

1 Pour commencer...

... il faut récupérer le fichier xhtml suivant :

<http://www-lipn.univ-paris13.fr/~toulouse/doc/igw/2009/xhtml-td-css-2-brut.html>

2 Poser les grandes lignes du “*look n feel*”

2.1 Organisation du projet

Les instructions de mise en forme sont externalisées dans un fichier css. Après avoir récupéré le fichier html “brut” (à nommer par exemple “cours-css.html”), créer un fichier texte et le nommer (de nouveau, par exemple !) “css/cours-css.css”. Par ailleurs, pour que ce fichier soit lu par les fichiers html qui vont l’utiliser, il faut signifier dans l’entête de ceux-ci :

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

Attention ! Vérifier que le répertoire de base (celui à partir duquel les chemins relatifs sont considérés) est correctement défini, *ex.*, si le fichier d’accueil du cours css est cours-css.html :

```
<base href="./cours-css.html"/>
```

2.2 Allons-y !

1. Propriétés de l’élément “body”

- image de fond d’url “http://www.w3.org/Icons/w3c_main.gif” (ou `img/w3c_main.gif` si le fichier a été récupéré en local) ;
- placer l’image en haut à gauche (image non répétée), l’image suit le défilement du texte ;
- couleur de fond #DEEDFE, alignement gauche, largeur et hauteur de 100% ;
- style, variante, épaisseur de trait et taille de police normaux ; couleur de police #000000 ;
- famille de police Arial, sans empattements (i.e., sans serifs).

2. Titres, paragraphes, images

- forcer pour les titres h1 la taille et couleur de police xx-large et #9FAFC0 ; alignement centré ;
- attribuer à h1 une bordure (le dessous uniquement) de 10 pixels, avec un espacement au texte (padding) de 10 pixels, la couleur #D0E0F0 et le style “solid” (trait plein) ;
- forcer pour les titres h2 les caractères ianiques en petites majuscules ;
- imposer une marge de 40 pixels pour la première ligne de chaque paragraphe ;
- imposer l’inscription en police “plus large” du premier caractère de chaque paragraphe ;

(f) alignement vertical au centre des images, affichage des images sans bordure.

3. liens

(a) liens de couleur #9999DD et sans décoration ;

(b) liens visités et lien actif respectivement de couleur #444488 et #FFAC00.

3 Raffinons un peu

1. Pour les exemples de code, créer une classe “code” (valable pour tout élément), qui a les spécificités suivantes :

(a) police “plus petite” ;

(b) on inverse les couleurs de fond (\Rightarrow devient #000000) et de police (\Rightarrow devient #DEEDFE).

Mettre en application ces modifications. *Attention !* Utiliser la bonne balise au bon endroit, notamment, choix des niveaux bloc ou texte.

2. Nombreux sont les paragraphes commençant par une sorte d’entête : “Exemple”, “Syntaxe”, “Avantage”. Pour mettre ces entêtes en exergue, définir la classe “entete” de l’élément “span” ayant les caractéristiques suivantes : couleur de texte bleue, caractères italiques.

3. Personnalisons également les listes “ul” : préférer la puce carrée “square” au disque (valeur par défaut), et le positionnement extérieur “outside” (la puce est placée à l’extérieur du cadre) ; de plus, affecter un fond #D0E0F0 aux éléments “li” fils d’une liste “ul” et une bordure de même couleur à “ul” ; enfin, espacer les “li” en définissant un espacement (padding) de valeur “5”.

4. Enfin, il serait plus élégant de centrer le tableau des propriétés. Comment faire ?

4 On peut faire des super menus

4.1 Remplacer le vieux tableau pour une liste toute neuve

Pour les liens de bas de page, nous allons utiliser une liste “ul” plutôt qu’un tableau : chaque lien sera alors contenu dans un item de liste, au lieu d’être contenu dans la cellule d’un tableau. L’utilisation des listes “ul” est simple :

- balises ouvrante et fermante `` et ``
- à l’intérieur : une succession d’item, qui sont des éléments “li”, eux-mêmes ayant une balise ouvrante et une balise fermante ;
- \Rightarrow exemple : Mon panier contient :
`des pommesdes poiresdes oranges`

1. Remplacer donc la structure de table par la structure de liste.

2. Pour obtenir une liste horizontale, il faut attribuer la valeur “inline” à la propriété “display” des éléments “li” contenus dans l’élément “ul”. Pour ce faire, définir une classe “ul” d’instance unique “menu”. D’une part, l’instance de cette classe n’a ni bordure, ni puce ; d’autre part, tous ses “li” fils doivent s’afficher en ligne ; les autres caractéristiques des “li” : marges gauche et droite de 10 pixels, bordure pleine, de 2 pixels, couleur orange.

3. Pseudo classe “:hover” : un lien (et autres objets graphiques) sait quand il est survolé par la souris ; pour le faire apparaître de façon plus marquée à l'utilisateur, changer le style de police (caractères italiques), ainsi que la couleur de fond (orange) et la couleur de bordure (#FFAC00), des liens au passage de la souris.

4.2 Second menu

C'est bien d'avoir un menu en bas, mais ce serait bien aussi d'en avoir un en-haut !

1. Dupliquer le menu pour le situer en haut de la page. *Attention !* Les identifiants devant être uniques, il faut, soit transformer la classe d'instance unique “ul#menu” en classe “ul.menu”, soit créer un nouvel identifiant “ul#menu-entete” pour le nouveau menu ; c'est ce que l'on va faire, car nous allons faire évoluer l'apparence de nouveau menu indépendamment du premier.
2. Horreur, il masque le logo W3C ! Nous allons positionner ce menu à droite. Pour cela, affecter la valeur “right” à l'attribut “float” de l'élément “ul”. La propriété “float” est définie pour tout élément de niveau bloc ; elle permet de déterminer si le bloc est ou non “flottant” dans le texte (= le texte s'inscrit autour de l'objet, ou le texte s'inscrit au-dessus et en-dessous de l'objet).
3. À présent que l'on a deux menus, différencions-les... en ajoutant un lien “bottom” sur le menu du haut, et un lien “top” sur celui du bas. Comment désigner le haut et le bas de la page... pourquoi ne pas utiliser nos menus eux-mêmes ?
4. Finalement, le menu d'entête prend trop de place ; affichons-le verticalement. Par ailleurs... on perd l'intérêt du menu en faisant défiler le texte ; ne pourrait-on pas avoir un menu persistant ? Si ! Il suffit de fixer sa position : attribuer la valeur “fixed” à la propriété “position”... et penser aussi à attribuer les coordonnées, “left” (gauche) et “top” (haut) de l'objet.

Pour positionner un bloc, on doit fournir ses coordonnées : propriétés “left”, “top”, “right” et “bottom”.

Ensuite, il existe 3 types de positionnement possibles :

- Fixe (valeur *fixed*) : les coordonnées sont prises relativement aux bords *de la fenêtre*, et l'élément *ne défile pas* avec la page (*i.e.*, il est fixé sur la fenêtre).
- Absolu (valeur *absolute*) : les coordonnées sont prises relativement aux bords *de la page*, et l'élément *défile* avec la page (*i.e.*, il est fixé sur la page).
- Relatif (valeur *relative*) : les coordonnées sont prises relativement au contexte (éléments qui entourent la boîte de l'élément que l'on positionne).