

Cours HTML

Les Feuilles de Style CSS



JPP - Juillet 2005

Table des matières

1. Introduction	3
Qu'est-ce qu'une feuille de style ?	3
2. Définition d'un style	4
2.1. Définition	4
2.2. A l'intérieur des balises <HEAD></HEAD> : CSS Globale	4
2.3. A coté des balises : CSS Intra-lignes	5
2.4. Dans un document séparé : CSS importée.....	5
3. Les classes et les ID	6
3.1. Notion de classes	6
3.2. Notions des Pseudo Classes	7
3.3. Notion des ID	7
4. Héritage et cascade	8
4.1. Le principe de l'héritage	8
4.2. Les feuilles de style en cascade	8
5. Les balises structurales	9
5.1. Utilité	9
5.2. DIV	9
5.3. SPAN.....	10
6. Le positionnement	10
6.1. Généralités	10
6.2. Position absolue ou relative ?	10
6.3. Attributs	12
7. Liste des propriétés	13
7.1. Les styles de police	13
7.2. Les styles du texte	14
7.3. Les arrière-plans	14
7.4. Les marges	15
7.5. Les bords et les «enrobages»	15
7.6. Les listes	16

1. Introduction

Qu'est-ce qu'une feuille de style ?

Une feuille de style est un document numérique qui va pouvoir spécifier toutes les caractéristiques de mises en formes du document lié à la balise à laquelle elle s'applique.

Concrètement, vous allez pouvoir définir pour chaque balise HTML, une mise en forme spécifique qui lui sera attachée.

Exemple 1 :

A la balise <H1> je vais pouvoir rattacher la police Arial, de taille 18 pixels, de couleur blanche. Chaque fois que j'utiliserai la balise <H1> dans mon document, tout le texte situé dans les limites de la balise recevra cette mise en forme.

Ainsi j'utiliserai les balises HTML uniquement pour spécifier le contenu du document. A une balise H1, je rattacherai un Titre, mais dont je serai libre de choisir la police, la taille, la couleur ... sans faire appel à d'autres balises qui alourdiraient mon document. La balise H2 indiquera un sous Titre, etc...

Toutes les balises HTML retrouveront leur sens premier.

Vous remarquez que l'on parle de feuilles de style [style sheets] car le but du jeu est d'en définir plusieurs. On parle aussi de feuilles de style en cascade [Cascading Style Sheets ou CSS] car en cas de styles identiques, un ordre de priorité sera déterminé par le browser.

Exemple 2 :

Je veux mettre une partie de mon texte dans une balise <H1> d'une autre couleur que celle définie par <H1>. J'utilise alors un autre marqueur de style (en l'occurrence peut être une classe dont on parlera plus loin), et le tour est joué. De même je peux vouloir étendre les propriétés d'H1, par exemple sur le style du caractère, il me suffira là encore de rajouter une mention de style par une balise ou un autre marqueur.

Précisons pour terminer que les feuilles de style ne sont pas une composante directe du langage Html mais un développement à part dans la publication de pages Web.

Utilité et avantages

- Séparation du contenu et de la mise en forme.
- Cohésion de la présentation tout au long du site avec les feuilles de style externes.
- Modifier l'aspect d'une page ou d'un site sans en modifier le contenu et cela en quelques lignes plutôt que de devoir changer un grand nombre de balises.
- Un «langage» neuf, compréhensible, simple et logique par rapport au Html et à ses différentes versions.
- Une façon d'écriture concise et nette par rapport au Html qui devient vite fouillis.
- Réduire le temps de chargement des pages.
- alier certaines insuffisances du langage Html (contrôle des polices, contrôle de la distance entre les lignes, contrôle des marges et des indentations (sans devoir utiliser de tableaux ou de balise <DD>...) et ainsi augmenter la créativité des écrivains du Web.
- Permettre le positionnement au pixel près du texte et/ou des images sans passer par les «layers» exclusifs à Netscape 4.0.

2. Définition d'un style

2.1. Définition

La syntaxe d'un style est toujours la même, elle précise la balise à laquelle le style va s'appliquer, et les différents attributs du style. Les attributs sont enfermés entre deux accolades ou 2 guillemets (précédés de Style=) selon les cas, et chacun des attributs est séparé par un point virgule. On donne la valeur de l'attribut par la syntaxe suivante «nom de l'attribut:valeur de l'attribut».

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

Exemple :

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

Donc ici, la balise H3 sera en Arial et en italique. Et dans votre document, toutes les balises <H3> auront comme style Arial et italique.

Commentaires :

- Les feuilles de style portent principalement sur des balises et quelques autres éléments comme par exemple A:link pour un lien non-visité et A:visited pour un lien visité. Comme balises souvent utilisées avec des feuilles de style, on peut citer les en-têtes Hn, P, BODY...
- Les propriétés de style sont entourées par des «{» et par des [ou des parenthèses.
- Le couple «propriété de style/valeur» est séparé par un double-point (:).
- Chaque couple «propriété de style/valeur» est séparé par un point-virgule (;).
- Il n'y a pas de limite pour le nombre de couples «propriétés de style/valeur».
- L'espace entre propriétés de style et valeur n'est pas obligatoire mais aide fortement à la lisibilité du code source.
- Pour la lisibilité toujours, vous pouvez écrire vos styles sur plusieurs lignes :

**H3 {font-family: Arial;
font-style: italic;
font-color: green}**

- 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).

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

Il existe 3 manières d'utiliser et d'intégrer les feuilles de style dans un document HTML, intra-lignes, globales, ou importées. Chacune de ces manières a ses avantages et ses particularités.

2.2. A l'intérieur des balises <HEAD></HEAD> : CSS Globale

Une feuille globale va se déclarer dans la tête du document, entre les balises HEAD. Elle va s'appliquer globalement à toutes les balises du document qu'elle a défini.

```

<HTML>
<HEAD>
<STYLE type=»text/css»>
<!--
La ou les feuille(s) de style
-->
</STYLE>
</HEAD>
<BODY>

```

- La balise <STYLE> avertit le navigateur que l'on va utiliser des feuilles de style.
- L'attribut type=»text/css» informe que ce qui suit est du texte et qu'il s'agit de cascading style sheets (css). Pour l'instant, il s'agit de la seule possibilité mais on peut prévoir à l'avenir d'autres versions de ce «langage».
- La balise Html de commentaires <!-- ... --> empêche que les browsers qui ne connaissent pas les feuilles de style, tentent d'interpréter ces instructions. Les informations à l'intérieur des tags de commentaires seront ignorées par ces browsers.
- Pour vos propres commentaires à propos des feuilles de style, on utilisera la convention désormais classique (C, C++, Javascript...) de /* commentaires */.

2.3. A coté des balises : CSS Intra-lignes

Une feuille intra-ligne s'insère directement à côté de la balise qu'elle définit, elle ne constitue donc pas véritablement une feuille, simplement elle permet très localement de fixer des attributs à une partie d'un document. Sa déclaration est un peu particulière, en voici une (notez la forme Style=» «):

```

<HTML>
<BODY>
<H1 style=»font-family: Arial; font-style: italic»> blabla </H1>
</BODY>
</HTML>

```

Signalons :

- que le style Arial, italique n'affectera que cette seule balise H1.
- que la syntaxe est légèrement différente de la précédente.
- que l'écriture : <STYLE type=»text/css»>H1 { «font-family: Arial; font-style: italic» }</STYLE> fonctionne aussi.

2.4. Dans un document séparé : CSS importée

Une feuille liée va se déclarer à part, dans un fichier avec une extension .css qui sera mis à côté des autres documents dans le répertoire du site web. Cette feuille de style sera valable pour toutes les pages du site qui l'appelleront dans les balises HEAD. C'est un outil très puissant pour uniformiser la mise en pages d'un grand nombre de documents.

Principe :

On crée d'abord, dans le répertoire du site, un fichier avec l'extension .css soit styles.css qui contiendra toutes les feuilles de style.

```
TD {font-family:Arial,Helvetica; font-size:10pt}
```

Ensuite, on crée une page soit page.htm avec dans la <HEAD> un lien vers ce fichier CSS :

```
<HTML>
<HEAD>
<LINK rel=stylesheet type=»text/css» href=»styles.css»>
</HEAD>
```

Commentaires :

- La balise <LINK> avertit le browser qu'il faudra réaliser un lien.
- L'attribut rel=stylesheet (sans s et sans guillemets) précise qu'il y trouvera une feuille de style externe.
- 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=» ... « donne le chemin d'accès et le nom du fichier à lier.

3. Les classes et les ID

3.1. Notion de classes

Définition

Un sélecteur contextuel, ou classe, associe une règle particulière à un élément en fonction de sa situation.

On appelle sélecteur simple les balises HTML classiques auxquelles on a attribué des caractéristiques de style.

Mais ces sélecteurs simples présentent une contrainte importante, définie de manière générale, il est difficile de changer momentanément certains de leurs attributs ou bien d'en rajouter.

Pour résoudre ce problème, les concepteurs du CSS ont inventé les classes.

Une classe est ce qu'on appelle un sélecteur contextuel. Il va pouvoir venir modifier contextuellement une caractéristique du document.

La définition d'un style était :

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

Elle devient :

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

remarquez le point entre balise et nom_de_classe

Ou, comme la mention de la balise est facultative,

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

Attention! L'emploi du point (.) devant le nom de classe est indispensable.

Pour appeler l'effet de style dans le document, on ajoute le nom de la classe à la balise.

```
<balise class=»nom_de-classe»> ... </balise>
```

Exemple :

Je souhaite mettre ce qui est important dans le texte en gras et en bleu. Je crée la classe .Toto :

.Toto { font-weight: bold; font-color: #000080 }

Et dans le document Html, il suffit d'appeler la classe quand cela se révèle nécessaire :

```
<P class=»Toto»> ... blabla ... </P>
<H1 class=»Toto <>Titre 1</H1>
<TABLE><TR><TD class=»Toto <>cellule</TD></TD>...
```

3.2. Notions des Pseudo Classes

Définition :

Une pseudo classe est un ensemble d'éléments qui répondent à un même critère de contexte forme.

Cette définition ne s'applique concrètement qu'à un cas particulier, celui de la balise A, la balise des liens.

Celle ci peut en effet connaître plusieurs contextes de formes selon que le lien est inactif, visité, ou en train d'être visiter.

Concrètement on peut définir alors pour chacun des états de la balise une mise en forme particulière, voici comment:

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;}
```

3.3. Notion des ID

Lorsqu'on désire utiliser les feuilles de style avec du Javascript on utilise les ID (appelés les identifiants).

Les ID fonctionnent exactement comme les classes.

La syntaxe est :

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

Et pour l'appeler :

```
<balise id=»nom_de_ID»> .... </balise>
```

Notons qu'on ne pourra effectuer qu'un seul appel à #nom_de_ID par page. Ainsi,

```
Pour #Toto{ ... }
```

<P id=Toto> est correct.

Mais si on rencontre dans la même page

```
<H1 id=Toto> ce n'est plus correct !
```

Si vous pensez utiliser des feuilles de style, mais sans vous compliquer la vie avec des scripts, oubliez au plus vite ID et utilisez exclusivement les classes.

4. Héritage et cascade

4.1. Le principe de l'héritage

Certains éléments de la page HTML héritent des propriétés des styles définis. Supposons le style suivant où la couleur turquoise en arrière-plan est appliquée à la balise H1 :

Lorsque nous appliquons la balise H1 dans le corps de la page, il arrive fréquemment que nous y combinions d'autres balises comme ceci :

```
<H1> Le principe de <I> l'héritage </I> </H1>
```

Nous nous retrouvons avec la balise italique à l'intérieur de la balise H1. En conséquence, la balise italique héritera des propriétés du style H1 et affichera également la couleur de fond turquoise. Le même principe s'applique à beaucoup d'autres styles.

Prenons l'exemple de 3 styles imbriqués



- Le Style 1 se caractérise par son propre style
- Le Style 2 se caractérise par son propre style et celui du style 1
- Le Style 3 se caractérise par son propre style et ceux des styles 1 et 2

4.2. Les feuilles de style en cascade

En cas de concurrence entre plusieurs éléments de style, intervient alors la notion de « cascade » ou d'ordre de priorité.

La concurrence entre plusieurs éléments de style peut provenir des différentes possibilités de localisation de feuilles de style :

- dans un fichier externe avec l'extension css.
- dans la balise HEAD du document.
- dans le BODY du document.

La règle de priorité appliquée par le browser sera d'appliquer pour la présentation du document la feuille de style la plus proche de l'élément.

Ainsi, un style spécifié dans le BODY sera retenu par rapport à un style déclaré dans un fichier externe.

Il y a cependant moyen de contourner ces règles de priorité par la déclaration ! important;

Exemple (Ne fonctionne que sous Explorer) :

Si nous tapons :

```
<HEAD>
<STYLE TYPE=»text/css»>
<!--
BODY { background-color : #0000FF }
-->
</STYLE>
</HEAD>
<BODY STYLE=»background-color:#FF0000; «>
```

Le fond d'écran sera...Rouge (#FF0000)

Par contre, si nous tapons :

```
<HEAD>
<STYLE TYPE=»text/css»>
<!--
BODY { background-color : #0000FF ! important;}
-->
</STYLE>
</HEAD>
<BODY STYLE=»background-color:#FF0000; «>
```

Le fond d'écran sera bleu (#0000FF)

5. Les balises structurales

5.1. Utilité

Il fallait penser à un système pour «déconnecter» certains morceaux de paragraphe ou plusieurs paragraphes de cette logique d'écriture avec des feuilles de style.

Ce sont les balises Structurales qui créaient ainsi des petits blocs particuliers dans le document sans devoir repasser par les éléments structurels du Html classique.

5.2. DIV

DIV est une balise de division qui permet de définir un bloc de texte particulier.

Procédures :

- Intégrer chaque grande portion de document dans une balise DIV particulière.
- Utiliser une feuille de style pour chaque DIV.

```
<HEAD>
<STYLE type=»text/css»>
.zone{font-size: x-small}
</STYLE>
</HEAD>
<BODY>
La balise <DIV>
<DIV class=zone>
bla bla
bla
</DIV>
```

Il est à noter que Netscape ne supporte pas l'imbrication de DIV (et généralement des balises

de bloc) et les gère très mal. Les DIV en cascades sont donc à éviter.

5.3. SPAN

SPAN est une balise de marquage, qui va considérer une petite portion particulière de texte. Elle va servir à mettre en valeur des citations, des exemples, des extraits ...

On peut l'utiliser de la même manière que DIV avec des feuilles de style. Elle s'utilise également couramment avec des feuilles de style intra-lignes.

Ainsi, si je voulais écrire :

Toto est beau.

Le code serait :

```
<STYLE type=»text/css»>
.element {font-size: x-large; color: navy}
</STYLE>
Toto est <SPAN class=element>beau</SPAN>.
```

6. Le positionnement

6.1. Généralités

Grâce au CSS, il est désormais possible de positionner, au pixel près, du texte ou une image avec les feuilles de style.

Le positionnement des éléments par les feuilles de style est repris sous la spécification CSS-P représentant le positionnement dynamique.

Le positionnement dynamique n'est rien d'autre en fait qu'une série de nouveaux attributs de styles, spécialisés dans la gestion de la position et de la superposition du texte. Ce n'est pas grand chose mais cela fait tout.

Notons que ce positionnement n'est possible que sous les versions 4 de Netscape et d'Explorer. Et que cette technique est encore un peu hasardeuse à ce jour, surtout sur le plan de la compatibilité avec les deux browsers susnommés. De plus, si Explorer accepte des attributs CSS-P sur à peu près toutes les balises, Netscape est plus capricieux et pourra générer des erreurs.

Les attributs CSS-P se rajoutent dans les attributs des feuilles de styles classiques. Mais toutes les balises HTML ne peuvent recevoir les attributs CSS-P. De manière générale, seules les balises de bloc (DIV) peuvent utiliser CSS-P.

6.2. Position absolue ou relative ?

Position spécifie le type de positionnement du document. Il en existe 3 types, static (le type par défaut), absolue, ou relative.

Position absolue

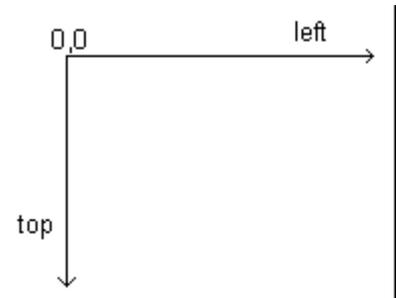
La position absolue {position: absolute} se détermine par rapport au coin supérieur gauche de la fenêtre du browser.

Les coordonnées de ce point sont top = 0 et left = 0.

Il y a 3 options pour top et left :

- auto (par défaut),
- pixels (vous précisez la valeur suivie de px),
- pourcentage (la même chose suivie de %).

Les coordonnées d'un point de haut en bas pour top et de gauche à droite pour left.

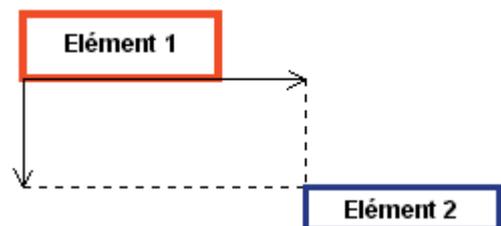


Pour appliquer une position absolue à un texte vous pouvez utiliser les balises SPAN ou DIV. La balise DIV est préférable pour les grands textes qu'elle fera précéder et suivre d'un saut de ligne; la balise SPAN, conçue comme simple marqueur, sera utilisée pour encadrer un paragraphe.

Position relative

La position relative {position: relative} se détermine par rapport à d'autres éléments de la page, par exemple un élément du code Html.

Un positionnement relatif sera donc traité dans le flux du document (qui est traité de bas en haut), et fera référence pour son positionnement à l'élément qui lui est immédiatement supérieur. Cette propriété est valable pour tous les types de balises.



Exemple d'application :

1

Soit l'image 1.jpg :

```
.toto {
position:absolute; top:10px; left:10px;
}
<BODY BGCOLOR=#000080 TEXT=#FF0000>
<IMG SRC=»1.jpg» BORDER=0 WIDTH=50 HEIGHT=50>
<B><DIV CLASS=»toto»>Toto et titi</DIV></B>
```

Donne en position absolue :



et en position relative avec «Toto et Tata»:



6.3. Attributs

Clip

Il s'applique uniquement aux éléments de position absolue. Il va spécifier la zone de visibilité du document, c'est à dire l'endroit où le document contenu dans les balises pourra ne plus être visible s'il dépasse les paramètres de largeurs et de hauteurs fixés par clip.

La syntaxe de clip varie selon les navigateurs, mais les recommandations du W3C sont du type:

clip:rect(haut,droite,bas,gauche)

Exemple : prenons l'image 1.jpg avec CLIP

```
<HEAD>
<STYLE TYPE=»text/css»>
<!--
.toto {
position:absolute; top:10px; left:10px;
clip:rect(0, 25, 25, 0);
}
-->
</STYLE>
</HEAD>
<BODY BGCOLOR=»#000080» TEXT=»#FF0000»>
<DIV CLASS=»toto»><IMG SRC=»1.jpg» BORDER=0 WIDTH=50 HEIGHT=50</DIV>
</BODY>
```

Donne :



Remarquez que les $\frac{3}{4}$ de l'image ont disparu !

Visibility :

Visibility, précise la visibilité ou non du document. Elle possède deux options qui n'ont pas la même syntaxe sous Explorer et sous Netscape.

- Visible pour Explorer, show pour Netscape;
- Hidden pour Explorer, hide pour Netscape.

L'option par défaut est bien entendu visible.

Z-index

Z-index, c'est l'attribut qui va définir l'empilement des feuilles de styles. Il indique l'axe vertical d'empilement, ou la priorité d'affichage entre éléments superposés. Il s'applique à tous les éléments de position relative ou absolue. Plus l'index est grand, plus l'élément est situé dessus. Plus il est bas, plus il est situé dessous.

Deux éléments de même index vont se superposer. C'est très pratique pour afficher du texte sur une image, ou vice-versa.

Exemple avec les deux images 1.jpg et 2.jpg :

```
<STYLE TYPE=»text/css»>
```

1

2

```

<!--
.toto {
position:absolute; top:10px; left:10px;
}
.titi {
position:absolute; top:30; left:30; z-index:2;
}
-->
</STYLE>
</HEAD>
<BODY BGCOLOR=#000080 TEXT=#FF0000>
<DIV CLASS=toto><IMG SRC=1.jpg BORDER=0 WIDTH=50 HEIGHT=50></DIV>
<DIV CLASS=titi><IMG SRC=2.jpg BORDER=0 WIDTH=50 HEIGHT=50></DIV>

```

donne :



Si l'on place le z-index:2 sur la classe Toto :



7. Liste des propriétés

7.1. Les styles de police

font-family

définit un nom de police ou une famille de police
 <nom> ou <famille>
 police précise (Arial, Times, Helvetica...) ou
 famille (serif, sans-serif, cursive, fantasy, monospace)
 H3 {font-family: Arial}

font-style

définit le style de l'écriture
 normal ou italique ou oblique
 H3 {font-style: italic}

font-weight

définit l'épaisseur de la police
 normal ou bold ou bolder ou lighter ou valeur numérique soit
 (100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900)
 P {font-weight: bold}

font-size

définit la taille de la police
 xx-small ou x-small ou small ou médium ou large ou x-large ou xx-large
 ou larger ou smaller
 ou taille précise en points (pt), inches (in), centimètres (cm), pixels (px)
 ou pourcentage (%)
 P {font-size: 12pt}

font-variant

définit une variante par rapport à la normale
 normal ou small-caps
 P {font-variant: small-caps}

font

raccourci pour les différentes propriétés de police
 P {font: bold italic}

7.2. Les styles du texte

text-align

définit l'alignement du texte
left ou center ou right
H1 {text-align: center}

text-indent

définit un retrait dans la première ligne d'un bloc de texte
souvent utilisé avec <P>, n'oubliez pas dans ce cas </P>.
spécifié en inches (in) ou en centimètres (cm) ou en pixels (px)
P {text-indent: 1cm}

text-decoration

définit une décoration (?) du texte, soit barré, clignotant, etc.
blink ou underline ou line-through ou overline ou none
A:visited {text-decoration: blink}

text-transform

définit la casse du texte (majuscule, minuscule)
uppercase (met les caractères en majuscules) ou
lowercase (met les caractères en minuscules) ou
capitalize (met le premier caractère en majuscule)
P {text-transform: uppercase}

color

définit la couleur du texte
par exemple en hexadécimal
H3 {color: #000080}

word-spacing

définit l'espace entre les mots
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
P {word-spacing: 5pt}

letter-spacing

définit l'espace entre les lettres
spécifié en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
P {letter-spacing: 2pt}

line-height

définit l'interligne soit l'espace entre les lignes du texte
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
P {line-height: 10pt}

width

détermine la longueur d'un élément de texte ou d'une image
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H1 {width: 200px}

height

détermine la hauteur d'un élément de texte ou d'une image
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H1 {height: 100px}

white-space

espace ou blanc
normal ou pre ou nowrap
PRE {white-space: pre}

7.3. Les arrière-plans

background-color

définit la couleur de l'arrière-plan
couleur (par exemple en hexadécimal) ou transparent
H1 {background-color: #000000}

background-image

définit l'image de l'arrière-plan
URL de l'image
BODY {background-image: image.gif}

background-repeat

définit la façon de répéter l'image d'arrière-plan
repeat ou no-repeat ou repeat-x (x = nombre de répétitions horizontales) ou
repeat-y (y = nombre de répétitions verticales)
P {background-image: image.gif; background-repeat: repeat-4}

background-attachment

spécifie si l'image d'arrière-plan reste fixe avec les déplacements de l'écran
scroll ou fixed
BODY {background-image: image.gif; background-attachment: fixed}

background-position

spécifie la position de l'image d'arrière-plan par rapport au coin
supérieur gauche de la fenêtre
{1, 2}
{top ou center ou bottom , left ou center ou right}
ou en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
BODY {background-image: img.gif; background-position: right top}

background

raccourci pour les différentes propriétés d'arrière-plan
P {background: image.gif fixed repeat}

7.4. Les marges

margin-top

détermine la valeur de la marge supérieure
en unité de longueur ou auto
{ margin-top: 5px }

margin-right

détermine la valeur de la marge droite
en unité de longueur ou auto
{ margin-right: 5px }

margin-bottom

détermine la valeur de la marge inférieure
en unité de longueur ou auto
{ margin-bottom: 5px }

margin-left

détermine la valeur de la marge gauche
en unité de longueur ou auto
{ margin-left: 5px }

margin

regroupe les différentes propriétés de la marge

7.5. Les bords et les «enrobages»

border-top-width

donne l'épaisseur du bord supérieur
thin ou medium ou thick ou spécifié par l'auteur
H3 {border-top-width: thin}

border-right-width

donne l'épaisseur du bord droit
thin ou medium ou thick ou spécifié par l'auteur
H3 {border-right-width: medium}

border-bottom-width

donne l'épaisseur du bord inférieur
thin ou medium ou thick ou spécifié par l'auteur
H3 {border-bottom-width: thick}

border-left-width

donne l'épaisseur du bord gauche
thin ou medium ou thick ou spécifié par l'auteur
H3 {border-left-width: 0.5cm}

border-width

regroupe les différentes propriétés de border-width

border-color

détermine la couleur de la bordure
H3 {border-color: yellow}

border-style

détermine le style du trait de la bordure
none ou solid ou dotted ou dashed ou double
ou groove ou ridge ou inset ou outset
{border-style: solid dashed}

border

regroupe toutes les propriétés des bords

padding-top

valeur de remplissage haut entre l'élément et le bord
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H3 {padding-top: 3px}

padding-right

valeur de remplissage droite entre l'élément et le bord
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H3 {padding-right: 3px}

padding-bottom

valeur de remplissage bas entre l'élément et le bord
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H3 {padding-bottom: 3px}

padding-left

valeur de remplissage gauche entre l'élément et le bord
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H3 {padding-left: 3px}

padding

regroupe les différentes propriétés de remplissage

7.6. Les listes

list-style-type

détermine le type de puces ou de numérotation
disc ou circle ou square
decimal ou lower-roman ou upper-roman ou lower-alpha ou upper-alpha
OL {list-style-type: square}

list-style-image

permet de remplacer les puces par une image
url ou none
OL {list-style-image: image.gif}

list-style-position

spécifie si les puces sont à l'intérieur ou à l'extérieur du texte
inside ou outside
UL {list-style-position: inside}

list-style

regroupe toutes les propriétés de liste