virgules.

H3 { font-family: Arial, Helvetica, sans-serif }

- On peut attribuer un même style à plusieurs balises (séparées par des virgules).

Exemple : H1, H2, H3 { font-family: Arial; font-style: italic }

Trois localisations possibles pour les définitions de styles

- **Style en ligne** : en tant qu'attribut des balises :
A l'intérieur de `<body>` `</body>`, dans une balise, avec l'attribut `STYLE`
- **Style du document**: déclaré dans l'en-tête, c'est-à-dire au sein des balises `<HEAD>` et `</HEAD>`
 - Pour une page HTML
- **Style externe** : déclaré dans un fichier à part dont l'extension est `.css`
 - Pour un ensemble de pages, dans un fichier externe

Notion de « cascade »

- Elles sont appelées « *feuilles de style en cascade* » (en anglais « **Cascading Style Sheets** ») car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade.
- **Ordre de priorité :**
- En cas de concurrence entre plusieurs déclarations s'appliquant à un même objet, l'ordre de priorité est le suivant :

Style en ligne > Style du document > Style externe

Règle de priorité appliquer pour la présentation du document le style le plus proche de l'élément.

Définition du style : A l'intérieur des balises

<HEAD></HEAD>

<STYLE TYPE="text/css"> <!-- La ou les feuille(s) de style --> **</STYLE>**

La balise <STYLE> avertit le navigateur que l'on va utiliser des feuilles de style.

La balise Html de commentaires <!-- ... --> empêche que les browsers qui ne connaissent pas les feuilles de style, tentent d'interpréter ces instructions.

L'attribut type="text/css" informe que ce qui suit est du texte et qu'il s'agit de cascading style sheets (css)

Définition du style : A l'intérieur des balises

<HEAD></HEAD>

Exemple :

```
<html>
```

```
<head>
```

```
<title>aa</title>
```

```
<STYLE type= "text/css" >
```

```
<!--
```

```
H1 {font-family: Arial;  
font-style: italic;  
color: green}
```

```
H2 {font-family: verdana;  
color: red}
```

```
-->
```

```
</STYLE>
```

```
</head>
```

```
<body>
```

```
<h1>Faculté des Sciences </h1>
```

```
<h2>Faculté des Sciences </h2>
```

```
</body>
```

```
</html>
```



Définition du style : Fichier externe

<HTML> <HEAD>

**avertit le browser
qu'il faudra réaliser
un lien**

<LINK rel = "stylesheet" type = "text/css" href = "monstyle.css">

</HEAD>

**l'information est du
texte et du genre
cascading style
sheets (css).**

**précise qu'il y
trouvera une feuille
de style externe.**

**le chemin d'accès
et le nom du fichier
à lier**

Définition du style : Fichier externe

Définition de monstyle.css

```
body,p,ul,li,td {  
font-size : 10pt;  
font-family : Verdana,  
Arial, Helvetica,  
Geneva, sans-serif;  
color : black;  
background-color :  
white;  
}
```

```
h1 {  
font-size : 20pt;  
font-family : Verdana,  
Arial, Helvetica,  
Geneva, sans-serif;  
color : navy;  
text-align: center;  
}
```

```
h2 {  
font-size : 14pt;  
font-family : Verdana, Arial, Helvetica,  
Geneva, sans-serif;  
color : blue;  
}
```

```
a:link {color: green; text-  
decoration:underline;}
```

```
a:visited {color: gray; text-  
decoration:underline;}
```

Définition du style : A l'intérieur des balises

`<Body></Body>`

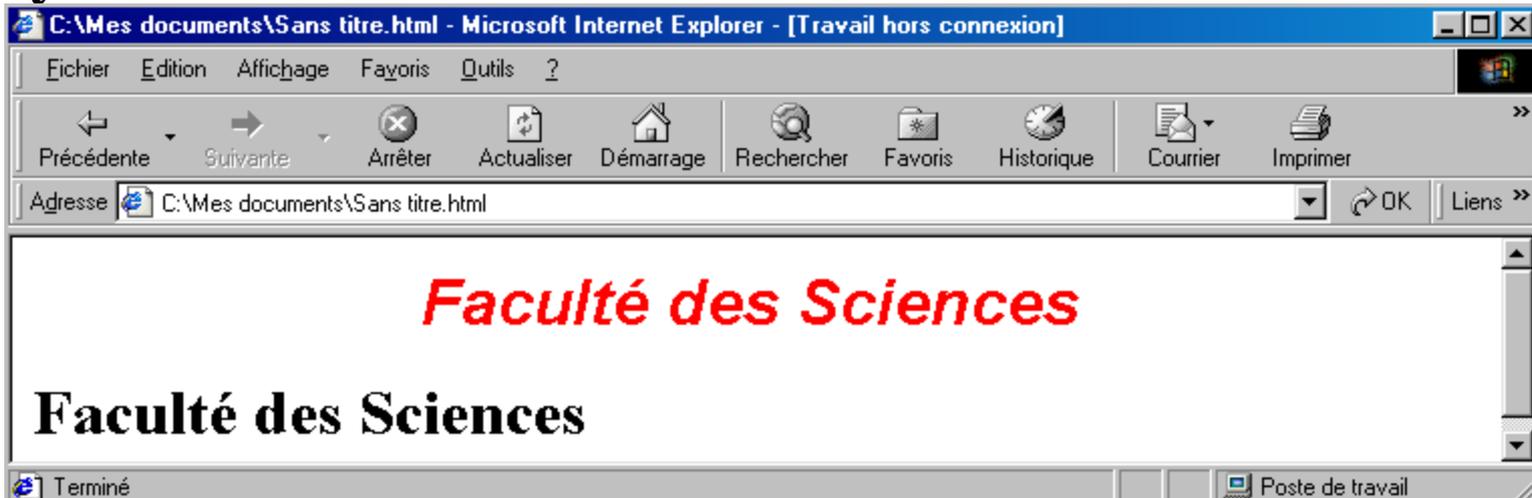
Exemple :

`<body>`

`<h1 style="font-family: Arial; font-style: italic; color: red ; text-align: center "> Faculté des Sciences </h1>`

`<h1>Faculté des Sciences </h1>`

`</body>`



peu conforme à l'esprit des feuilles de style (définir un style déterminé valable pour la globalité du document)

Notion de classe

le concept de classe sert à affecter des styles différents à des mêmes balises.

Syntaxe :

*.Nom-de-la-classe {propriété de style: Valeur;
propriété de style: Valeur ...}*

Notion de classe

- **Exemple :**

- **Définition de la classe**

```
.Style1 { font-size : 20pt;  
font-family : Verdana;  
color : navy;  
text-align: center;  
}
```

```
.Style2 { color: green; }
```

```
.Style3 { color: blue; }
```

- **Appel de la classe Dans le document HTML**

```
<P class="style1"> Texte </P>
```

```
<P class="style2"> Texte </P>
```

```
<H1 class="style3"> Texte </H1>
```

Pseudo-classe

Un ensemble d'éléments qui répondent à un même critère de contexte forme \Rightarrow Applicable concrètement à la balise **A**
Celle ci peut connaître **plusieurs contextes de formes** selon que le lien est inactif, visité, ou en train d'être visité. \Rightarrow on peut définir alors pour chacun des états de la balise une mise en forme particulière.

a:link : couleur des liens qui n'ont pas été visités

a:visited : couleur des liens visités.

a:hover, change l'apparence du lien quand l'utilisateur désigne un élément avec le pointeur de sa souris, sans l'activer.

Pseudo-classe

a:active, change l'apparence du lien quand l'utilisateur active un élément. Par exemple, entre le moment où l'utilisateur presse le bouton de la souris et le relâche.

a:focus, qui s'applique quand un élément reçoit une couleur de fond, par exemple.

Exemple:

```
A:link {font-family:Arial; color:"#0099FF"; font-size:12px; cursor:text;}
```

```
A:visited {font-family:Arial; color:blue; font-size:12px; cursor:text;}
```

```
A:active {color:red;}
```

```
A:hover {color:red; text-decoration:none;}
```

Les ID

Les ID servent à remplacer les classes lorsqu'il y a du JavaScript dans la page, car le JavaScript emploie lui aussi la syntaxe `nom.nom` (`nom-point-nom`).

La syntaxe est alors:

```
#nom_ID { propriété de style: Valeur;  
propriété de style: Valeur ... }
```

On l'appellera de la manière suivante:

```
<BALISE ID="nom_ID" > ... </BALISE>
```

On ne peut faire appel qu'à un seul même ID par page!

Déf :

```
#essentiel{ ... }
```

Appel

```
<P id=essentiel> texte </p>
```

Les balises SPAN et DIV

Il faut pouvoir dans un même paragraphe appliquer des styles différents à des morceaux de texte, c'est à cela que servent les balises `` et `<DIV>`

•La balise ``

La balise `` sert à appliquer des styles à des morceaux de paragraphe. Elle s'utilise aussi bien avec *ID* qu'avec *CLASS*.

```
<SPAN class=Nom_de_la_classe> Texte </SPAN>
```

•La balise `<DIV>`

L'idée est la même pour la balise `DIV`, cependant au lieu de s'appliquer à quelques mots dans un paragraphe, elle s'applique sur plusieurs paragraphes.

```
<DIV class=Nom_de_la_classe> paragraphes </DIV>
```

La mise en page

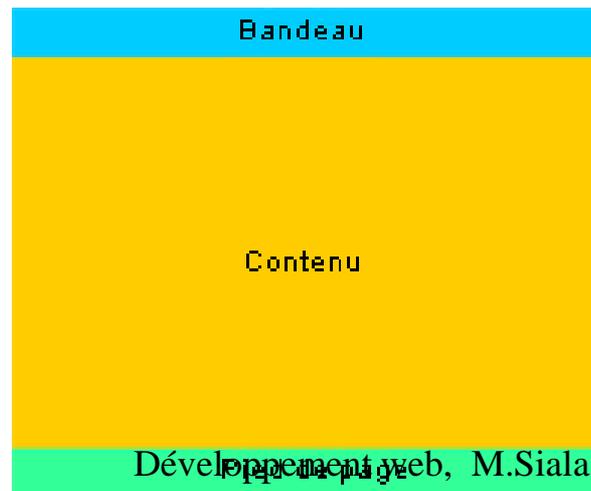
```
div {
    text-align:center;
}
div#bandeau {
    width:600px;
    height:50px;
    background-color:#00CCFF;}
div#contenu {
    width:600px;
    height:400px;
    background-color:#FFCC00;}
div#pied_page {
    width:600px;
    height:50px;
    background-color:#33FF99;}

```

La mise en page

```
<body>  
  <div id="bandeau" >bandeau</div>  
  <div id="contenu" >contenu</div>  
  <div id="pied_page" > pied de page</div>  
</body>
```

<http://www.yoyodesign.org/doc/w3c/css2/visuren.html#float-position>



La mise en page

```
div {text-align:center;}
div#bandeau {width:600px; height:50px;background-color:#00CCFF;}
div#menu {
    float:left;
    width:100px;
    height:400px;
    background-color:#FF6699;}
div#contenu {
    float:left;
    width:500px;
    height:400px;
    background-color:#FFCC00;}
div#pied_page {clear:both;
    width:600px;
    height:50px;
    background-color:#33FF99;}
```

La mise en page

```
<body>
```

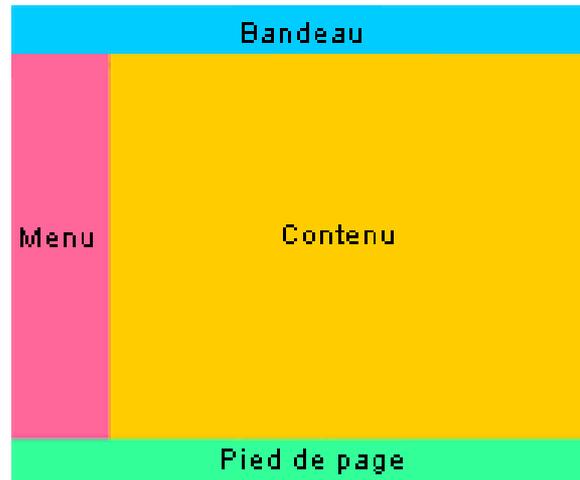
```
  <div id="bandeau">Ceci est le bandeau</div>
```

```
  <div id="menu">Ceci est le menu</div>
```

```
  <div id="contenu">Ceci est le contenu</div>
```

```
  <div id="pied_page">Ceci est le pied de page</div>
```

```
</body>
```



La mise en page

- Pour afficher deux div l'un à côté de l'autre, il va donc nous falloir utiliser une autre propriété. C'est la propriété float
-
- Il y a une propriété qui va permettre de dire qu'on veut que notre élément se remette à se comporter normalement (qu'il se positionne en dessous du reste). C'est la propriété clear.
-

La mise en page

```
div#menuhaut
```

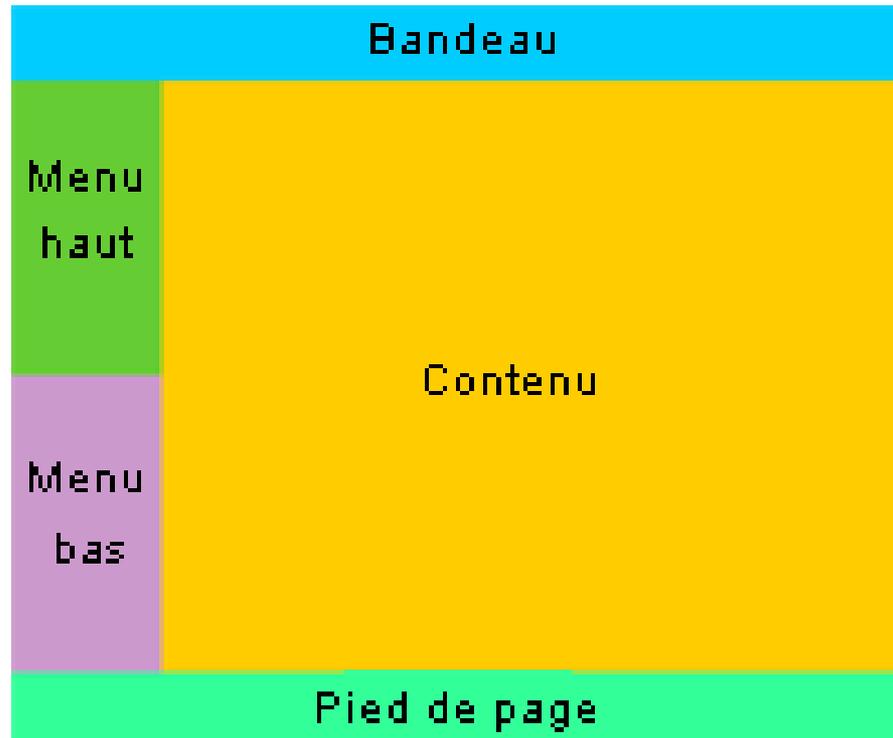
```
{ width:100px;  
height:200px;  
background-color:#66CC33; }
```

```
div#menubas
```

```
{  
width:100px;  
height:200px;  
background-color:#CC99CC; }
```

La mise en page

- `<div id="menu">`
 `<div id="menuhaut">Menu haut</div>`
 `<div id="menubas">Menu bas</div>`
`</div>`



Notion de boite

- Le w3c définit un élément boite comme une zone rectangulaire constituée :
 - D'un contenu
 - D'une marge intérieure (padding)
 - D'une bordure
 - D'une marge extérieure (margin)

