



# Créons le design de votre site web !

---

## Informations sur le tutoriel

[Ajouter à mes tutoriels favoris](#) (29 fans)



Auteur : [M@teo21](#)

Licence :

[Plus d'informations](#)

## Popularité

Visualisations : 372 758 711

Appréciation 15

des lecteurs : 15

60

888

## Publicité

## Historique des mises à jour

- *Le 08/08/2010 à 03:23:12*  
Correction conjugaison.
- *Le 03/08/2010 à 01:07:48*  
Correction orthographique, #2605
- *Le 31/07/2010 à 18:20:55*  
Correction orthographique, ticket n°2593

Enfin nous y voilà 😊

C'est un chapitre un peu particulier, assez différent de ce que vous avez lu jusqu'à maintenant. En fait, c'est ce que j'appelle un "TP" (= Travaux Pratiques). Maintenant, vous ne pouvez plus vous contenter de lire mes chapitres à moitié endormis, vous allez devoir mettre la main à la pâte en même temps que moi 😊

On va réutiliser tout ce qu'on a appris jusqu'ici. Si vous en ressentez le besoin, il est encore temps d'aller relire les chapitres précédents pour vous rafraîchir la mémoire.

Quel est l'objectif de ce TP ?

Je crois que le titre est assez clair pour le coup : nous allons créer le design de votre site web 😊

Comme je me doute bien que pour vous c'est la première fois, il va falloir être attentifs. Créer un design, ce n'est pas une mince affaire : il faut de la créativité (ça j'en ai pas trop), du goût pour les couleurs (ça j'en ai pas du tout), du talent (décidemment je suis mal barré...) et enfin de l'expérience.

Tout ce que je peux vous communiquer moi, vous l'aurez compris, c'est mon expérience. Après, la créativité et tout le reste c'est inné : on l'a ou on l'a pas 😊

Au boulot moussaillon ! 🧐

Sommaire du chapitre :



- [Bon, par quoi on commence ?](#)
  - [Primo, le XHTML](#)
  - [Secundo, le CSS](#)
  - [Attendez, c'est pas fini !](#)
-

## Bon, par quoi on commence ?

Vous le savez maintenant, une page web c'est une combinaison de 2 fichiers :

- **Un fichier XHTML** (.html ou .htm) : c'est dans ce fichier que se trouve le contenu de la page (le texte). Ce fichier est constitué de balises comme <strong> qui servent à indiquer si tel texte est important, si c'est un paragraphe ou une citation etc...
- **Un fichier CSS** (.css) : c'est le fichier qui permet de créer la présentation de notre page web. C'est lui qui indique qu'un texte est en rouge, qu'il est centré, dans la police "Arial" etc... En théorie, le CSS est facultatif. Ceci dit, sans CSS, nos pages web seraient bien moches 🙄

Si vous avez bien suivi jusqu'ici et que vous avez fait quelque tests, vous savez par quoi on commence :

1. En premier on tape le contenu de la page (XHTML)
2. Et ensuite on modifie la présentation de la page (CSS)

Nous allons donc suivre cet ordre-là.

Vous allez voir que la création du fichier XHTML sera (très) rapide et facile. En revanche, nous allons passer plus de temps sur le CSS. Parce que c'est délicat, qu'il y a pas mal de propriétés CSS à connaître, et aussi parce que des fois, on a beau être un pro, le navigateur n'affiche pas exactement les choses comme on voudrait.

### Design fixe ou extensible ?

Un design fixe, c'est un design dont la largeur est de taille fixée (par exemple de 800 pixels). Il y a en général une marge à gauche et à droite, et le contenu du site se trouve au milieu. Voici un site au design fixe :



*Un design de taille fixe*

Un design extensible, c'est un design qui s'élargit automatiquement en fonction de la résolution du visiteur. Si le visiteur est en 1024\*768, le design fera 1024 pixels de large. Si le visiteur est en 1600\*1200, il fera 1600 pixels de large.



Un design de taille extensible

Personnellement, j'ai tendance à préférer les designs extensibles : je n'aime guère avoir une partie de mon écran inutilisée. Cependant, vous verrez si vous faites le tour des sites web que vous visitez régulièrement que nombre d'entre eux sont en design fixes. Certains disent trouver ça plus joli, mais je vais vous dire la vérité : c'est aussi parce que c'est plus facile à faire 😊

Quel type de design allons-nous faire ?

Eh bien pour tout vous dire, la création d'un design extensible est assez complexe, et comme c'est votre premier design je crains que ça ne fasse trop à la fois. Nous allons donc réaliser un **design de taille fixe**, ce qui est déjà pas si mal. Lorsque vous vous serez améliorés et que vous aurez un peu plus d'expérience, vous pourrez vous lancer dans la création d'un design extensible. Vous aurez tout le temps de vous y intéresser 😊

## Primo, le XHTML

Première étape : on ouvre notre éditeur de texte (Notepad++ si vous avez pris celui que je vous ai montré au début), et vous créez un nouveau fichier XHTML.

Allez-y allez-y, faites-le en même temps que moi ! 😊

Pour commencer, on va taper le code de "base" d'une page XHTML que je vous ai montré dans un des premiers chapitres. Le revoici à nouveau :

Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Bienvenue sur mon site !</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <!--
    <div id="contenu">
      <div id="titre">
        <h1>Bienvenue sur mon site !</h1>
      </div>
    </div>
  </body>
</html>
```

Comme vous le savez, nous allons écrire désormais uniquement entre les balises <body></body>.

## La structure générale

Bien souvent, on structure sa page web à l'aide de blocks <div>, la fameuse balise universelle de type "block" dont je vous ai parlé. On va avoir en général au moins 4 blocks, présentés comme ceci :



1. En haut, l'en-tête (aussi appelé "header"). Cet en-tête contient en général le titre de votre site, sous forme de bannière.
2. En-dessous, vous avez la partie principale de votre page avec :
  - A gauche ou à droite, le menu.
  - De l'autre côté, le corps de la page, c'est-à-dire la partie qui contiendra le texte de votre site.
3. Enfin, on met en général en bas un "pied de page" (aussi appelé "footer"), dans lequel on indique l'auteur du site, éventuellement son e-mail et un petit copyright pour faire classe 😊

Voilà pour les grandes lignes.

En XHTML, on va créer ces 4 blocks à l'aide de 4 <div> différents :

Code : HTML

```
<div id="en_tete">
  <!-- Ici on mettra la bannière -->
</div>

<div id="menu">
  <!-- Ici on mettra le menu -->
</div>

<div id="corps">
  <!-- Ici on mettra le contenu principal de la page (tout le texte quoi) -->
</div>

<div id="pied_de_page">
  <!-- Enfin, on mettra en bas de la page le nom de l'auteur, un copyright... -->
</div>
```

C'est tout simple, on a mis un block <div> pour chaque gros block de la page. On verra tout à l'heure ce qu'on met dedans.

Particularité qui a dû vous choquer : on utilise ici des attributs *id* et non des *class* comme on a eu l'habitude d'en utiliser jusqu'ici. Si vous ne vous souvenez pas de la différence entre un id et une class, retournez voir les premiers chapitres sur le CSS.

Grosso modo, un id et une class c'est quasiment la même chose, sauf qu'on ne peut mettre un id qu'une seule fois dans une page XHTML, contrairement aux class qu'on peut utiliser plusieurs fois. Il y a peu d'id dans une page en général, on en met surtout pour les grands "blocks" du design comme on vient de le faire ici.

Vous devez écrire vos blocks dans l'ordre où ils seront affichés (de haut en bas) dans le code XHTML. En premier l'en-tête, en dernier le pied de page.

Pour ce qui est du menu et du corps qui se trouvent au même niveau, l'ordre entre eux deux n'importe pas. On pourra modifier leur position dans le CSS (menu à gauche, ou menu à droite).

Bon, maintenant que fait-on ? Eh bien on va voir plus en détail ce qu'on met dans chacun des 4 <div> 😊

## L'en-tête

Alors ça, ça va être rapide. En général un en-tête c'est juste une bannière (une image), voire pour ceux qui n'ont pas envie de dessiner un simple titre `<h1>` (titre du site).

Pour ma part, je vais mettre une bannière, par exemple celle-ci :



Pour ceux qui croiraient que je leur aurais menti sur mon niveau en graphisme : non, cette image n'est pas de moi 🤖

Toutes les images que vous allez voir dans ce chapitre sont copyright **zaz et venom** tous droits réservés (je leur ai promis de mettre leurs noms en lettres d'or dans le tuto, une promesse est une promesse 😊). Merci donc à eux pour l'aide importante qu'ils m'ont fournie pour ce chapitre !

Revenons à nos moutons. Comment mettre cette image dans le `<div>` de l'en-tête ? On a 2 possibilités :

- Soit on crée une balise `<img />` à l'intérieur, afin d'insérer notre bannière. C'est une méthode facile pour insérer une image, ça on sait le faire.
- Soit on ne met rien à l'intérieur du `<div id="en_tete">` et on mettra la bannière sous forme d'image de fond tout à l'heure dans le CSS.

La seconde solution n'est pas particulièrement difficile mais un peu surprenante je dois l'avouer. En fait, on va créer un bloc "vide" qui va servir uniquement à afficher une image, et **cette image sera affichée par le CSS**.

Le résultat sera apparemment le même que si on avait utilisé `<img />` mais nous verrons à la fin de ce chapitre que cette technique s'avèrera payante si vous voulez pouvoir changer le design de votre site facilement.

Code : HTML

```
<div id="en_tete">

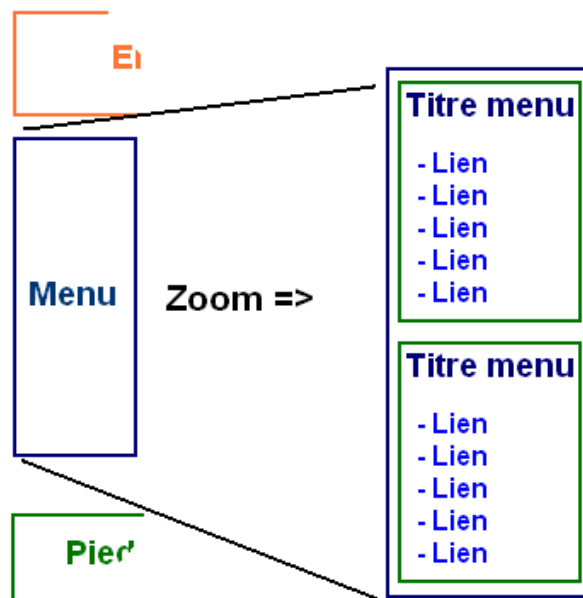
</div>
```

On ne met donc rien entre le `<div>` et le `</div>` (juste un espace à la rigueur 😊)

## Menu et sous-menus

Dans le block du menu (en bleu marine), on va avoir à créer plusieurs sous-blocks (eh oui, ça s'emboîte) pour séparer les différents éléments du menu.

Regardez-bien le schéma ci-dessous : c'est un zoom sur le block "Menu" du schéma de tout à l'heure :



Comme vous le voyez, il contient 2 sous-blocks (cadres verts) qui vont servir de sous-menus. Comment faire cet emboîtement de blocs ? C'est très simple en XHTML :

#### Code : HTML

```
<div id="menu"> <!-- Cadre englobant tous les sous-menus (en bleu marine sur le schéma)

  <div class="element_menu"> <!-- Cadre correspondant à un sous-menu -->
    Texte du premier menu
  </div>

  <div class="element_menu">
    Texte du second menu
  </div>

</div>
```

Il nous suffit d'imbriquer des balises <div>, ces fameuses balises blocks universelles (on les appelle aussi "balises génériques" d'ailleurs)

J'ai mis des commentaires dans le code source pour que vous puissiez bien repérer à quoi correspond chaque <div>. Le premier englobe tous les autres sous-menus (j'ai donné à ces sous-menus une class "element\_menu" pour qu'on puisse les repérer tout à l'heure dans le CSS).

Bien, rajoutons maintenant le texte à l'intérieur du menu pour faire comme dans le dernier schéma.

On va mettre dans chaque "element\_menu" un titre de menu. Vous vous souvenez de la balise qui sert à faire des titres ? Oui, il y a 6 balises (selon l'importance du titre) : de <h1> à <h6>. Comme les titres de menus sont un peu moins importants, je vais les mettre en <h3>.

Ensuite, on va mettre une liste à puces (<ul>) pour avoir un menu bien organisé.

#### Code : HTML

```

<div id="menu">

  <div class="element_menu">
    <h3>Titre menu</h3> <!-- Titre du sous-menu -->
    <ul>
      <li>Lien</li> <!-- Liste des liens du sous-menu -->
      <li>Lien</li>
      <li>Lien</li>
    </ul>
  </div>

  <div class="element_menu">
    <h3>Titre menu</h3>
    <ul>
      <li>Lien</li>
      <li>Lien</li>
      <li>Lien</li>
    </ul>
  </div>

</div>

```

En général, on met souvent des listes à puces pour les menus car cela permet de créer une bonne organisation facilement. Vous avez peut-être remarqué que j'ai marqué des "Lien", mais que je n'ai pas mis de balise `<a href="...">`. C'était juste pour ne pas trop encombrer le schéma de suite, bien entendu qu'on va mettre des liens dans notre menu sinon il ne servirait pas à grand chose



#### Code : HTML

```

<div id="menu">

  <div class="element_menu">
    <h3>Titre menu</h3>
    <ul>
      <li><a href="page1.html">Lien</a></li>
      <li><a href="page2.html">Lien</a></li>
      <li><a href="page3.html">Lien</a></li>
    </ul>
  </div>

  <div class="element_menu">
    <h3>Titre menu</h3>
    <ul>
      <li><a href="page4.html">Lien</a></li>
      <li><a href="page5.html">Lien</a></li>
      <li><a href="page6.html">Lien</a></li>
    </ul>
  </div>

</div>

```

Et voilà un menu tout beau tout propre, avec ses sous-menus !

Vous pouvez, si vous le désirez, ajouter d'autres sous-menus. Il n'y a pas de limite, mais attention à ne pas trop abuser de sous-menus sinon le visiteur risque de se perdre facilement.

Vous n'êtes pas obligés (même si c'est ce qu'on fait le plus souvent) de mettre des liens et une liste à puces dans un menu. Si vous voulez écrire un peu de texte dedans, utilisez la balise de paragraphe `<p></p>`.

## Le corps

Le corps, c'est la partie principale de la page. C'est là qu'il y aura tout le contenu de votre page.

Alors, comme je ne sais pas de quoi votre site va parler, je ne peux pas vous dire ce que vous devez mettre : seuls vous le savez 😊

Je vais quand même vous donner quelques indications pour vous aider à démarrer.

Pour commencer, on va mettre un titre `<h1>` qui sera le titre principal de la page. En général, je mets le même titre dans la balise `<title>` (dans `<head>`) que dans la balise `<h1>`.

Ensuite, on écrira nos paragraphes `<p>` éventuellement séparés par des sous-titres `<h2>` pour un peu mieux structurer son texte 😊

Cela nous donne un code relativement simple pour le corps :

## Code : HTML

```

<div id="corps">
  <h1>Mon super site</h1>

  <p>
    Bienvenue sur mon super site !<br />
    Vous allez adorer ici, c'est un site génial qui va parler de... heu... Je cherche
  </p>

  <h2>A qui s'adresse ce site ?</h2>
  <p>
    A tout le monde ! Si je commence à privilégier certaines personnes, on va m'accuser
    Que vous soyez fans de fusils à pulsion plasma ou de Barbie et Ken, ce site est f
  </p>

  <h2>L'auteur</h2>
  <p>
    L'auteur du site ? Bah, c'est moi, quelle question :-p<br />
    Je vais essayer de faire le meilleur site du monde (ça doit pas être bien compliqué)
    Je prendrai ensuite le contrôle du Monde. Une fois que ce sera fait, j'irai explo
  </p>
</div>

```

Ne faites pas attention au texte que j'ai écrit, c'est juste un petit délire perso pour "remplir" un peu la page 🤪

N'oubliez pas que cette zone est libre. Si vous voulez mettre des images, des liens, d'autres paragraphes, faites-le ! C'est le contenu de votre site après tout 😊

## Allez le dernier : le pied de page

Comme je vous l'ai dit, le pied de page sert à écrire le nom de l'auteur en général, et si ça vous chante vous pouvez mettre un copyright pour faire classe auprès de vos copains (mais ça servira qu'à ça hein 😊)

## Code : HTML

```

<div id="pied_de_page">
  <p>Copyright "Tout pourri Corporation" 2005, tous droits réservés</p>
</div>

```

## Le code XHTML final

Nous avons vu en détail ce que nous devons mettre dans chacun des 4 gros blocks du design. Maintenant, on va assembler tous les codes sur lesquels on a travaillé, en n'oubliant pas de mettre en haut les balises DOCTYPE, <html>, <head>, <body> etc...

Ca nous donne ça :

## Code : HTML



```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Mon super site</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  </head>

  <body>

    <!-- L'en-tête -->

    <div id="en_tete">

    </div>

    <!-- Les menus -->

    <div id="menu">
      <div class="element_menu">
        <h3>Titre menu</h3>
        <ul>
          <li><a href="page1.html">Lien</a></li>
          <li><a href="page2.html">Lien</a></li>
          <li><a href="page3.html">Lien</a></li>
        </ul>
      </div>

      <div class="element_menu">
        <h3>Titre menu</h3>
        <ul>
          <li><a href="page4.html">Lien</a></li>
          <li><a href="page5.html">Lien</a></li>
          <li><a href="page6.html">Lien</a></li>
        </ul>
      </div>
    </div>

    <!-- Le corps -->

    <div id="corps">
      <h1>Mon super site</h1>

      <p>
        Bienvenue sur mon super site !<br />
        Vous allez adorer ici, c'est un site génial qui va parler de... heu... Je
      </p>

      <h2>A qui s'adresse ce site ?</h2>
      <p>
        A tout le monde ! Si je commence à privilégier certaines personnes, on va
        Que vous soyez fans de fusils à pulsion plasma ou de Barbie et Ken, ce si
      </p>

      <h2>L'auteur</h2>
      <p>
        L'auteur du site ? Bah, c'est moi, quelle question :-p<br />
        Je vais essayer de faire le meilleur site du monde (ça doit pas être bier
        Je prendrai ensuite le contrôle du Monde. Une fois que ce sera fait, j'ir
      </p>
    </div>

    <!-- Le pied de page -->

    <div id="pied_de_page">
      <p>Copyright "Tout pourri Corporation" 2005, tous droits réservés</p>
    </div>

  </body>
</html>

```

Eh bien moi je dis : félicitations !

Vous venez de réaliser votre première **vraie page XHTML** ! 😊

Je me doute bien que vous n'auriez pas deviné tout seuls, mais le principal c'est que vous compreniez et que vous suiviez en même

temps que moi. Il n'y a aucune nouveauté dans ce code XHTML, vous avez déjà appris toutes ces balises. En plus, avouez que ce code XHTML n'est pas franchement très long (on faisait bien pire sur le web il y a quelques années).

Cependant, on n'a pas encore fini. En effet, un code XHTML comme ça tout seul sans CSS, ça ne donne pas quelque chose de bien extraordinaire, voyez plutôt :

[Essayer !](#)

QUOOOOIII ??? C'est sur ça que je travaille depuis une heure ?!

Oui, et le plus drôle c'est que ce code XHTML suffit pour réaliser le design. Il ne nous reste qu'à ajouter un peu de CSS, et notre site sera tout beau tout propre ! 😊

Vous ne me croyez pas ?

Faites attention... Ne sous-estimez pas la puissance du CSS, vous pourriez le regretter 😬

## Secundo, le CSS

Nous attaquons maintenant la partie CSS.

C'est une partie très importante, car il faut bien que vous compreniez comment chaque ligne de code modifie le design.

Nous allons donc là encore y aller très "pas à pas" et je vous proposerai de tester la page XHTML avec le CSS que nous avons fait au fur et à mesure.

Quant à vous, si vous voulez comprendre, vous avez tout intérêt à essayer de faire quelques modifications sur le CSS. Si une ligne vous perturbe, essayez de l'enlever tout simplement, et regardez le résultat que ça produit. C'est comme ça que j'ai appris, et je pense que c'est la meilleure façon de faire. Je vous invite donc à faire comme moi 😊

### Centrer le design

Pour commencer, nous allons travailler juste sur la balise <body>. Oui, je sais c'est une balise sur laquelle on n'a pas vraiment travaillé jusqu'ici, mais le principe reste le même pour le CSS.

<body>, je vous le rappelle, c'est la balise qui englobe tout le contenu de votre page web. Si on dit que notre <body> fait tant de pixels de large (pour avoir un design de taille fixe) et qu'on veut le centrer (avec margin:auto;), bah on a déjà pratiquement gagné 😊

Pour la taille de la page, je vais mettre 760 pixels. Pourquoi 760 ? Parce que c'est moins que 800px, et ainsi les visiteurs qui sont en résolution 800\*600 n'auront pas à se déplacer vers la droite pour voir le reste de la page.

Pour ce qui est des marges, un margin:auto suffira à centrer le design.

Je rajouterai une marge en haut (top) et en bas (bottom) de 20 pixels pour éviter que notre page soit trop "collée" avec le haut et le bas du navigateur.

Enfin, je mettrai une petite image de fond (background-image) à la page pour faire un peu moins vide.

Voilà ce que ça devrait donner :

Code : CSS

```
body
{
    width: 760px;
    margin: auto; /* Pour centrer notre page */
    margin-top: 20px; /* Pour éviter de coller avec le haut de la fenêtre du navigateur. E
    margin-bottom: 20px; /* Idem pour le bas du navigateur */
    background-image: url("images/fond.png"); /* Une petite image de fond pour éviter d'av
```

[Essayer !](#)

Permettez-moi d'insister : essayez d'enlever quelques lignes pour observer le résultat sur la page :

- Si vous enlevez le fond (background-image), vous retrouverez un fond blanc de base.
- Si vous enlevez le margin-top ou le margin-bottom, vous verrez que votre page sera un peu trop "collée" avec les bords de la page.
- Si vous enlevez le margin:auto, votre page ne sera plus centrée et se retrouvera à gauche. Notez que le margin:auto ne marche pas sous IE 5 : ces visiteurs-là auront un design placé à gauche, mais ce n'est pas bien grave, vous pouvez me croire il y a pire 😬

## L'en-tête

Comme promis, nous allons afficher la bannière du site à l'aide du CSS (avec *background-image*). On a fait un truc un peu nouveau et apparemment idiot tout à l'heure : on n'a rien mis dans le <div> de l'en-tête. C'est une technique que vous pourrez utiliser si besoin est pour faire en sorte que tout votre design soit compris dans le fichier CSS. Ca sera très pratique (entre autres) pour réaliser un site qui propose plusieurs designs différents.

A nous maintenant d'afficher notre bannière avec uniquement du CSS :

1. La première chose à faire, si on veut que la bannière soit visible, ce sera d'agrandir la taille du block "en\_tete" afin qu'il puisse contenir la bannière entière.  
Comme notre bannière fait 758x100pixels, on va définir une largeur et une hauteur de ces dimensions :  
`width:758px;`  
`height:100px;`
2. Ensuite, on va indiquer quelle image de fond on veut mettre à notre grand block qui est pour l'instant vide :  
`background-image:url("images/banniere.png");`
3. Normalement notre fond n'a la place d'apparaître qu'une seule fois, mais ça ne nous coûte pas grand chose de s'assurer que cette image ne sera pas répétée en mosaïque avec `background-repeat` :  
`background-repeat:no-repeat;`
4. Enfin, c'est un détail mais afin d'éviter que l'en-tête ne soit trop collé avec ce qui se trouvera dessous (le menu et le corps), on va définir une petite marge en-dessous (`margin-bottom`) de quelques pixels :  
`margin-bottom:10px;`

Nous y sommes. Ca donne finalement cela :

Code : CSS

```
body
{
    width: 760px;
    margin: auto;
    margin-top: 20px;
    margin-bottom: 20px;
    background-image: url("images/fond.png");
}

/* L'en-tête */

#en_tete
{
    width: 760px;
    height: 100px;
    background-image: url("images/banniere.png");
    background-repeat: no-repeat;
    margin-bottom: 10px;
}
```

[Essayer !](#)

A partir d'un block vide, nous avons pu afficher notre bannière uniquement en CSS. Désormais, si vous voulez changer la bannière de votre site, il faudra modifier le fichier CSS et non le fichier XHTML. Et ça, c'est une bonne chose vous le comprendrez tout à l'heure 😊

Euh, pourquoi il y a un # devant "en\_tete" ? C'est pas un point normalement qu'on met ???

Non, je vous rappelle pour ceux qui l'auraient oublié :

- Pour une class, on met un "." (point) devant le nom dans le CSS.
- Pour un id, on met un "#" (dièse) devant le nom. Or, regardez le code XHTML, "en\_tete" est un id !

## Le menu

Ca c'est un peu plus difficile.

Il va nous falloir placer le menu "à gauche" du corps, alors que pour l'instant le menu se trouve au-dessus.

Pour faire cela, la technique la plus courante et la plus rapide, c'est d'utiliser la propriété CSS *float*. C'est une propriété un peu particulière comme nous l'avons vu, et elle va être très pratique ici pour placer correctement le menu et le corps.

On va aussi donner une taille de 120 pixels au menu, pour qu'il ne soit pas trop grand.

Ensuite, nous allons travailler sur notre class "element\_menu" (qui correspond à une partie de menu). On va lui mettre une couleur de fond un peu plus gris foncé, une image de fond en repeat-x (qui se répètera uniquement horizontalement). Ensuite, on rajoutera une bordure à chaque élément de menu pour qu'on voie bien ses limites. "2px" solid black ça devrait être bon. Enfin, on mettra un *margin-bottom* à chaque élément de menu pour éviter qu'ils ne soient trop collés les uns par rapport aux autres.

#### Code : CSS

```
body
{
    width: 760px;
    margin: auto;
    margin-top: 20px;
    margin-bottom: 20px;
    background-image: url("images/fond.png");
}

/* L'en-tête */

#en_tete
{
    width: 760px;
    height: 100px;
    background-image: url("images/banniere.png");
    background-repeat: no-repeat;
    margin-bottom: 10px;
}

/* Le menu */

#menu
{
    float: left; /* Le menu flottera à gauche */
    width: 120px; /* Très important : donner une taille au menu */
}

.element_menu
{
    background-color: #626262;
    background-image: url("images/motif.png");
    background-repeat: repeat-x;

    border: 2px solid black;


    margin-bottom: 20px; /* Pour éviter que les éléments du menu ne soient trop collés */
}
```

[Essayer !](#)

Encore une fois, si vous voulez comprendre tout ce charabia je vous conseille fortement d'essayer d'enlever quelques lignes du CSS pour voir ce que ça change.

Comme vous pouvez le constater si vous testez ce code, c'est pas encore la joie mais ça s'améliore. On distingue déjà bien nos bouts de menus, par contre le corps se retrouve sous le menu une fois qu'il l'a dépassé par le bas.

C'est normal car c'est comme ça que fonctionne un *float*, rappelez-vous ce petit schéma simple sur une image flottante :



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vitae lorem imperdiet lacus molestie molestie. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu purus. Phasellus metus lorem, blandit et, posuere quis, tincidunt vitae, ante. Vivamus consequat mauris a diam. Vivamus nibh erat, hendrerit nec, aliquet ut, hendrerit quis, nunc. Vestibulum et turpis et elit tempor euismod.

Comment faire alors pour que le corps de la page ne passe pas sous le menu flottant ?

Il nous faudra rajouter une marge à gauche sur le corps (*margin-left*). On fera ça tout à l'heure quand on travaillera sur le corps. Si ça peut vous rassurer, c'est une astuce toute simple et pourtant je ne la connaissais pas il y a encore 5 minutes ! ... Comment ça ça ne vous rassure pas ? 🤔

Bon, on n'a pas encore fini de travailler sur les menus. On va rajouter quelques effets. Je ne vais pas m'attarder à expliquer ça trop en détail, c'est du tout simple, et c'est juste pour améliorer la présentation.

Le principal qu'il faut bien comprendre ici, c'est que je me sers des "imbrications de balises". Vous vous souvenez, quand on écrit par exemple :  
`.element_menu ul`  
... cela signifie "Ce CSS va concerner les balises `<ul>` se trouvant à l'intérieur d'un `element_menu`".

**Code : CSS**

```

body
{
    width: 760px;
    margin: auto;
    margin-top: 20px;
    margin-bottom: 20px;
    background-image: url("images/fond.png");
}

/* L'en-tête */

#en_tete
{
    width: 760px;
    height: 100px;
    background-image: url("images/banniere.png");
    background-repeat: no-repeat;
    margin-bottom: 10px;
}

/* Le menu */

#menu
{
    float: left;
    width: 120px;
}

.element_menu
{
    background-color: #626262;
    background-image: url("images/motif.png");
    background-repeat: repeat-x;

    border: 2px solid black;

    margin-bottom: 20px;
}

/* Quelques effets sur les menus */

.element_menu h3 /* Tous les titres de menus */
{
    color: #B3B3B3;
    font-family: Arial, "Arial Black", "Times New Roman", Times, serif;
    text-align: center;
}

.element_menu ul /* Toutes les listes à puces se trouvant dans un menu */
{
    list-style-image: url("images/puce.png"); /* On change l'apparence des puces */
    padding: 0px; /* Tous les côtés ont une marge intérieure de 0 pixels */
    padding-left: 20px; /* ... mais on modifie ensuite la marge de gauche, donc celle-là
margin: 0px; /* Idem pour margin, ça nous évite d'avoir à en écrire 4 (margin-left, n
margin-bottom: 5px; /* Même chose que tout à l'heure, on modifie ensuite juste margin
}

.element_menu a /* Tous les liens se trouvant dans un menu */
{
    color: #B3B3B3;
}

.element_menu a:hover /* Quand on pointe sur un lien du menu */
{
    background-color: #B3B3B3;
    color: black;
}

```

[Essayer !](#)

C'est déjà plus joli non ?

En plus, l'effet du :hover sur les liens du menu est très facile à réaliser, donc c'est tout bénéf' pour nous 😊

Petite précision quand même, c'est un détail mais je préfère que tout soit clair. Vous avez vu que j'ai enchaîné un "padding" (qui modifie toutes les marges intérieures) et un "padding-left" (qui modifie uniquement la marge intérieure gauche). En fait c'est une technique pour gagner de la place. Il faut lire les lignes dans l'ordre :

1. Premièrement, on modifie toutes les marges et on leur met à toutes la valeur "0px" avec *padding*
2. Deuxièmement, on modifie la marge de gauche. Elle valait 0px, désormais on change sa valeur et on lui met 20px avec *padding-left* (on dit qu'on "écrase" l'ancienne valeur)

Si j'avais voulu, j'aurais pu faire à la place :

```
padding-top:0px;  
padding-bottom:0px;  
padding-right:0px;  
padding-left:20px;
```

Mais ça aurait été plus long, et comme tout bon webmaster je suis une feignasse 🤔

## Le corps

Pour le corps, il va être important de définir de bonnes marges.

Le problème qu'on veut régler en priorité, c'est le texte du corps qui passe "sous" le menu quand il est trop long. Comme je vous l'ai expliqué, c'est un comportement normal car on a fait un *float:left* sur le menu.

Comment éviter que le texte passe sous le menu alors ?

Il suffit de modifier la marge "à gauche" du corps. C'est une marge extérieure, donc un *margin-left*.

On lui met une valeur suffisamment grande pour "pousser" le corps sur la droite, afin qu'il ne passe plus sous le menu (donc une valeur supérieure à la largeur du menu). Ici par exemple, on va mettre une valeur de 140px. Comme par magie, vous allez voir, le corps sera correctement placé !

Ne nous arrêtons pas en si bon chemin 😊

On va rajouter un *margin-bottom* afin que le pied de page en-dessous ne soit pas trop "collé" au corps :

```
margin-bottom:20px;
```

On va aussi mettre un petit padding (marge intérieure) sur tous les côtés afin que le texte ne colle pas trop avec les bords du corps :

```
padding:5px;
```

On va aussi mettre une couleur de fond, une petite image de fond qui se répète horizontalement, une bordure... Bref, tout ça c'est de la déco 😊

Code : CSS

```
body
{
    width: 760px;
    margin: auto;
    margin-top: 20px;
    margin-bottom: 20px;
    background-image: url("images/fond.png");
}

/* L'en-tête */

#en_tete
{
    width: 760px;
    height: 100px;
    background-image: url("images/banniere.png");
    background-repeat: no-repeat;
    margin-bottom: 10px;
}

/* Le menu */

#menu
{
    float: left;
    width: 120px;
}

.element_menu
{
    background-color: #626262;
    background-image: url("images/motif.png");
    background-repeat: repeat-x;

    border: 2px solid black;

    margin-bottom: 20px;
}

/* Quelques effets sur les menus */

.element_menu h3
{
    color: #B3B3B3;
    font-family: Arial, "Arial Black", "Times New Roman", Times, serif;
    text-align: center;
}

.element_menu ul
{
    list-style-image: url("images/puce.png");
    padding: 0px;
    padding-left: 20px;
    margin: 0px;
    margin-bottom: 5px;
}

.element_menu a
{
    color: #B3B3B3;
}

.element_menu a:hover
{
    background-color: #B3B3B3;
    color: black;
}

/* Le corps de la page */

#corps
{
    margin-left: 140px; /* Une marge à gauche pour pousser le corps, afin qu'il ne passe
```



[Essayer !](#)

Il nous reste quand même quelques petits trucs à décorer dans notre corps. Par exemple, les titres : on va changer leur police pour mettre une "Arial" plus jolie je trouve sur les titres.

On va aussi rajouter une image de fond sur les titres h2, avec *background-repeat:no-repeat*;, ce qui signifie que le fond ne se répètera pas (il restera donc à gauche).

Pour éviter que le texte du titre ne s'écrive sur l'image de fond, on met un *padding-left* de quelques pixels. Comme ça, on aura une image d'engrenage devant chaque titre <h2> automatiquement !

**Code : CSS**

```
body
{
    width: 760px;
    margin: auto;
    margin-top: 20px;
    margin-bottom: 20px;
    background-image: url("images/fond.png");
}

/* L'en-tête */

#en_tete
{
    width: 760px;
    height: 100px;
    background-image: url("images/banniere.png");
    background-repeat: no-repeat;
    margin-bottom: 10px;
}

/* Le menu */

#menu
{
    float: left;
    width: 120px;
}

.element_menu
{
    background-color: #626262;
    background-image: url("images/motif.png");
    background-repeat: repeat-x;

    border: 2px solid black;

    margin-bottom: 20px;
}

/* Quelques effets sur les menus */

.element_menu h3
{
    color: #B3B3B3;
    font-family: Arial, "Arial Black", "Times New Roman", Times, serif;
    text-align: center;
}

.element_menu ul
{
    list-style-image: url("images/puce.png");
    padding: 0px;
    padding-left: 20px;
    margin: 0px;
    margin-bottom: 5px;
}

.element_menu a
{
    color: #B3B3B3;
}

.element_menu a:hover
{
    background-color: #B3B3B3;
    color: black;
}

/* Le corps de la page */

#corps
{
    margin-left: 140px;
```

[Essayer !](#)

Sympathique n'est-ce pas ? 😊

## Le pied de page (et on a fini !)

Allez, le dernier pour la route et c'est fini, promis ! 😊

Pour le pied de page, rien de compliqué ni d'extraordinaire, ce sont des modifications similaires au reste. On met une couleur de fond, on met une bordure, on vérifie si les marges nous plaisent... Notez que je fais un "clear: both" pour supprimer l'effet flottant du menu. Cela permet de s'assurer que le pied de page est bien sous le menu.

Code : CSS

```
/*
Design d'exemple du Site du Zéro
Réalisé par zaz, venom et mateo21
<lien url="http://www.siteduzero.com">http://www.siteduzero.com</lien>
*/

body
{
    width: 760px;
    margin: auto;
    margin-top: 20px;
    margin-bottom: 20px;
    background-image: url("images/fond.png");
}

/* L'en-tête */

#en_tete
{
    width: 760px;
    height: 100px;
    background-image: url("images/banniere.png");
    background-repeat: no-repeat;
    margin-bottom: 10px;
}

/* Le menu */

#menu
{
    float: left;
    width: 120px;
}

.element_menu
{
    background-color: #626262;
    background-image: url("images/motif.png");
    background-repeat: repeat-x;

    border: 2px solid black;

    margin-bottom: 20px;
}

/* Quelques effets sur les menus */

.element_menu h3
{
    color: #B3B3B3;
    font-family: Arial, "Arial Black", "Times New Roman", Times, serif;
    text-align: center;
}

.element_menu ul
{
    list-style-image: url("images/puce.png");
    padding: 0px;
    padding-left: 20px;
    margin: 0px;
    margin-bottom: 5px;
}

.element_menu a
{
    color: #B3B3B3;
}

.element_menu a:hover
{
    background-color: #B3B3B3;
    color: black;
}
```

[Essayer !](#)

C'était notre dernier CSS, le design est terminé ! (ouf 😊)

Et voilà notre site fin prêt. Bon ce design "Mechanical Spirit" inventé pour l'occasion ne vous conviendra sûrement pas. Remarquez, c'est un peu fait exprès 😊

Maintenant que le design est fini, vous savez comment procéder, quelles sont les grandes lignes à suivre. C'est à vous de créer le design entier de votre site, d'y mettre vos couleurs, vos images de fond, vos effets, vos polices... Bref, tout ça vous savez le faire maintenant, depuis le temps que je vous rabâche et re-rabâche du CSS 😊

Vous êtes autorisés (et invités) à vous inspirer de mon CSS. Je sais que ça ne coule pas de source quand on débute et qu'on fait son premier design, donc je comprends tout à fait que toutes ces lignes de code vous donnent la nausée.

La bonne nouvelle, c'est qu'avec de la pratique non seulement vous n'aurez plus la nausée, mais surtout vous saurez créer un design entier sans l'aide de personne ! 😊

---

## Attendez, c'est pas fini !

Vous ne croyiez quand même pas que j'allais vous lâcher aussi rapidement 😊

J'ai quelques informations "en vrac" à vous livrer. Ce ne sont pas des petits détails pour la plupart, et vous allez peut-être découvrir de nouvelles choses, donc soyez attentifs y'en a plus pour longtemps 😊

### Rappel : insérer un CSS dans le XHTML

Ça peut paraître idiot mais vous l'avez peut-être déjà oublié : pour que notre fichier XHTML s'affiche avec la présentation indiquée dans le CSS, il faut rajouter la ligne suivante entre <head> et </head> :

```
<link rel="stylesheet" media="screen" type="text/css" title="Exemple"
href="design_mecanique.css" />
```

Cette ligne indique que l'on utilise le fichier "design\_mecanique.css" pour la présentation.

### Changez le CSS, changez le design !

En modifiant le CSS, on peut changer tout le design sans toucher au code XHTML !

En fait, regardez à nouveau attentivement le code XHTML final qu'on a écrit : il ne contient aucune information sur le design. C'est tout juste s'il dit qu'il y a un en-tête, un menu qui contient tels liens, un corps qui contient tel texte etc. **Et c'est tout !**

Ca, c'est ce que je me tue à répéter depuis le premier chapitre de ce cours (et j'ose espérer que depuis le temps ça commence à rentrer 😊) : le XHTML c'est le contenu de votre page, le CSS c'est la présentation de votre page.

En gros, changer de CSS c'est comme changer de vêtement : on modifie son look (son apparence), mais la personne à l'intérieur est toujours la même 😊

Pour illustrer ce que je viens de vous dire, j'ai demandé à zaz et venom de me faire un second design sans toucher au code XHTML. Seul le CSS a été modifié, et regardez tout ce qu'on peut changer sur un site rien qu'à coup de CSS :

[Essayer !](#)

Petite précision : nous avons porté un peu moins d'attention sur ce second design, et si les couleurs rose-barbie vous donnent la nausée, dites-vous bien que nous aussi :). C'était juste pour illustrer que l'on peut faire un design très différent en modifiant le CSS.

Regardez la différence. Un seul fichier XHTML, 2 fichiers CSS, des résultats très différents :

*Avec design\_mecanique.css :*



Avec `design_poney_rose.css`



Regardez le code source XHTML, c'est le même que tout à l'heure. Avec le CSS, on a :

- Changé la couleur de fond de la page
- Changé l'image de fond de l'en-tête, donc on a changé la bannière
- Changé les bordures
- Changé le menu de position (il suffit de remplacer le `float:left` par un `float:right` !)
- etc...

**Laissez le visiteur choisir le design**

Une dernière chose que je souhaite vous montrer : *les feuilles de styles alternatives*.

En fait, si vous avez fait plusieurs designs pour votre site (comme c'est le cas pour moi là), vous pouvez proposer aux visiteurs de changer d'un seul coup le design.  
Pour cela, vous devez rajouter des balises `<link />` entre `<head>` et `</head>`.

En plus de la balise que je vous ai rappelé il y a 2 minutes (qui indiquera le design "par défaut" de votre site), vous pourriez rajouter cette ligne à la suite :  
`<link rel="alternate stylesheet" media="screen" type="text/css" title="Design rose pour les filles" href="design_poney_rose.css" />`

Cette ligne indique qu'il existe une feuille de style alternative ("alternate stylesheet"). Le visiteur pourra choisir le design de votre site qui lui plaît.

Pour info, si vous utilisez Firefox il suffit d'aller dans le menu "Affichage > Style de la page" et de choisir le design que vous voulez.

Les sites qui proposent des feuilles de style alternatives sont plutôt rares. Beaucoup de webmasters ne savent même pas que ça existe, d'autres ont tout simplement un peu la flemme de réaliser plusieurs designs 😊

Vous pouvez mettre autant de feuilles de style alternatives que vous le désirez. Il vous faudra écrire une balise `<link rel="alternate stylesheet" />` par design différent que vous proposez.

Un exemple, et on s'arrêtera là :

#### Code : HTML

```
<head>
  <title>Bienvenue sur mon site !</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

  <!-- Ci-dessous le design "par défaut" du site -->
  <link rel="stylesheet" media="screen" type="text/css" title="Mécanique" href="design_
  <!-- Ci-dessous les designs alternatifs que vous proposez -->
  <link rel="alternate stylesheet" media="screen" type="text/css" title="Rose pour les
  <link rel="alternate stylesheet" media="screen" type="text/css" title="Sombre et terr
  <link rel="alternate stylesheet" media="screen" type="text/css" title="Aquatique" href=
</head>
```

Exceptionnellement, vous n'avez pas de Q.C.M. pour ce chapitre. En effet, le meilleur moyen de vérifier que vous avez compris était de suivre en même temps que moi et de faire vos propres petites manipulations.

Oui je sais, ça a l'air lourd tout ce bazar à première vue. J'espère vous avoir livré le maximum d'informations pour que vous compreniez bien la méthode.

Néanmoins, ne vous affolez pas en vous disant "Mince ! J'aurais jamais pu deviner tout ça !". Moi aussi c'est ce que je me suis dit quand j'ai appris à faire ça la première fois, mais avec un peu de pratique ça ira de plus en plus vite.  
Je reconnais qu'en XHTML / CSS la création d'un design est assez laborieuse, surtout au début. La bonne nouvelle, c'est qu'on ne fait pas de nouveau design tous les jours quand même 😊