

CSS débutants

par Pascale Lambert ([Mammouthland](#))

Date de publication : 14/04/2005

Dernière mise à jour : 14/02/2008

Mon intention n'est pas de faire ici un cours de référence sur le css, mais juste de regrouper les différentes astuces de code que je publie de temps en temps sur le weblog.

- I - Introduction
 - I-A - CSS, kezaoko ?
 - I-B - Balises html de base
- II - Feuille de style de base
 - II-A - Enregistrement
- III - Sélecteurs "class" et "id"
 - III-A - Le sélecteur class
 - III-B - Le sélecteur id
- IV - Tableaux à bordures fines avec les CSS
 - IV-A - Premier exemple
 - IV-B - Deuxième exemple
 - IV-B-1 - Accessibilité
 - IV-C - Troisième exemple : le tableau sans tableau !
- V - Boutons CSS
- VI - Menu horizontal en CSS type "boutonnière"
- VII - Puces en gifs avec les CSS
 - VII-A - Version classique
 - VII-B - Variante Made-in Sam
- VIII - Pseudo-éléments CSS :before ; :after
 - VIII-A - Exemple
- IX - Mise en forme CSS d'un formulaire
 - IX-A - Exemple
 - IX-B - Compléments
 - IX-B-1 - Rollover
 - IX-B-2 - Focus
- X - Faire apparaître caractères ou images au survol de la souris (rollover)
 - X-A - Première méthode
 - X-B - Deuxième méthode
 - X-C - Troisième méthode
- XI - Zoomer une image avec les CSS
 - XI-A - Première technique
 - XI-B - Seconde technique
 - XI-C - Variante
 - XI-D - Compléments d'informations
- XII - Zoomer une image façon thumbnail en CSS
 - XII-A - Première technique
 - XII-B - Seconde technique
- XIII - Image de fond en CSS
 - XIII-A - Code CSS de base
 - XIII-B - Fond de page fixe
 - XIII-C - Positionner l'image
 - XIII-C-1 - Exemple positionnement 2
 - XIII-C-3 - Exemple positionnement 3
 - XIII-C-4 - Exemple positionnement 4
 - XIII-C-5 - Remarques
- XIV - Plusieurs images de fond grâce aux CSS
- XV - Blockquote et retraits
- XVI - Balise <object> et molécules Rasmol

I - Introduction

I-A - CSS, kezaoko ?

Le design d'un site évolue toujours au fil du temps. Le problème, lorsqu'on n'utilise pas de feuilles de style, c'est qu'il faut reprendre toutes les pages une à une pour modifier une police de caractère ou une couleur de fond...

Avec les "Cascading Style Sheet" (CSS), ce lourd handicap est résolu.

C'est dans la feuille de style que l'on va déclarer toute la mise en forme des pages : la couleur de fond, les polices de caractère, leurs couleurs, etc.

Celle-ci sera liée à chaque page html. Ainsi, lorsqu'on en modifiera un élément, cela se répercutera immédiatement sur toutes les pages web. Puissant et pratique ;-))

Mais cela ne s'arrête pas là. Outre les couleurs et les polices, on peut aussi changer radicalement l'apparence d'un site en abandonnant la mise en page à l'aide de tableaux. Soyons clair : un tableau sert à présenter des données tabulaires -point barre-

Pour une mise en page souple et cohérente, on utilisera des "blocs" (appelés aussi "boîtes" ou "calques"), qui ont l'avantage de pouvoir être déplacés de gauche à droite, ou de haut en bas grâce aux CSS.

Pour vous en convaincre, rendez-vous sur la [page d'accueil](#) de mammouthland et testez les différentes mises en pages à disposition grâce au menu déroulant "choisir son habillage" : ce ne sont pas les pages html qui changent, mais uniquement les feuilles de style.

Si vous êtes intéressés, voire convaincus, voyez les liens présents en bas du tutoriel sur les tableaux qui vous aideront à créer des pages sans tables.

Et ce n'est pas plus difficile que d'utiliser des tableaux contrairement à ce que certains de mauvaise foi voudraient bien faire croire !

Cela dit, même si on utilise un logiciel wysiwyg tel que Dreamweaver, il convient tout de même de connaître quelques balises html, car ce sont celles-ci qui vont être indiquées dans la feuille de style.

I-B - Balises html de base

Document

Document HTML
`<html></html>`

Corps du document
`<body></body>`

Titres

Titre 1er niveau
`<h1></h1>`

Titre 2ème niveau

```
<h2></h2>
```

Titre 3ème niveau

```
<h3></h3>
```

Et ainsi de suite jusqu'à 6.

Éléments de texte

Paragraphe

```
<p></p>
```

Liste à puce

```
<ul></ul>
```

Liste numérotée

```
<ol></ol>
```

Élément d'une liste

```
<li></li>
```

Lien hypertexte

```
<a href=""></a>
```

Tableau

Tableau

```
<table></table>
```

Légende de ligne ou de colonne

```
<th></th>
```

Ligne du tableau

```
<tr></tr>
```

Colonne du tableau

```
<td></td>
```

Positionnement

Définition de bloc

```
<div></div>
```

II - Feuille de style de base

Fonctionne avec :

- FireFox
- IE 5.x et +
- NS 4.x et +
- Mozilla 1.x
- Opéra 5.x et +
- Safari

Attributs utilisés :

- background-color
- color
- font-family ; font-size
- padding
- text-align ; text-decoration

Cette feuille de style (encore une fois, de base) donnera une page à fond blanc, avec une police de caractère par défaut Verdana noire, un titre de premier niveau bleu marine centré, un titre de deuxième niveau rouge décalé de 15 pixels, des liens passant du [vert](#) au [gris](#) avec un petit effet **rouge non souligné** au passage de la souris.

A noter que les couleurs sont ici sous la forme "white, black, blue" etc... Pour plus de lisibilité mais qu'il convient plutôt d'utiliser les valeurs hexadécimales qui vont de #000000 (noir) à #ffffff (blanc)

Les tailles de caractères sont ici exprimées en %, mais on peut aussi utiliser **em** ou **px** qui sont aussi des unités de longueurs relatives à privilégier, pour des raisons d'homogénéité d'affichage, aux unités absolues telles que les **pt** ou les **pc**

Code CSS

```
html,body
{
  background-color : white;
}

p,ul,li,td
{
  font-size : 100%;
  font-family : Verdana, Arial, Helvetica, Geneva, sans-serif;
  color : black;
  background-color : white;
}

h1
{
  font-size : 200%;
  font-family : Verdana, Arial, Helvetica, Geneva, sans-serif;
  color : navy;
  text-align: center;
}
```

Code CSS

```
h2
{
  font-size : 150%;
  font-family : Verdana, Arial, Helvetica, Geneva, sans-serif;
  color : red;
  padding-left:15px;
}

a:link {color: green; text-decoration:underline;}
a:visited {color: gray; text-decoration:underline;}
a:hover {color:red; text-decoration:none;}
```

II-A - Enregistrement

Enregistrer cette page "style.css", et mettre dans l'en-tête de la page html (entre les balises **<head></head>**) :

Code HTML

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

Pour éviter les mauvaises surprises avec le (presque) feu NS 4.x on peut aussi à la place (cette option désactive la feuille de style avec ce navigateur) :

Code HTML

```
<style type="text/css" media="screen">@import "style.css";</style>
```

Le tour est joué !

Si vous désirez générer simplement une feuille de style un peu plus fournie, un générateur de feuille de style est à **disposition**.

III - Sélecteurs "class" et "id"

Fonctionne avec :

- FireFox
- IE 5.x et +
- NS 6/7
- Mozilla 1.x
- Opéra 5.x et +
- Safari

Attributs utilisés :

- background-color
- float
- width
- margin
- text-align

Maintenant que la **feuille de style** de base n'a plus de secrets pour vous, il est temps de passer à la vitesse supérieure. L'apprentissage de l'utilisation des sélecteurs **class** et **id** va vous ouvrir bien d'autres horizons...

A quoi sert un sélecteur ? Tout simplement à définir des styles particuliers que l'on voudra reproduire de façon ponctuelle ou récurrente dans la page (X)HTML.

III-A - Le sélecteur class

Prenons un exemple : les pages web sont souvent ponctuées d'ancres permettant de remonter en haut de page, et celles-ci sont souvent placées à droite de la page. Il va donc falloir déclarer un style spécial pour cette mise en forme spécifique, et le sélecteur **class** va nous y aider. A noter que dans la feuille de style, le nom du sélecteur **class** est toujours précédé d'un point.

Source CSS

```
.haut
{
  text-align:right;
}
```

Source (X)HTML

```
<p class="haut">
  <a href="#haut">Haut de page</a>
</p>
```

Et voilà, et cette mise en forme peut-être répétée autant de fois que besoin dans la page.

III-B - Le sélecteur id

Le sélecteur **id** a presque la même fonction, à la différence importante qu'on ne peut l'utiliser qu'**une seule fois dans la page**, contrairement au sélecteur **class**. C'est pour cela qu'il est plutôt utilisé à la mise en page qu'à la mise en forme de caractères.

Exemple : construisons une section de page à deux colonnes. L'une des deux colonnes servira pour un menu de 100 pixels de large flottant à gauche, l'autre pour un contenu placé à 110 pixels du bord gauche (pour ne pas empiéter sur le menu bien sûr). Cette fois, ce n'est pas d'un point qu'il faudra faire précéder le nom du sélecteur **id**, mais d'un #.

Source CSS

```
#menu
{
  background-color: silver;
  width: 100px;
  float: left;
}

#contenu
{
  margin-left: 110px;
}
```

Source (X)HTML

```
<h1 id="haut">Exemple des sélecteurs "class" et "id"</h1>

<div id="menu">
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
    <li>item 4</li>
    <li>item 5</li>
  </ul>
</div>

<div id="contenu">
  <p>
    Lorem ipsum dolor sit amet,
    consectetur adipiscing elit,
    sed diam nonummy nibh euismod
    tincidunt ut laoreet dolore
    magna aliquam erat volutpat.
    Ut wisi enim ad minim veniam,
    quis nostrud exerci tation
    ullamcorper suscipit lobortis
    nisl ut aliquip ex ea commodo consequat.
    Duis autem vel eum iriure dolor
    in hendrerit in vulputate velit
    esse molestie consequat,
    vel illum dolore eu feugiat
    nulla facilisis at vero eros
    et accumsan et iusto odio
    dignissim qui blandit praesent
    luptatum zzril delenit augue duis
    dolore te feugait nulla facilisi.
  </p>

  <p class="haut">
    <a href="#haut">Haut de page</a>
  </p>
</div>
```

Source (X)HTML

```
</p>  
</div>
```

Il n'aura pas échappé à votre sagacité qu'un **id** non déclaré dans les styles est venu se glisser dans la balise de titre. C'est le deuxième effet **id** ! En effet, cet attribut peut aussi servir d'ancre. Ainsi, le lien interne de haut de page y est rattaché directement. Cela explique aussi pourquoi on ne peut utiliser qu'un seul nom de id par page. s'il y avait deux **id="haut"** dans la page, où irait le lien ?! hein ? je vous le demande !

 **Tester l'exemple**

IV - Tableaux à bordures fines avec les CSS

Comment faire un tableau à bordures fines !? Sans les feuilles de style, impossible. Rien à faire à part des grosses bordures saillantes et peu élégantes. Mais avec une petite dose de CSS, c'est fait en un rien de temps.

Fonctionne avec :

- FireFox
- IE 5.x et +
- NS 6/7
- Mozilla 1.x
- Opéra 5.x et +
- Safari

Propriétés utilisées :

- background-color
- border-color ; border-style ; border-width
- float
- height
- left
- margin
- padding
- position
- top
- width

IV-A - Premier exemple

Un habillage très rapide de la table (**table**) et des cellules (**td**) va déjà donner un aperçu de ce que l'on peut faire.

Indiquez dans votre **feuille de style** externe le code suivant

Source CSS

```
table
{
  border-width:1px;
  border-style:solid;
  border-color:black;
  width:50%;
}

td
{
  border-width:1px;
  border-style:solid;
  border-color:red;
  width:50%;
}
```

Source CSS

Vous noterez que, grâce à l'attribut **width**, les tailles du tableau et des cellules sont aussi définies. Cela va permettre d'épurer le code (X)HTML qui suit.

Dans votre page html, indiquez ceci :

Source (X)HTML

```

<table>
  <tr>
    <td><p>Oh qu'il est beau !</p></td>
    <td><p>Ce jouli tableau à traits fins !!</p></td>
  </tr>
</table>

```

Cela doit vous donner :

Oh qu'il est beau !	Ce jouli tableau à bordures fines !!
---------------------	--------------------------------------

Pour fusionner les bordures de tour de tableau et de cellules, il suffira de rajouter, dans les styles de **table**, l'élément suivant : **border-collapse:collapse**, ce qui donnera l'effet suivant :

Oh qu'il est beau !	Ce jouli tableau à bordures fusionnées...
---------------------	---

 Pour centrer verticalement le texte dans la cellule, rajouter dans le code CSS du **td** :

Code CSS

```
vertical-align:middle;
```

IV-B - Deuxième exemple

J'en vois d'ici qui disent : "oui, mais je n'en veut pas du trait noir du tour !! ... et puis d'abord, je veux ma cellule de gauche bordée de noir et avec un fond gris, et celle de droite bordée de bleu."

Pas de problèmes !

Oh la belle cellule grise !	Oh la belle bleue !!
-----------------------------	----------------------

Source CSS

Source CSS

```
.grise
{
  border-width:1px;
  border-style:solid;
  border-color:black;
  background-color:silver;
}

.blue
{
  border-width:1px;
  border-style:solid;
  border-color:blue;
}

.none
{
  border-style:none;
}
```

C'est avec l'attribut "class" l'appel aux styles va se faire :

Source (X)HTML

```
<table class="none">
  <tr>
    <td class="grise">Oh la belle cellule grise !</td>
    <td class="blue">Oh la belle bleue !!</td>
  </tr>
</table>
```

Vous remarquez donc que les styles ".grise" et ".blue" sont appelés dans les cellules du tableau par la syntaxe **class="grise"** et **class="blue"**. Avec ce système, les combinaisons sont infinies...

Le style des bordures peut-être "**solid**" comme dans cet exemple, mais aussi en relief grâce à "**outset**" ou "**inset**", en pointillé "**dotted**"...

On peut aussi avoir les 4 côtés différents puisque chaque bord peut être détaillé : **border-style-left**, **border-color-right**, **border-width-top** etc.

Possibilités infinies je vous dis !



Le code CSS des bordures peut être allégé en indiquant dans l'ordre la taille, le style et la couleur. Cela donnerait donc pour les exemples ci-dessus :

Code CSS

```
border : 1px solid black;
```

IV-B-1 - Accessibilité

Les tableaux servent à présenter les données tabulaires. Ca paraît évident comme ça, mais cela mérite pourtant d'être rappelé car de nombreux sites s'en servent encore pour la mise en page alors que ce n'est pas nécessaire, voire contre productif.

Afin que l'**accessibilité** soit meilleure, il existe quelques règles à mettre en place. En voici trois importantes et faciles :

- 1 Indiquer le **summary** à l'intérieur de la balise **table**, qui fait office de légende pour expliquer le contenu du tableau.
- 2 Utiliser la balise **caption** pour indiquer un titre au tableau
- 3 Utiliser les balises **th** pour les cellules d'en-têtes. A noter que par défaut le contenu des balises **th** sont centrées et en gras.

Un petit exemple vite fait :

Code (X)HTML

```
<table summary="répartition, en pourcentage, des 5 navigateurs les plus utilisés pour visiter CSS débutants">
<caption>Top 5 des navigateurs utilisés sur CSS débutants en décembre 2007</caption>
<tr>
  <th>Firefox</th>
  <th>Explorer</th>
  <th>Mozilla</th>
  <th>Safari</th>
  <th>Opéra</th>
</tr>
<tr>
  <td>49,6 %</td>
  <td>39,4 %</td>
  <td>5,0 %</td>
  <td>2,4 %</td>
  <td>0,9 %</td>
</tr>
</table>
```

Code CSS

```
table
{
  border-collapse:collapse;
  width:90%;
}

th, td
{
  border:1px solid black;
  width:20%;
}

td
{
  text-align:center;
}

caption
{
  font-weight:bold
}
```

Top 5 des navigateurs utilisés sur CSS débutants en décembre 2007

Firefox	Explorer	Mozilla	Safari	Opéra
49,6 %	39,4 %	5,0 %	2,4 %	0,9 %

Pour la mise en forme de tableaux plus complexes et pour plus de critères d'accessibilité, consulter ces différentes pages :

- [Tableaux CSS - OpenWeb](#)
- [Critères accessiweb](#)
- [Générateur de tableaux accessibles](#)

IV-C - Troisième exemple : le tableau sans tableau !

 *En attendant MSIE 8, cette option ne fonctionne pas encore avec Internet Explorer !*

Grâce à la propriété **display** et l'attribut **table-cell**, on peut aussi simuler une présentation tabulaire sans faire de tableau.

Mettons que je veuille présenter le top 5 de mes navigateurs non pas sous forme de tableau mais sous forme de liste :

- Firefox : 49,6 %
- Internet Explorer : 39,4 %
- Mozilla : 5,0 %
- Safari : 2,4 %
- Opéra : 0,9 %

Mais je voudrais que les pourcentages soient alignés verticalement...

Pour cela, on peut encadrer les éléments que l'on veut aligner par la balise **span** et lui attribuer un **table-cell** ce qui va permettre de lui affecter une largeur fixe comme un élément de type "block", chose impossible par défaut.

Petit écueil néanmoins au niveau de l'alignement des images de puces : elles vont se trouver décalées par rapport au texte. Mais cela peut-être réglé par la méthode "Made in Sam" décrite dans le tutoriel sur les **puces**.

Code (X)HTML

```
<ul>
  <li><span>Firefox :</span> <span>49,6 %</span></li>
  <li><span>Internet Explorer :</span> <span>39,4 %</span></li>
  <li><span>Mozilla :</span> <span>5,0 %</span></li>
  <li><span>Safari :</span> <span>2,4 %</span></li>
  <li><span>Opéra :</span> <span>0,9 %</span></li>
</ul>
```

Code CSS

```
li span
{
  display:table-cell;
  width:130px;
}

li
{ /*pour l'alignement des puces*/
  list-style-image:none;
  list-style-type:none;
}
```

Code CSS

```
background: url(pucegrise.gif) no-repeat 0% 50%;  
padding-left:20px;  
}
```

● Firefox :	49,6 %
● Internet Explorer :	39,4 %
● Mozilla :	5,0 %
● Safari :	2,4 %
● Opéra :	0,9 %

Pardon ? Vous dites ? *"Oui mais les nombres, ça doit s'aligner sur la droite ?"*. Mais vous êtes pire que moi dans le souci du détail !!

Ok, vos désirs sont des ordres. On va utiliser les **"combinateurs d'enfants adjacents"** (E + F) pour arranger cela et qui va permettre de ne styler que le second **span** de chaque item. On va donc en fixer la largeur à 50 pixels et aligner le texte sur la droite en rajoutant ceci au code CSS :

Code CSS

```
li span + span  
{  
width:50px;  
text-align:right;  
}
```

● Firefox :	49,6 %
● Internet Explorer :	39,4 %
● Mozilla :	5,0 %
● Safari :	2,4 %
● Opéra :	0,9 %

V - Boutons CSS

Fonctionne avec :

- FireFox
- IE 5.x et +
- NS 6/7
- Mozilla 1.x
- Opéra
- Safari

Attributs utilisés :

- background-color ; background-image
- border-color ; border-style ; border-width
- color
- float
- font-weight
- height
- padding
- text-align ; text-decoration
- width

L'exemple ci-après permet de faire un bouton dont le fond change d'aspect au passage de la souris. Cet effet, largement répandu, utilise souvent un JavaScript ou un applet java lourd à charger.

Ici, la légèreté est de mise...

 [Voir le résultat](#)

Les styles (source css) sont à enregistrer dans une feuille de style externe selon la procédure indiquée [ici](#)

Ces styles seront appelés dans la page html (source html) par l'attribut "class" placés à l'intérieur des balises <p> et <a>

Source HTML

```
<p class="bouton">
  <a href="cours3.php" class="bouton">Cliquez ici !</a>
</p>
```

Source CSS

```
a.bouton:link
{
  width:150px;
  height:30px;
```

Source CSS

```
text-decoration:none;
color:white;
text-align:center;
font-weight:bold;
background-color:#000080;
padding:5px
}

a.bouton:visited
{
width:150px;
height:30px;
text-decoration:none;
color:white;
text-align:center;
font-weight:bold;
background-color:#000080;
padding:5px
}

a.bouton:hover
{
width:150px;
height:30px;
text-decoration:none;
color:white;
text-align:center;
font-weight:bold;
background-color:#0000FF;
background-image:url(aqua.jpg);
padding:5px
}

.bouton
{
text-align:center;
padding:5px;
}
```

On peut même rajouter, sans plus de poids, un effet de relief et une couleur finale différente une fois que le site a été visité

 **Voir le résultat**

Source HTML

```
<p class="bouton2">
  <a href="toto.html" class="bouton2">Cliquez ici !</a>
</p>
```

Source CSS

```
a.bouton2:link
{
width:150px;
height:30px;
text-decoration:none;
color:white;
text-align:center;
font-weight:bold;
}
```

Source CSS

```
background-color:#000080;
padding:5px;
border-style:outset;
border-width:2px;
border-color:silver
}

a.bouton2:visited
{
width:150px;
height:30px;
text-decoration:none;
color:white;
text-align:center;
font-weight:bold;
background-color:#0066FF;
padding:5px;
border-style:outset;
border-width:2px;
border-color:silver
}

a.bouton2:hover
{
width:150px;
height:30px;
text-decoration:none;
color:white;
text-align:center;
font-weight:bold;
background-color:#0000FF;
background-image:url(aqua.jpg);
padding:5px;
border-style:outset;
border-width:2px;
border-color:silver
}

.bouton2
{
text-align:center;
padding:5px;
}
```

...Ou encore une impression de bouton qui s'enfonce au passage de la souris en remplaçant "**border-style:outset**" par "**border-style:inset**" dans le "**a.bouton2:hover**".

 **Voir le résultat**

Petit truc tout bête à respecter. Si vous voulez que l'effet rendu par le survol du bouton par la souris perdure, même après visite, indiquez bien le **a:hover** en dernier !

En complément, un **autre tutoriel**, réalisé par la même, sur OpenWeb.

VI - Menu horizontal en CSS type "boutonnière"

Fonctionne avec :

- Firefox
- IE 6 et +
- NS 6/7
- Mozilla 1.x ; SeaMonkey
- Opéra
- Safari

Attributs utilisés :

- background-color
- border-color ; border-style ; border-width
- color
- float
- height
- list-style-type
- padding
- text-align ; text-decoration
- width

A l'inverse de la technique décrite dans [ce tutoriel](#), pour créer un menu horizontal dont chaque bouton a la même largeur quelque soit la longueur du texte, on va donner l'aspect d'un paragraphe à une liste.

Source (X)HTML

```
<ul>
  <li><a href="#">Item 1</a></li>
  <li><a href="#">Item 2</a></li>
  <li><a href="#">Item plus long</a></li>
</ul>
```

Le principe est donc de transformer cette liste à puces verticale en liste (sans puces) horizontale. Pour commencer, on va donc déclarer la liste (**ul ; list-style-type:none**) sans puces et mettre mes marges extérieures (**margin**) et intérieures (**padding**) à zéro.

Chaque item (**li**) devra néanmoins être légèrement écarté de son voisin, on va donc déclarer une marge droite à 2 pixels (**margin-left:2px**).

Vient ensuite le comportement des liens de cette liste (**ul li a**). Le **display:block** va permettre de ne pas passer à la ligne à chaque item, et pour ne pas qu'ils s'empilent les uns sur les autres, on déclare les différents élément flottants à droite **float:left**. Mais comme MSIE ne fait rien comme tout le monde, cet attribut doit être aussi déclaré pour les items seuls.

Reste à déclarer l'aspect du bouton : sa taille (**width**) ; sa couleur de fond (**background-color**) ; la couleur du texte, son aspect, et son positionnement dans le bouton (**color** ; **text-decoration** ; **text-align** ; **padding**) ; l'aspect des bordures (**border-width** ; **border-style** ; **border-color**).

Et pour figurer le tout, on définit l'aspect que prendra ce même bouton lors du survol de la souris (**ul li a:hover**).

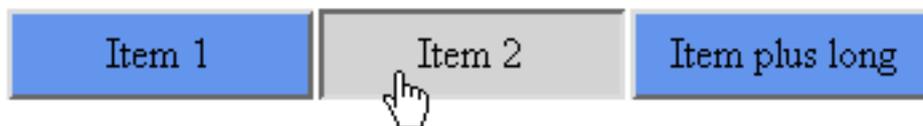
Source CSS

```
ul
{
  padding:0;
  margin:0;
  list-style-type:none;
}

li
{
  margin-left:2px;
  float:left; /*pour IE*/
}

ul li a
{
  display:block;
  float:left;
  width:100px;
  background-color:#6495ED;
  color:black;
  text-decoration:none;
  text-align:center;
  padding:5px;
  border-width:2px;
  border-style:solid;
  /*pour avoir un effet "outset" avec IE :*/
  border-color:#DCDCDC #696969 #696969 #DCDCDC;
}

ul li a:hover
{
  background-color:#D3D3D3;
  border-color: #696969 #DCDCDC #DCDCDC #696969;
}
```



 **Tester le résultat**

VII - Puces en gifs avec les CSS

VII-A - Version classique

Fonctionne avec :

- FireFox
- IE 5.x et +
- NS 6/7
- Mozilla 1.x
- Opéra 5.x et +
- Safari

Attributs utilisés :

- color
- display
- font-family ; font-size
- list-style-image ; list-style-position

Comment faire une liste à puces avec les styles dont les puces sont des images (.gifs, .png, .jpg) ?

Petite astuce que j'utilise pour le menu du [côté jardin...](#)

Indiquez dans votre **feuille de style externe** le code suivant :

Code CSS

```
li
{
  font-family: Verdana, Arial, Helvetica, Geneva, sans-serif;
  font-size: 100%;
  color: black;
  display : list-item;
  list-style-image : url(puce.gif);
  list-style-position: outside;
}
```

Code (X)HTML

```
<ul>
  <li>liste 1</li>
  <li>liste 2</li>
  <li>liste 3</li>
</ul>
```

- liste 1
- liste 2
- liste 3

VII-B - Variante Made-in Sam

L'inconvénient de méthode précédemment décrite est que les images choisies comme puces ne sont pas forcément alignées avec le texte. On le voit d'ailleurs très bien ci-dessus.

Pour palier à ce problème, Docteur **Sam** prescrit de mettre ces images en fond (**background-image:horizontal vertical**) et d'ajuster la position à l'aide de l'attribut **background-position** et d'un petit **padding-left** pour ne pas que l'image se superpose au texte.

Code CSS

```
li
{
  font-family: Verdana, Arial, Helvetica, Geneva, sans-serif;
  font-size: 100%;
  color: black;
  list-style-type:none;
  background-image:url(puce.gif);
  background-repeat:no-repeat;
  background-position:0% 65%;
  padding-left:15px;
}
```

- Item 1
- Item 2
- Item 3

La potion du Docteur Sam : c'est de la bonne, buvez-en.

VIII - Pseudo-éléments CSS :before ; :after

Fonctionne avec :

- FireFox
- NS 6/7
- Mozilla 1.x
- Opéra 6/7
- Safari

Attributs utilisés :

- content

Petite astuce css amusante et pratique, mais attention qui ne fonctionne pas encore avec IE6 ! Il ne faut donc l'utiliser pour l'instant que pour des choses peu importantes.

Avec les pseudo classes :before et :after, on peut automatiser l'inclusion de caractères ou d'images.

VIII-A - Exemple

Je désire, pour indiquer une astuce, qu'une petite image d'ampoule soit présente devant la mention "Astuce" pour la mettre plus en valeur.

De façon classique, ceci pourrait donner cela :

Source HTML

```
<p style="color:red; font-weight:bold">  
   Astuce !  
</p>
```



Mais comme je donne beaucoup d'astuces (sic !), je trouve cette opération répétitive un peu pénible, sans compter que cela alourdit le poids de la page.

C'est là que le css prend le relais.

Pour automatiser l'inclusion de l'image et, soyons fainéant jusqu'au bout, le point d'exclamation, j'inscris donc cela dans ma feuille de style :

Source CSS

```
.astuce  
{  
  color:red;
```

Source CSS

```
font-weight:bold;
background-color:transparent;
}

.astuce:before {content:url(images/lightmov.gif)}

.astuce:after {content:" ! "}
```

Puis dans la page html :

Source HTML

```
<p class="astuce"> Astuce</p>
```



Évidemment, si vous regardez ceci avec IE, vous ne voyez ni l'ampoule ni le point d'exclamation. Il vous faut au minimum Netscape 6, Opéra 6 ou Mozilla pour en profiter.

A vous de choisir : coder des lignes superflues qui encombreront la bande passante ou profiter pleinement des standards ;-).

IX - Mise en forme CSS d'un formulaire

Fonctionne avec :

- FireFox
- NS 6/7
- Mozilla 1.x
- Opéra 7 et plus
- MSIE 5.5 et plus
- Safari

Attributs utilisés :

- background-color
- border
- color
- font-family, font-size, font-weight
- width

Les formulaires sont tristounets. En voici la preuve :

Savez-vous ce que veut dire CSS ? : oui non

Si oui, les utilisez-vous plutôt :

Votre email :

Vos commentaires :

Grâce aux feuilles de style, on peut facilement égayer tout cela.

IX-A - Exemple

Pour commencer, regardons le code (X)HTML de plus près :

Source (X)HTML

```
<form action="toto.php" method="post" >

  <p>
    <label>Savez-vous ce que veut dire CSS ? : </label>
    <input type="radio" name="CSS" value="oui" checked="checked" /> oui
    <input type="radio" name="CSS" value="non" /> non
  </p>

  <p>
    <label>Si oui, les utilisez-vous plutôt : </label>
    <select name="utilise">
      <option value="toujours"> toujours</option>
      <option value="parfois"> parfois</option>
      <option value="jamais"> jamais</option>
    </select>
  </p>

  <p>
    <label for="email">Votre email :</label><br />
    <input type="text" name="email" size="20" maxlength="40" value="email" id="email" />
  </p>

  <p>
    <label for="comments">Vos commentaires :</label><br />
    <textarea name="comments" id="comments" cols="20" rows="4"></textarea>
  </p>

  <p>
    <input type="submit" value="Envoyer" />
    <input type="reset" value="Annuler" />
  </p>

</form>
```

On constate donc les présences des balises **<p>**, **<form>**, **<label>**, **<input>**, **<select>**, **<option>** et **<textarea>**.

Ce sont elles qui seront déclarées dans la feuille de style pour la mise en forme du formulaire.

Les styles (source css) sont à enregistrer dans une feuille de style externe selon la procédure indiquée **ici**.

Source CSS

```
p
{
  font-family:"trebuchet ms",sans-serif;
  font-size:80%;
}

form
{
  background-color:#F5F5F5;
  padding:10px;
  width:350px;
}

label
{
  font-family:"trebuchet ms",sans-serif;
  font-weight:bold
}
```

Source CSS

```
}  
  
input  
{  
  border:1px solid black;  
  background-color:red;  
  font-family:"trebuchet ms",sans-serif;  
  color:white;  
}  
  
select, option  
{  
  background-color:red;  
  color:white;  
}  
  
textarea  
{  
  border:1px solid black;  
  background-color:red;  
  font-family:"trebuchet ms",sans-serif;  
  color:white;  
}
```

Le résultat sera alors le suivant :

Savez-vous ce que veut dire CSS ? : oui non

Si oui, les utilisez-vous :

Votre email :

Vos commentaires :

IX-B - Compléments

IX-B-1 - Rollover

L'apparence des boutons "envoyer" et "annuler" peut être rendue plus dynamique de la façon suivante :

Source (X)HTML

```
<p>
  <input type="submit" value="Envoyer" class="bouton" />
  <input type="reset" value="Annuler" class="bouton" />
</p>
```

Source CSS

```
input.bouton
{
  border:2px outset red;
  font-weight:bold;
  cursor:pointer;
}

input.bouton:hover
{
  border:2px outset white;
  background-color:white;
  color:red;
}

input.bouton:active
{
  border:2px inset red;
  background-color:red;
  color:white;
}
```

Le premier style concerne l'apparence du bouton au "repos", le second (**input.bouton:hover**) correspond au bouton survolé, et le dernier (**input.bouton:active**) est celui du bouton enfoncé.

Le résultat devrait être alors le suivant :



 Les pseudos-classes **:hover** et **:active** ne sont supportées par MSIE6 que pour les liens (a); idem pour **:active** avec MSIE7.

IX-B-2 - Focus

Il est possible aussi d'attribuer un effet sur le focus des champs. Mais cette propriété CSS, pourtant si pratique, n'est pas implantée par MSIE (même la version 7).

Source CSS

```
input:focus, textarea:focus
```

Source CSS

```
{  
  background-color:white;  
  color:red;  
}
```

Savez-vous ce que veut dire CSS ? : oui non

Si oui, les utilisez-vous :

Votre email :

Vos commentaires :

 [Testez la mise en forme de ce formulaire.](#)

Exemples de mises en forme de formulaires grâce aux CSS : [CSS-Based Form.](#)

X - Faire apparaître caractères ou images au survol de la souris (rollover)

A la question : "Comment faites-vous pour faire apparaître des guillemets devant vos liens du menu ?", plusieurs réponses possibles :

X-A - Première méthode

Celle que j'utilisais au début **a fait débat**. Au vu des différents commentaires, suggestions et critiques, on peut l'abandonner au profit de celles qui suivent.

X-B - Deuxième méthode

Fonctionne avec :

- FireFox
- NS 6/7
- Mozilla 1.x
- Opéra 7
- MSIE 5+
- Safari

Attributs utilisés :

- background-image
- background-repeat
- background-position
- padding-left

Le guillemet est ici une image qui va être déclarée en image de fond (**background-image**) associé au **a:hover** (lien survolé) dans la feuille de style. Afin qu'elle n'apparaisse pas en mosaïque, on indiquera l'option **no-repeat** à l'attribut **background-repeat**. Reste à positionner correctement l'image par rapport au texte : à l'attribut **background-position** on indique les positions relatives à la ligne (horizontale - verticale). Le **padding-left** indique le retrait que prendra le texte lors du survol de la souris afin qu'il ne recouvre pas l'image.

Source (X)HTML

```
<div id="menu">
  <p>
    <a href="#">menu 1</a><br />
    <a href="#">menu 2</a><br />
    <a href="#">menu 3</a>
  </p>
</div>
```

Source CSS

```
#menu a:hover
{
  background-image:url(images/guill_rouge.png);
```

Source CSS

```
background-repeat:no-repeat;
background-position:1% 50%;
padding-left:15px;
}
```

 **Résultat****X-C - Troisième méthode**

Fonctionne avec :

- FireFox
- Mozilla 1.3+
- Opéra 7

Attributs utilisés :

- content
- background-image
- background-repeat

La méthode précédente interdit forcément une autre image de fond... Pour pouvoir combiner image de fond (ici gris_anim.gif) et guillemet, on peut utiliser l'attribut **content** et la pseudo-classe **before**.

Attention, MSIE 5.x et 6 n'implémentent pas ces propriétés, mozilla ne comprend la combinaison **a:hover:before** qu'à partir d'une version supérieure à 1.2.

CSS

```
#menu a:hover
{
  background-image:url(images/gris_anim.gif);
  background-repeat:no-repeat;
}

#menu a:hover:before
{
  content:" » ";
}
```

Résultat théorique en image :

 **Voir le résultat en ligne.**

XI - Zoomer une image avec les CSS

Fonctionne avec :

- FireFox
- MSIE 7
- NS 6/7
- Mozilla 1.x
- Opéra 7
- Safari

éléments utilisés :

- `img:hover`

XI-A - Première technique

Zoomer une image, à l'aide des **CSS2**, lors de son survol de la souris peut se faire très facilement. Il suffit de combiner la valeur **:hover** à la balise **img** (image).

L'image est déclarée dans le code html sans valeurs de tailles (ni **width**, ni **height**) :

Source (X)HTML

```
<div class="zoom">
  <p>
    
  </p>
</div>
```

Pour éviter tout saut, ou recouvrement, du texte sous-jacent lors du zoom, on peut déclarer cette image dans un cadre (ici appelé **zoom**) de hauteur égale à la photo zoomée (ici 400px). De même, ce cadre servira aussi à limiter cette propriété grossissante à cette seule image.

Les valeurs de tailles sont déclarées au sein de la feuille de style, tout d'abord dans son état initial (**img**), puis dans son état survolé (**img:hover**)

Ici, les tailles de l'image passeront de 200x267 pixels à 300x400 pixels.

Source CSS

```
.zoom
{
  height:400px;
}

.zoom p
{
  text-align:center;
}
```

Source CSS

```
.zoom img
{
  width:200px;
  height:267px;
}

.zoom img:hover
{
  width:300px;
  height:400px;
}
```

 **Résultat**

Vous ne voyez rien se passer ?! Pauvre de vous... Vous utilisez sûrement MSIE 6 (ou 5.5) !

C'est tout le problème de cette technique pourtant si simple : MSIE 6 n'implémente pas la valeur **:hover** pour autre chose que des liens. Grave et encombrante lacune heureusement comblée par MSIE7.

XI-B - Seconde technique

Fonctionne avec :

- FireFox
- NS 6/7
- Mozilla 1.x
- MSIE 6+
- Opéra 7
- Safari

Attributs utilisés :

- background-image
- background-repeat
- background-position
- cursor
- display
- height
- margin
- width

Pour cet âne bâté d'Internet Explorer, il va falloir tricher en faisant croire que l'image est un lien.

Cette fois l'image ne sera plus déclarée dans la page html mais uniquement dans la feuille de style en fond d'écran (**background**).

De plus, il faudra avant tout travailler la photo ou figurera, l'une en dessous de l'autre, l'image initiale et l'image à apparaître lors du zoom.

Inconvénients de la méthode :

- 1 L'image sera plus lourde, donc plus longue à charger.
- 2 En cas d'impression, la photo n'apparaîtra pas si l'internaute a configuré la non-impression des fonds d'écran.

Source (X)HTML

```
<div class="zoom1">
  <p>
    <a href="#" title="pensées de Giverny"></a>
  </p>
</div>
```

L'image **pensees_2.jpg** a donc été créée avec un bon logiciel de traitement d'images (PUB : **PhotoFiltre** est agréablement performant, et gratuit en cas d'utilisation privée). Elle a une taille finale de 300x800 pixels pour un poids de 49 ko.

Toute l'astuce va être de faire apparaître les morceaux choisis de cette image : la partie haute par défaut (300x400 pixels) puis la partie basse (300x400 pixels aussi) au survol de la souris.

Cette dernière devant "remonter" de 400 pixels vers le haut, on indiquera pour la position de fond de page **background-position:0px -400px** ; le **0** étant relatif à la position horizontale et le **-400px** à la position verticale.

Ne pas oublier surtout le très important **display:block** qui permet ici le "remplacement" de la première portion d'image par l'autre.

Source CSS

```
.zoom1
{
  width: 300px;
  height: 400px;
  background-image: url(pensees_2.jpg);
  background-repeat: no-repeat;
  margin: 0 auto;
}

.zoom1 p, .zoom1 a
{
  height: 400px;
  display: block;
  margin:0
}

.zoom1 a:hover
{
  background-image: url(pensees_2.jpg);
  background-position: 0 -400px;
  background-repeat:no-repeat;
  cursor:help;
}
```



margin:0 auto déclaré dans **.zoom1** permet de centrer horizontalement l'image dans la page. Hélas, encore une fois, MSIE ne l'implémente pas...

cursor:help déclaré dans **zoom1_1 a:hover** permet de changer l'apparence du curseur lors du survol de l'image. D'autres valeurs, comme **cursor**, **crosshair**, **default** sont aussi disponibles.

Résultat

XI-C - Variante

On peut choisir évidemment de garder les mêmes dimensions d'images pour ne zoomer qu'une partie.

Pour cette variante, j'ai choisi de zoomer une seule de ces fleurs. L'image, dans ce cas, est [pensees.jpg](#).

Elle a une taille de 200x534 pixels pour un poids de 27 ko. Cette fois, la "remontée" de l'image ne doit être que de 267 pixels.

Avec le même code (x)html que ci-dessus, le code CSS devient :

```
.zoom1
{
  width: 200px;
  height: 267px;
  background-image: url(pensees.jpg);
  background-repeat:no-repeat;
  margin: 0 auto;
}

.zoom1 p, .zoom1 a
{
  height: 267px;
  display: block;
  margin:0
}

.zoom1 a:hover
{
  background-image: url(pensees.jpg);
  background-position: 0px -267px;
  background-repeat: no-repeat;
  cursor:help;
}
```

Résultat

XI-D - Compléments d'informations

Cette technique d'images "glissantes" appliquée aux zooms est inspirée de l'article "[Les Portes Couissantes de CSS - Chapitre II](#)".

D'autre part, elle peut être aussi appliquée à des pseudo découpes d'images comme le décrit très bien cet autre article : "[Sprites CSS : Meurs, découpe d'images, meurs !](#)", lui aussi traduit par l'équipe de [pompage.net](#).

XII - Zoomer une image façon thumbnail en CSS

Les deux techniques maintenant abordées seront peut-être à recommander à des débutants avertis ;-)

XII-A - Première technique

Fonctionne avec :

- FireFox
- NS 7
- Mozilla 1.x
- MSIE 6+
- Opéra 7
- Safari

éléments utilisés :

- background-color
- color
- display
- height
- width
- margin
- padding
- position
- top
- left

Cette première technique m'a été inspirée par Olivier, le webmestre du site [Boolsite](#). Merci à lui.

Les images sont déclarées dans le code html sans valeurs de tailles (ni **width**, ni **height**), chacune d'elle est dans un **<div class="thumb">** dont les propriétés seront déclarées dans la feuille de style. Une seule image par zoom est nécessaire, le poids de la page doit pouvoir rester raisonnable si l'on n'utilise pas d'images trop grandes.

Code (X)HTML

```
<div class="thumb">
  <a href="#">
    
  </a>
</div>
<div class="thumb">
  <a href="#">
    
  </a>
</div>
<div class="thumb">
  <a href="#">
    
  </a>
</div>
```

Code (X)HTML

```
</a>
</div>
```

J'ai choisi de présenter les photos sur un fond noir déclaré dans **body**. La hauteur de 500px n'est là que pour faire apparaître un ascenseur vertical en cas de résolution d'écran inférieure à 1024*768, et dont l'absence peut rendre difficile la visualisation des photos agrandies. Puis on fixe une bonne fois pour toute la largeur des bords des images à zéro.

Les vignettes déclarées dans la classe **.thumb** ont une taille de 100*75 pixels. Je désire qu'elles flottent les unes par rapport aux autres à gauche (**float:left**), qu'elles soient un petit peu espacées (**margin:1px**). Lorsqu'elles disparaîtront au profit de l'image zoomée, il apparaîtra à la place un rectangle gris. (**background-color:#D3D3D3;**). Et surtout, pour éviter des sauts intempestifs, on les fixe grâce au **display:block**.

Vient ensuite le comportement que doivent avoir ces vignettes lors du survol par le curseur de la souris, et c'est là que ça se corse.

Tout d'abord il va falloir s'affranchir des différences de comportements entre MSIE et Mozilla. On commence par stabiliser le tout avec (encore...) un **display:block** pour les liens.

Il va falloir ensuite avoir recours à un "hack". Pour MSIE, il faut que les liens soient déclarés en position absolue, mais pour Mozilla, il faut que ce soit en position relative sinon le rollover ne fonctionne pas de gauche à droite... Ceci explique le **body>.thumb a:hover** que MSIE n'interprète pas mais qui donnera à Mozilla la bonne définition.

Il est ensuite nécessaire de déclarer la taille des images liées : **.thumb a img** donne donc au navigateur les instructions nécessaires à la taille des images mises en lien dans le **<div class="thumb">**.

Et enfin, on déclare la taille et la position que devront avoir les images zoomées. La position est donc relative aux vignettes, et la taille de 300*225 px.

Vous avez tout suivi ? tout compris ? non ?!! ah flute... voilà le code quand même :

Code CSS

```
body
{
  background-color:black;
  color:white;
  height:500px;
}

img {border:0}

.thumb
{
  width:100px;
  height:75px;
  margin:1px;
  float:left;
  background-color:#D3D3D3;
  display:block;
}

.thumb a
{
  display:block;
}
```

Code CSS

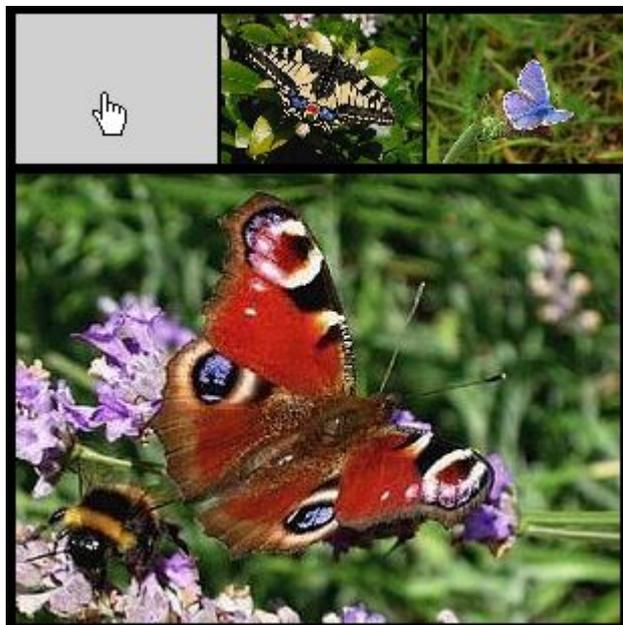
```
.thumb a:hover
{
  position:absolute;
}

/*hack pour permettre le rollover
de gauche à droite avec mozilla*/

body>.thumb a:hover
{
  position:relative;
}

.thumb a img
{
  margin:0;
  padding:0;
  width:100px;
  height:75px;
}

.thumb a:hover img
{
  position:relative;
  left:0px;
  top:80px;
  width:300px;
  height:225px;
}
```



 [Tester le résultat](#)

XII-B - Seconde technique

Fonctionne avec :

- FireFox
- NS 7
- Mozilla 1.x
- MSIE 6+
- Opéra 7
- Safari

Attributs utilisés :

- background-color
- color
- display
- height
- width
- margin
- position
- top
- left
- text-decoration

La première technique fait disparaître la vignette au profit de l'image zoomée et celles-ci se décalent au fur et à mesure vers la droite. La vignette ne disparaît pas avec cette seconde technique, et les images zoomées apparaissent toujours au même endroit. Par contre, le poids de la page sera plus élevé car dans le code (x)html on déclarera la vignette et l'image zoomée.

Pour cet exemple, il y aura donc six images en tout contre trois précédemment.

Code (X)HTML

```
<div class="thumb">
  <a href="#">
    
    
  </a>

  <a href="#">
    
    
  </a>

  <a href="#">
    
    
  </a>
</div>
```

Comme vous le constatez, il n'y a cette fois qu'un seul cadre (**<div class="thumb">**), par contre une autre classe est nécessaire pour les photos zoomées (**class="grand"**).

Même topo que précédemment pour les attributs de **body**, par contre un bord de 1 pixel noir est déclaré pour les images, mais on peut s'en passer, c'est juste une histoire de goût.

La classe **.thumb** définit une position relative du cadre en haut à gauche (à partir de ce qu'il y a juste au-dessus). Dans **.thumb a** on fixe les marges à zéro et le non soulignement des liens pour éviter tout parasite. Et alors là, attention : pour que le survol puisse prendre corps, il faut absolument déclarer quelque chose pour les liens survolés, quelque chose de neutre... un fond de couleur noire par exemple (**thumb a:hover**).

On passe maintenant aux caractéristiques des images zoomées.

La classe **.grand** est appliquée aux images liées, incluses elle-mêmes dans la classe **thumb** (ah ben oui, je sais, ça fait un peu poupées gigognes tout ça...). Donc, dans **.thumb a .grand** on aura un **display:block** et une position absolue (sinon, tout ce petit monde va aller se balader n'importe où), et surtout, une largeur de 0px pour éviter des chevauchements intempestifs.

Il n'y a plus qu'à s'occuper de la taille et de la position de l'image zoomée au moment du survol du **curseur : position absolue**, à 80px en dessous de la position initiale (cad du haut des vignettes), complètement à gauche, et de taille 300*225 pixels.

OUF ! Mal à la tête ? c'est normal, c'est le métier qui rentre. Bon, en même temps ce n'est pas très grave si vous n'avez pas tout compris, moi non plus de toute façon...

Code CSS

```
body
{
  background-color:black;
  color:white;
  height:500px;
}

img
{
  border:1px solid black;
}

.thumb
{
  position:relative;
  top:0;
  left:0;
}

.thumb a
{
  margin:0;
  text-decoration:none;
}

.thumb a:hover
{
  background-color:black;
}

.thumb a .grand
{
  display:block;
  position:absolute;
  width:0px;
}
```

Code CSS

```
.thumb a:hover .grand  
{  
  position:absolute;  
  top:80px;  
  left:0px;  
  width:300px;  
  height:225px;  
}
```



Tester le résultat

C'était un peu plus dur, mais ça valait le coup non ?

XIII - Image de fond en CSS

Fonctionne avec :

- FireFox
- Mozilla 1.x
- MSIE 5+
- NS 6+
- Opéra 6+
- Safari

Attributs utilisés :

- background-image
- background-repeat
- background-position

Mettre une image de fond dans une page web est un exercice très répandu. Avec tout éditeur de page html, l'option classique est de générer un code du type : **<body background="images/image_de_fond.png">**, auquel on peut même ajouter **bgproperties="fixed"** qui rend fixe l'image de fond ; mais cet attribut ne fonctionne qu'avec MSIE.

Avec les feuilles de style on peut obtenir beaucoup plus !

Petits avertissements préliminaires :

- Une image de fond doit être **légère**. Comprenez par là qu'une image de 20 ko c'est déjà beaucoup : cela peut être long à charger et provoquer des problèmes d'affichage.
- Attention aux difficultés de lecture qu'une image de fond trop contrastée peut provoquer.
- Attention aussi à prendre une couleur de fond différente de celle de la police de caractère, car en attendant que l'image de fond se charge, on n'y verra que pouic.

Pour illustrer tout cela, **visionnez ce mauvais exemple** : la photo fait 53 ko ; la couleur utilisée, pour la police de caractère et le fond, est blanche.

XIII-A - Code CSS de base

L'image de fond se déclare en général pour la balise **body** (corps de page), mais on peut aussi l'appliquer à un bloc (**<div></div>**), ou un titre (**<h1></h1>**), une citation (**<blockquote></blockquote>**), etc.

Pour le corps de page, la syntaxe CSS de base sera celle-ci :

```
body
{
  color:black;
  background-color:white;
  background-image:url ( images/image_de_fond.png );
}
```

color renseigne la couleur de la police de caractère, **background-color** est de la couleur de fond de page ; **background-image:url()** est bien sûr l'adresse de l'image choisie pour le fond de page.

Sans plus de renseignements, l'image de fond apparaît en mosaïque et défile en même temps que le texte.

Comme d'habitude, cette syntaxe CSS est à mettre dans l'en-tête de la page (entre les balises **<head></head>**) ou dans une feuille de style externe (voir [ici](#) pour ceux qui auraient loupé l'explication).

 [Visionner l'exemple](#)

XIII-B - Fond de page fixe

Pour que le fond de page devienne fixe, on rajoutera dans le code :

```
background-attachment:fixed;
```

 [Visionner l'exemple](#)

XIII-C - Positionner l'image

On peut aussi empêcher la mosaïque grâce à l'attribut **background-repeat** et positionner l'image dans une partie très précise de l'écran.

Par défaut, l'image se positionne en haut à gauche de l'écran :

```
body
{
  color:black;
  background-color:white;
  background-image:url(images/image_de_fond.png);
  background-repeat:no-repeat;
}
```

 [Visionner l'exemple](#)

XIII-C-1 - Exemple positionnement 2

Si l'on veut qu'elle se place en haut à droite, on rajoutera le code :

```
background-position:right top;
```

ou :

```
background-position:100% 0%;
```

 **Visionner l'exemple**

XIII-C-3 - Exemple positionnement 3

Pour une position au centre de la page, cela donnera :

```
background-position:center center;
```

ou :

```
background-position:50% 50%;
```

 **Visionner l'exemple**

XIII-C-4 - Exemple positionnement 4

Pour une position en bas à droite :

```
background-position:right bottom;
```

ou :

```
background-position:100% 100%;
```

 **Visionner l'exemple**

XIII-C-5 - Remarques

- **background-repeat** accepte 4 attributs : **no-repeat** ; **repeat-x** pour une répétition de l'image uniquement horizontale ; **repeat-y** pour une répétition uniquement verticale ; et **repeat** pour la mosaïque complète.
- Vous aurez noté, bien sûr, que les pourcentages dans le **background-position** offrent plus de latitude que les attributs [**right|center|left**] [**top|center|bottom**].



*On peut aussi cumuler ces différentes informations dans un unique **background**.*

```
body  
{
```

```
background: white url(image_de_fond.png) no-repeat right bottom;  
}
```

XIV - Plusieurs images de fond grâce aux CSS

Fonctionne avec :

- Firefox
- IE 6 et +
- NS 6/7
- Mozilla 1.x ; SeaMonkey
- Opéra
- Safari

Attributs utilisés:

- background-position ; background-repeat ; background-image
- margin
- text-align

Oui, on peut mettre plusieurs images de fond dans la même page. Cela dit, il faudra adapter à chaque besoin, même si le principe de base reste le même : il faut faire plusieurs cadres avec, pour chacun, une image.

Prenons l'exemple d'une image (houx2.gif) placée aux quatre coins de la page. On va définir quatre cadres (hg, hd, bg, bd), puis leur attribuer un **background-position** adéquat pour que les images se placent aux quatre coins.

Les marges définies dans le cadre **contenu** sont précisées en fonction de la taille de l'image pour ne pas que le texte empiète dessus.

Code CSS

```
#hg, #hd, #bg, #bd
{
  background-repeat:no-repeat;
  background-image:url(houx2.gif);
}

#hg
{
  background-position:top left;
}

#hd
{
  background-position:top right;
}

#bg
{
  background-position:bottom left;
}

#bd
{
  background-position:bottom right;
}

#contenu
```

Code CSS

```
{
  margin-left:100px;
  margin-right:100px;
}

#contenu p
{
  text-align:justify;
}
```

Code (X)HTML

```
<div id="hg">
  <div id="hd">
    <div id="bg">
      <div id="bd">
        <div id="contenu">
          <h1>Les quatre coins</h1>
          <p> [bla bla]</p>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
```

 **Voir le résultat**

Attention tout de même à ne pas trop abuser de la méthode, trop de cadres tuent les cadres...

XV - Blockquote et retraits

Tout débutant (enfin, beaucoup...) en création de page web constate que l'utilisation balise **blockquote** induit un retrait du texte. Déduction : la balise **blockquote** est faite pour mettre un texte en retrait...

Et bien non ! Elle n'est pas du tout faite pour cela. "Quote" en anglais veut dire "citer". La balise **blockquote** doit donc être utilisée pour mettre en relief un "bloc" de citation.

Exemple :

*L'élément **BLOCKQUOTE** indique une citation longue (avec un contenu de type bloc) et l'élément **Q** est censé être utilisé pour des citations courtes (avec un contenu de type en-ligne) ne nécessitant pas un saut de paragraphe.*

*L'utilisation de l'élément **BLOCKQUOTE** pour obtenir le retrait d'un texte est déconseillée en faveur de l'utilisation de feuilles de style.*

Code (X)HTML

```
<blockquote cite="http://www.la-grange.net/w3c/html4.01/struct/text.html#h-9.2.2">
  <p>
    L'élément BLOCKQUOTE indique une citation longue
    (avec un contenu de type bloc) et l'élément Q
    est censé être utilisé pour des citations courtes
    (avec un contenu de type en-ligne) ne nécessitant
    pas un saut de paragraphe.
  </p>
  <p>L'utilisation de l'élément BLOCKQUOTE
    pour obtenir le retrait d'un texte est déconseillée
    en faveur de l'utilisation de feuilles de style.
  </p>
</blockquote>
```

Tout est dit dans cet [extrait des spécifications du html 4.01](#). Pour obtenir un retrait, il faut donc utiliser les CSS.

Fonctionne avec :

- FireFox
- Mozilla 1.x
- MSIE 5+
- NS 6+
- Opéra 6+
- Safari

Attributs utilisés :

- `margin-left`

Pour induire le retrait d'un paragraphe, on lui assignera une marge gauche (**margin-left**) d'autant de pixels que souhaités.

Si l'on désire répéter plusieurs fois cette mise en forme dans la page, il sera évidemment préférable d'utiliser le **sélecteur class**.

Code CSS du retrait

```
.retrait  
{  
  margin-left:50px;  
}
```

Code (X)HTML

```
<p class="retrait">  
  Ce paragraphe est en retrait de 50 pixels  
  du bord gauche.  
</p>
```

Résultat

XVI - Balise <object> et molécules Rasmol

 *Petit tutoriel plus réservé à des enseignants de physique-chimie...*

La balise **<object>** a été introduite dans les  **spécifications du html 4.0** pour **présenter des objets** (applets java, vidéos, sons, flash etc.) dans les pages html.

Auparavant, la balise **<embed>** était couramment utilisée, mais celle-ci, initialement balise propriétaire de Netscape, n'a jamais été homologuée par le W3C.

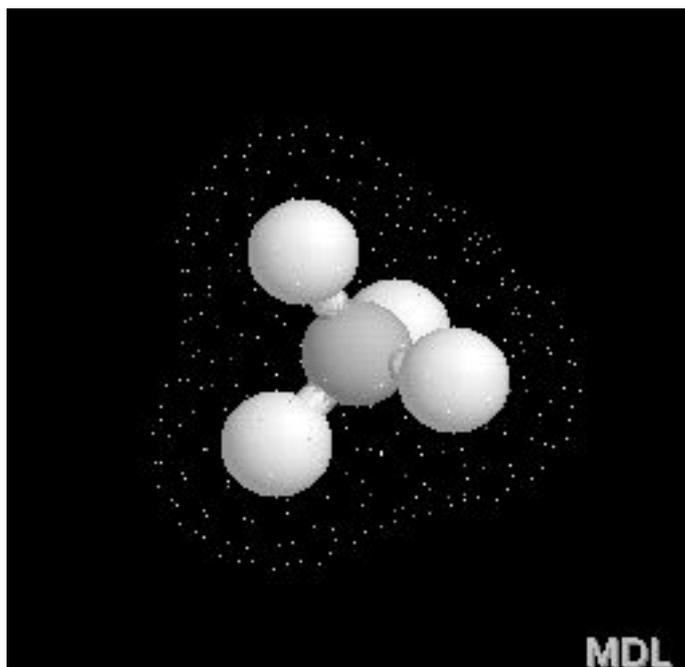
Pour permettre la visualisation des molécules 3D, avec le **plug-in Chime**, le principe est le même : il faut utiliser la balise **<object>** pour respecter les standards.

Pour déclarer la molécule de méthane (fichier **Rasmol** methane.pdb), le code pourra donc être le suivant :

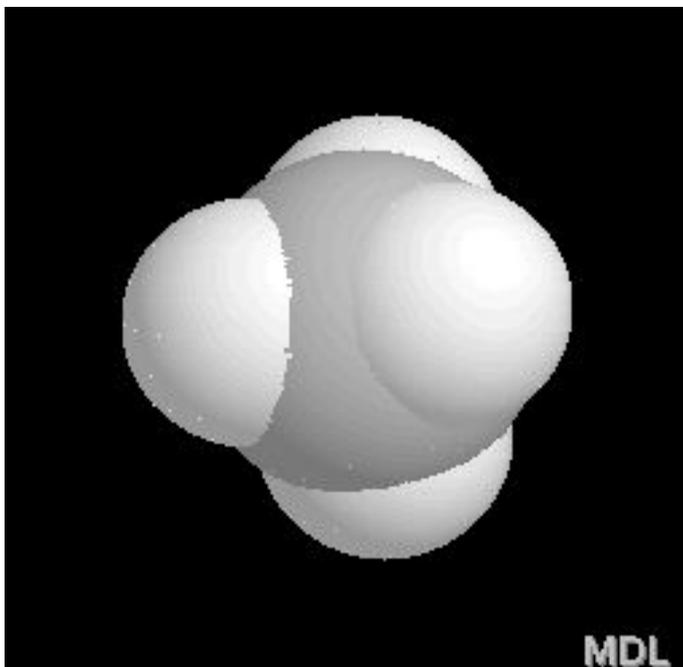
Code (X)HTML

```
<object data="chime/methane.pdb" height="250" width="250">
  <param name="display3D" value="ball&stick" />
  <param name="options3d" value="dots" />
  <param name="frank" value="yes" />
  <param name="spinX" value="10" />
  <param name="spinY" value="50" />
  <param name="spinZ" value="50" />
  <param name="spinfps" value="30" />
  <param name="startspin" value="true" />
  <param name="script" value="zoom 150" />
</object>
```

Ce qui donnera l'apparence suivante (ici statique puisque c'est une capture) :



Comme vous le constatez, on peut indiquer de nombreux paramètres qui influent sur l'aspect de la molécule. Ainsi, par exemple, si l'on veut voir le modèle compact, il faudra changer la valeur **ball&stick** de l'attribut **display3D** en **value="spacefill"**.



Tous  ces paramètres, et bien d'autres, sont disponibles sur le site de **MDL**, concepteur du plug-in chime nécessaire pour visualiser et manipuler ces molécules.

Extraits :

- **display3D={backbone | ball&stick | cartoons | ribbons | spacefill | sticks | strands | wireframe}** : apparence de la molécule
- **options3d={dots | hetero | hydrogen | labels | shadows | slab | specular | stereo}** : options d'apparence
- **frank={false | no | true | yes}** : voir le logo MDL ou pas
- **spinX={degrees per second}** : degré de rotation selon X (idem spinY et spinZ)
- **startspin={true | yes | false | no}** : rotation de la molécule
- **bgcolor={black | white | #rrggbb}** : couleur de fond
- ...

Visionner des exemples de molécules 3D

