

Les feuilles de style

Chapitre 1: Concept et utilité

1.1 Présentation

Le concept des feuilles de style n'est pas à proprement parler une nouveauté dans le domaine de la publication Html. Introduit en 1997 par Microsoft avec son Internet Explorer 3.0 (mais elles existaient déjà avec Arena sous Unix), ces feuilles de style n'ont connu qu'un intérêt limité du fait que celles-ci n'étaient pas reconnues par Netscape Navigator 3.0.

Depuis les choses ont bien changé. D'abord les browsers 4.0 de Microsoft et de Netscape reconnaissent tous deux les feuilles de style et surtout, la norme Html 4.0 en a repris le concept (CSS version 1) et le recommande d'ailleurs vivement aux "Web designers".

1.2 Concept

Dans un document d'une certaine importance, il arrive fréquemment que l'on attribue à certains éléments des caractéristiques de mise en forme identiques. Par exemple, les noms de chapitres seront mis en police Arial, en gras et en couleur bleue.

On peut imaginer que l'on puisse donner à cette définition de mise en forme un nom soit "titre" et qu'à chaque nouveau chapitre, plutôt que d'écrire chaque fois le nom du titre et puis de devoir le mettre en Arial, gras, bleu, l'on puisse dire à l'ordinateur, nom du chapitre mais dans la mise en forme que j'ai défini sous le nom de "titre". Cette définition de mise en forme particulière, on va l'appeler feuille de style.

Le concept de feuilles de style [Style Sheets] est né. Il existait déjà dans les traitements de texte comme dans Word de Microsoft (comme par hasard...). Allez dans le menu Format de Word, vous y trouvez Style ! Il ne restait plus qu'à coupler le concept au langage Html par des propriétés spécifiques.

<code><H1>Titre1</H1></code>	STYLE des titres
<code><H2>- A -</H2></code>	STYLE des sous-titres
<code><H3>...a...</H3></code>	STYLE du texte
<code><H1>Titre2</H1></code>	STYLE des titres
<code><H2>- B-</H2></code>	STYLE des sous-titres
<code><H3>...b...</H3></code>	STYLE du texte

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 (voir FAQ).

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.

1.3 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'un 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.
- Palier 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.

1.4 Compatibilité

Les feuilles de style fonctionnent sous :

Explorer 3.0 mais de façon incomplète
 Explorer 4.0
 Netscape 4.0

Attention !!! Les feuilles de style ne sont pas reprises par Netscape 3.0.

Chapitre 2 : Définition d'un style

La définition de base d'un style est simple :

```
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.

Simple! Mais de nombreux commentaires s'imposent :

- Les feuilles de style portent sur des balises principalement 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 pas 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}
```

Chapitre 3 : Styles internes

Il faut maintenant incorporer les styles dans le document Html. Commençons par le plus simple, soit l'incorporation à l'intérieur d'une page. D'où le titre "Styles internes".

3.1 A l'intérieur des balises <HEAD></HEAD>

Cette façon de procéder est de loin la plus commune et la plus logique. D'abord parce que les balises HEAD contiennent des informations pour le browser et les feuilles de style appartiennent à celles-ci. Ensuite parce que l'on rejoint ainsi l'essence même des feuilles de style qui est de séparer les éléments de mise en forme du contenu.

```

<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 */.

3.2 A l'intérieur des balises <BODY></BODY>.

On peut aussi utiliser, au coup par coup, les feuilles de style dans le corps (BODY) du document. Cette façon de faire nous paraît illogique et peu conforme à l'esprit des feuilles de style qui est de définir un style déterminé valable pour la globalité du document. Mais elle existe pour quelques utilisations spécifiques...

```

<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.

Chapitre 4 : Styles externes

C'est déjà bien de pouvoir définir une présentation de style valable pour une page (styles internes). Mais CSS nous propose mieux. Définir une présentation de style valable pour plusieurs pages et même pour toutes les pages d'un site. Ce qui est possible, en créant une page externe qui regroupera toutes les feuilles de style, et en reliant chaque page à cette page de style.

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.

```

<HTML>
<HEAD>
--- Les différentes feuilles de style ---
</HEAD>
<BODY>
</BODY>
</HTML>

```

Ensuite, on crée une page normale soit page1.htm (bien entendu dans le même répertoire que le fichier styles.css).

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

- 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.

Chapitre 5 : Les classes et les ID

5.1 Notion de classes.

Mais on désire parfois affecter des styles différents à une même balise. Pas de problèmes, les feuilles de style vous proposent la solution des classes [class].

La syntaxe est ici aussi des plus simple.

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>
```

Un exemple ?

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

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

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

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

5.2 Notion des ID

Comme la convention nom/point/nom est utilisée aussi en Javascript (voir Apprendre le Javascript du même auteur : www.ccim.be/ccim328/js/index.htm) , il a fallu trouver une autre convention d'écriture lorsqu'on désire utiliser les feuilles de style avec du Javascript. Ce sont les ID, aussi appelés les identifiants.

Les ID fonctionnent exactement comme les classes. Pas mieux, pas plus. C'est la même chose!

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 #essentiel{ ... }
```

```
<P id=essentiel> est correct.
```

Mais si on rencontre dans la même page

```
<H1 id=essentiel> 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.

Si par contre vous souhaitez utiliser des scripts avec les feuilles de style pour faire du DHTML par exemple (voir plus loin dans le site), la notion de ID vous sera alors indispensable.

Chapitre 6 : et <DIV>

6.1 Utilité

Dernier point, 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 respectivement les balises SPAN et DIV qui créaient ainsi des petits blocs particuliers dans le document sans devoir repasser par les éléments structurels du Html classique.

Notons que SPAN et DIV s'utilisent toujours avec les classes et les ID.

6.2 SPAN

La balise ... permet d'appliquer des styles à des éléments de texte d'un paragraphe ou si vous préférez à un morceau de paragraphe. Ainsi je voudrais écrire :

Un monde de **géants**.

```
<HTML>
<HEAD>
<STYLE type="text/css">
.element{font-size: x-large; color: navy}
</STYLE>
</HEAD>
<BODY>
<P>Un monde de <SPAN class=element>géants</SPAN>.</P>
</BODY>
</HTML>
```

6.3 DIV

La balise <DIV> ... </DIV> permet de regrouper plusieurs paragraphes ou si vous préférez, de délimiter une zone comportant plusieurs paragraphes.

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

```
<P>N'oubliez pas l'attribut class!</P>
</DIV>
</BODY>
</HTML>
```

Donne comme résultat :

La balise <DIV>

Commentaire :

N'oubliez pas l'attribut class!

Chapitre 7 : Positionner avec CSS

Quel concepteur de pages Web n'a pas laissé échapper quelques jurons bien sentis en essayant, à grand renfort de tableaux, de placer précisément du texte ou une image là où il le désirait ?

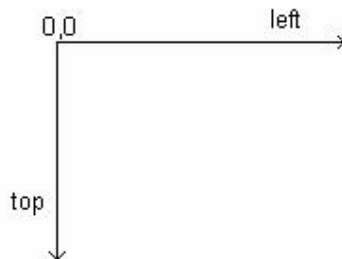
Le miracle est arrivé ! Outre la balise <LAYER> (mais qui n'est comprise que par Netscape 4.0), il est désormais possible de positionner, au pixel près, du texte ou une image avec les feuilles de style. Notons que ce positionnement n'est possible que sous les versions 4 de Netscape et d'Explorer. Et que cette technique est encore un peu hasardeuse à ce jour, surtout sur le plan de la compatibilité avec les deux browsers susnommés.

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

7.1 Position absolue ou relative.

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

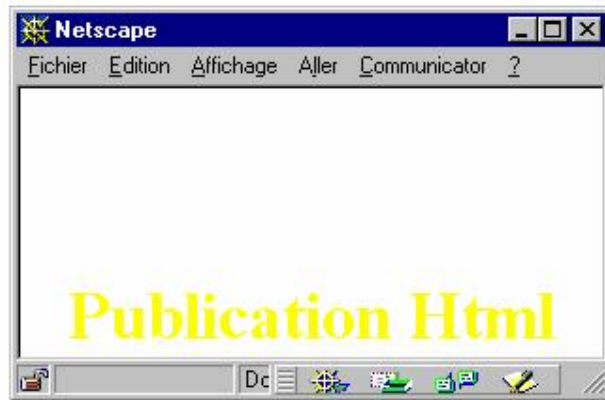
La position relative {position: relative} se détermine par rapport à d'autres éléments de la page, par exemple un élément du code Html.



Précisons que l'on utilisera presque toujours le positionnement absolu car plus facile et plus sûr.

7.2 Positionner du texte.

Plaçons en position absolue le texte "Publication Html" à 50 pixels du haut de la fenêtre (top) et à 150 pixels à gauche (left).



```
<HTML>
<HEAD>
<STYLE type="text/css">
.pub{position: absolute; top: 100px; left: 25px;
color: yellow; font-size: x-large; font-weight: bold;}
</STYLE>
</HEAD>
<BODY>
<DIV class=pub> Publication Html </DIV>
</BODY>
</HTML>
```

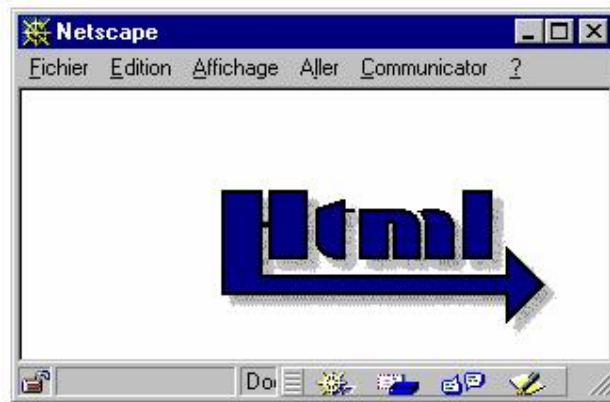
Ajoutons que plusieurs encodages sont possibles.

7.3 Positionner une image.

Plaçons l'image htmlplus.gif en position absolue à 50 pixels de haut de la fenêtre (top) et à 100 pixels à gauche (left). Les dimensions de l'image sont width=242 et height=84.

```
<HTML>
<BODY>
<span style="position: absolute; top: 50px; left: 100px; width: 242px;
height: 84px;">
<IMG src="htmlplus.gif">
</span>
</BODY>
</HTML>
```

Spécifiez toujours les propriétés width et height avec les feuilles de style car par défaut, Netscape 4.0 et Explorer 4.0 ne réagissent pas de la même façon.



Ajoutons que plusieurs encodages sont possibles.

7.4 Superposer du texte sur une image.

Reprenons l'image htmlplus.gif et on y superposera le nom de l'auteur de ce tutorial, au pixel près dans la barre qui souligne le terme Html.

```
<HTML>
<BODY>
<span style="position: absolute; top: 50px; left: 100px; width: 242px; height: 84px;">
<IMG src="htmlplus.gif">
</span>
<span style="position: absolute; top: 96px; left: 145px;
color: yellow; font-size: x-small; font-weight: bold;">
Van Lancker Luc
</span>
</BODY>
</HTML>
```



Ajoutons que plusieurs encodages sont possibles.

Chapitre 8 : Mini FAQ sur les styles

8.1 Les feuilles de style sont-elles "case sensitive" ?.

Non, les feuilles de style ne sont pas sensibles à la case [case insensitive]. Ecrire CLASS ou class ou Class est donc équivalent. Cependant les éléments qui ne sont pas sous le contrôle des feuilles de style comme les noms

de police ou les URLs peuvent être case sensitive. Pour le système d'exploitation (et je pense à Unix), Arial n'est peut-être pas égal à arial, de même IMAGE.gif n'est pas forcément égal à image.gif.

8.2 Quels caractères peut-on utiliser en CSS ?

Les noms des feuilles de style, des sélecteurs, des classes et ID peuvent contenir les lettres de a-z ou de A-Z, les chiffres de 0-9, le trait d'union et le caractère _. Les noms ne peuvent commencer par un chiffre ou un tiret. La documentation officielle affirme que les caractères spéciaux ASCII 160-255 peuvent être utilisés. Mais cela ne fonctionne pas chez moi. On prendra vite l'habitude (voir les langages de programmation) de les éviter.

8.3 Peut-on utiliser dans la même page, à la fois des éléments de style CSS et Html ?

Oui. Les feuilles de style seront ignorées par les browsers qui ne supportent pas les feuilles de style. Et c'est tant mieux pour la compatibilité de votre site sous les différents navigateurs.

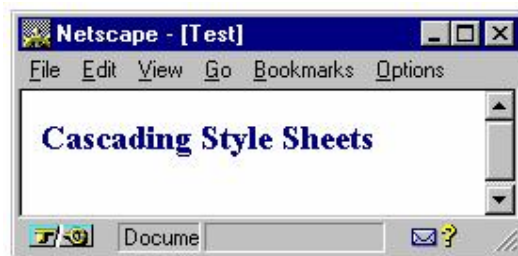
8.4 Qui l'emportent : les attributs Html ou les propriétés CSS ?

Les propriétés des feuilles de style l'emportent sur les attributs Html. Si les deux sont spécifiés, les attributs Html seront affichés avec les browsers qui ne supportent pas les CSS et n'auront aucun effet avec les browsers qui supportent les feuilles de style.

Ainsi, `<H3 style="color: red">Cascading Style Sheets</H3>` apparaîtra en rouge sous un browser compatible CSS (par exemple Microsoft Explorer).



et en bleu sous un browser qui ne reconnaît pas les feuilles de style (par exemple Netscape 3.0).



Ce qui est très bien pour la compatibilité mais qui ne simplifie pas la clarté de l'écriture.

8.5 Et s'il y a concurrence entre plusieurs éléments de style ?

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;. Du genre BODY {background: white ! important; color: black}. Nous en resterons là dans le cadre de ce tutorial.

Chapitre 9 Liste des propriétés

La liste complète (et officielle) des propriétés et recommandations concernant les feuilles de style version 1, peut être trouvée à l'adresse <http://www.w3.org/pub/WWW/TR/REC-CSS1> . En voici une sélection :

9.1. 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 }

9.2. 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}

9.3. 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}

9.4. 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

9.5. 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

9.6. 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

Les feuilles de style
www.ccim.be/ccim328/htmlplus/idxstyle.htm
© copyright 1998

Van Lancker Luc
Rue des Brasseurs, 22
7700 Mouscron
Belgium
Vanlancker.Luc@ccim.be