

Formation XHTML / CSS

Introduction

Ce cours est, en grande partie récupéré sur un tutoriel trouvé sur le site du zéro : www.siteduzero.com/tutoriel-3-13666-apprenez-a-creer-votre-site-web.html

Ceci dans le mépris le plus élémentaire des règles du copyright. Donc ne pas diffuser.

Le HTML

Le HTML pour HyperText Markup Language n'est pas un langage de programmation. C'est un langage de présentation qui sert à mettre en forme un texte et où l'on peut glisser des liens hypertextes.

En HTML, les commandes écrites entre < > sont appelées tags (Balises en Français). Ils ne sont pas affichés à l'écran mais donnent des directives à votre navigateur. Les tags vont toujours - disons, la plupart du temps - par deux, il y a le tag ouvrant et le tag fermant

Exemple, pour écrire en gras (Bold en Anglais) on encadre le texte par le tag et on le ferme par (le même tag, mais précédé de la barre de division)

Exemple :

```
Petit <b>padawan</b> deviendra grand
```

S'affichera :

```
Petit padawan deviendra grand
```

Si on oublie de fermer le tag, la directive reste valable jusqu'à la fin du document :

```
Si il <b>ne se fait pas manger avant
```

S'affichera :

```
Si il ne se fait pas manger avant
```

Exercice : Afficher une page avec un navigateur

- Ouvrir un éditeur (Ex : Notepad sous Windows)
- Écrire « Bonjour le monde »
- Enregistrer sous «exercice1.html »
- Avec un navigateur (Firefox, Opéra, MIE, ...) Ouvrir le fichier selon les trois méthodes suivantes :
 - Faire « Fichier / Ouvrir un fichier ... » et retrouver exercice1.html
 - Double-cliquer sur le fichier .html
 - Écrire l'adresse absolue du fichier dans la barre de navigation selon la syntaxe suivante : "file:///localhost/C/.../exercice1.html"

Le XHTML

XHTML est un langage de balisage servant à écrire de pages pour le World Wide Web. Conçu à l'origine comme le successeur d'HTML, XHTML se fonde sur la syntaxe définie par XML, plus récente, mais plus exigeante que celle définie par SGML sur laquelle repose HTML : il s'agit en effet de présenter un contenu affichable non seulement par les ordinateurs classiques, mais également sans trop de dégradation par des PDA bien moins puissants.

En résumé, c'est la nouvelle génération de langage WEB.

Tant qu'a en apprendre un, autant apprendre le plus récent...

Les CSS

Le langage CSS (Cascading Style Sheets : feuilles de style en cascade) sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web depuis les années 2000.

Les outils

Firefox

Le programme qui interprète les directives données par le XHTML et les CSS pour en faire une jolie page web, c'est le navigateur.

Même si ce ne devrait pas être le cas, l'affichage d'une même page, sera différent d'un navigateur à un autre.¹

Les exercices de ce cours sont testé sur Firefox 3.

C'est donc celui que je vous recommande d'installer : <http://www.firefox.fr/>

Alors, Opéra, Chrome, Safari, Konqueror, ou le-nouveau-truc-de-geek-à-la-mode ... Comme vous voulez mais ne venez pas râler parce que « M'sieur, chez moi, ça ne marche pas »²

Notepad++

Un éditeur a coloration syntaxique est quand même bien pratique.

Même remarque que précédemment : PSPad, Notepad2 ou Kéké-edit : Comme vous voulez.

Mais veillez bien à ce que ce soit un **éditeur** de texte et non un machin-qui-vous-rajoute-du-texte-et-de-la-pub-quand-on-sauvegarde.

Je vous recommande Notepad++ (OpenSource et en Français) :

http://sourceforge.net/project/showfiles.php?group_id=95717&package_id=102072

Les balises

Type de balises

Les balise par paire

Ce sont les plus courantes. On écrit la première balise, on tape du texte, puis on "ferme" la balise en la réécrivant avec un slash ("/") devant. Par exemple :

<titre>Bienvenue sur mon site **!</titre>**

Contrairement au HTML le XHTML est sensible à la case³ : On écrit en minuscules

Les balises seules

Elles sont un peu plus rares, mais il y en a quand même. On s'en sert en général pour insérer un élément dans une page.

1 Voir même carrément port'nawak pour ceux qui utilisent Internet Explorer 6.

2 Rigolez ... Je vous parie qu'il y en a un qui me fais le coup avant a fin du trimestre !

3 Différence Majuscule / Minuscules

Ce type de balise se termine toujours par un slash ("/"), mais cette fois le slash se trouve à la fin de la balise précédé d'un espace. Par exemple la balise qui permet d'insérer une image :

```
<image />
```

Les attributs

Les attributs sont un moyen de donner des précisions sur une balise. Des paramètres.

Par exemple, la balise `<image />` . Pour préciser quelle image on insère :

```
<image nom="soleil.jpg" />
```

Dans le cas d'une balise fonctionnant "par paire", on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante. Par exemple, ce code indique que la citation est de Neil Armstrong et qu'elle date du 21 Juillet 1969 :

```
<citation auteur="Neil Armstrong" date="21/07/1969">
```

C'est un petit pas pour l'homme. un bond de géant pour l'humanité.

```
</citation>
```

Résumé :

- Les noms des balises et attributs s'écrivent toujours en minuscules (ex. : "citation, auteur, date")
- les attributs ne sont pas répétés dans la balise fermante.
- Les valeurs des attributs peuvent contenir des majuscules (ex. : "Neil Armstrong")

Structure générale d'une page

Doctype

Évidemment, on commence par une exception ...

La première ligne que doit contenir une page c'est celle ci :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

On vas faire simple : C'est une ligne obligatoire, à recopier telle quelle.

<html>

La balise `<html>` indique au navigateur qu'il a bien reçu une page en html

Il se ferme par `</html>` et les deux balises doivent encadrer tout le document

La balise html admet deux paramètres :

```
xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
```

Le premier est – aussi – à recopier tel quel.

Le second à pour valeurs possibles "fr", "en", "li", ... selon que votre page soit en français, anglais, lituanien ...

<head>

Les balises `<head>` et `</head>` encapsulent l'entête du document.

Cet entête ne contient aucune information affichée.

Elle concerne principalement les balises `<title>` et `<meta>`.

<title>

Les balises `<title>` et `</title>` contiennent le titre de la page (celui qui s'affiche dans le cadre bleu de la fenêtre du navigateur)

<meta />

Attention : C'est une balise seule (Pas de </meta>)

Il existe plusieurs type de balises meta, les autres seront abordés plus tard.

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

Seule celle ci est obligatoire : elle sert à indiquer que vous allez taper des caractères spécifiques au français (èèâê etc...).

C'est le 3^{ème} (et dernier) machin incompréhensible à recopier tel quel.

<body>

<body> et </body> encadrent le contenu de la page. C'est la partie affichée du document.

<p>

P pour Paragraphe. Elle permet d'écrire du texte.

Attention, contrairement au HTML ou on pouvait mettre du texte hors balise, en XHTML c'est « interdit ».

Après </p> le navigateur saute une ligne.

Résumé

La structure générale d'une page Web sera toujours comme suit :

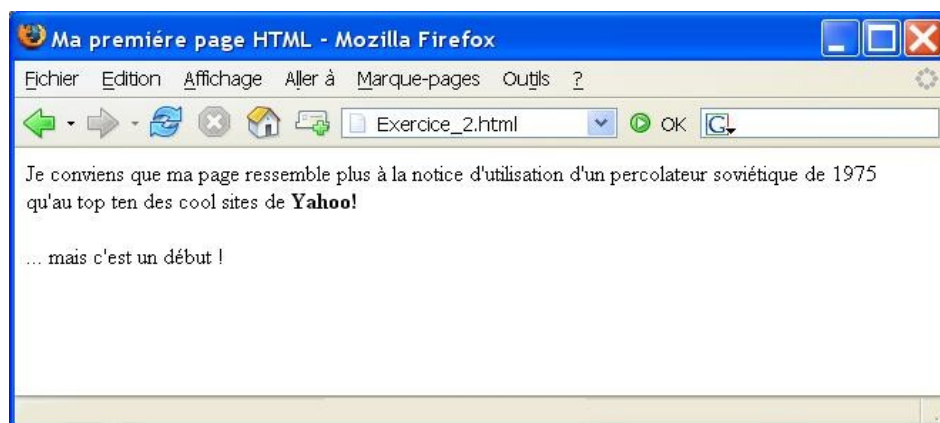
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <title>Ma première page web</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <p>Les plans secrets pour fabriquer une bombe atomique dans son évier.</p>
  </body>
</html>
```

Note :

- Les indentation sont facultatives ... mais conseillées.
- On peut passer des lignes, ce n'est pas affiché.

Exercice : Ma première page en XHTML.

A l'aide de la balise <p> pour décrire un paragraphe et pour mettre en gras reproduire la page suivante le plus fidèlement possible :



Organiser son texte

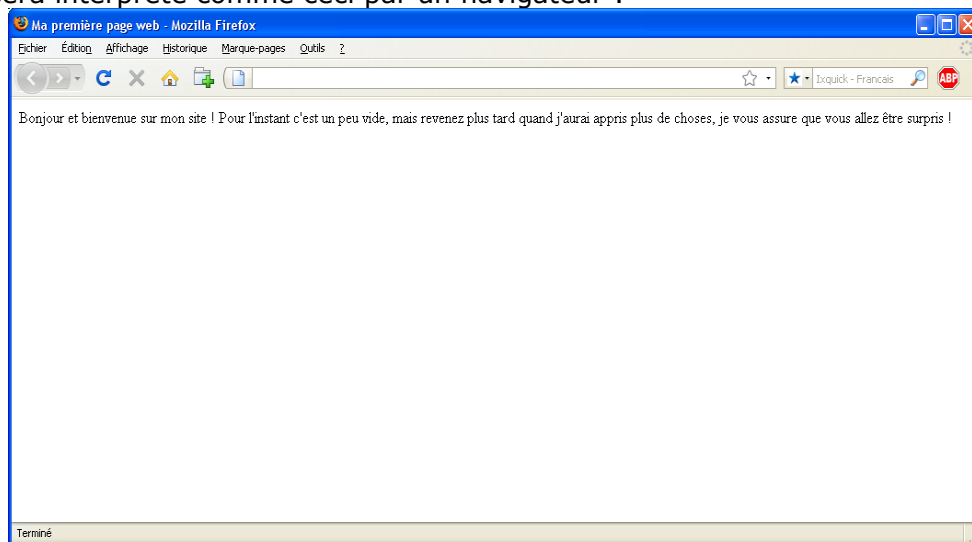
Différence entre le texte « de l'éditeur » et le texte « du navigateur »

Voici le body d'une page html :

```
<body>
  <p>
    Bonjour et bienvenue sur mon site !

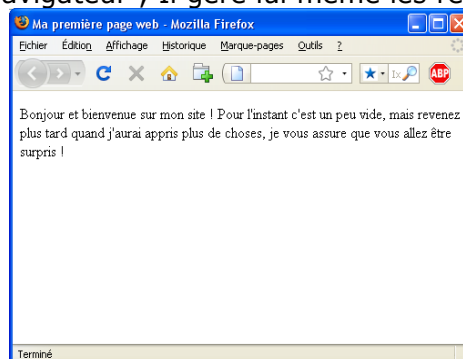
    Pour l'instant c'est un peu vide, mais revenez plus tard quand j'aurai
    appris plus de choses,
    je vous assure que vous allez être surpris !
  </p>
</body>
```

Cette page sera interprétée comme ceci par un navigateur :



On remarque que les espaces à gauche de « bonjour ... » et de « pour ... » ont été ignorés. Ainsi que tous les sauts de ligne.

Et si on diminue la fenêtre du navigateur ; Il gère lui-même les retours à la ligne ...



Sauter une ligne

Si on veut sauter une ligne :

il faut le demander explicitement au navigateur avec la balise **
**.

Attention cette balise doit obligatoirement être dans un paragraphe. Pas en dehors.

Commentaire

La balise de commentaire est la seule à ne pas être symétrique :

<!-- ceci est un commentaire qui ne sera pas affiché **-->**

Comme le navigateur n'affiche ni les retours chariot ni le contenu des balises commentaire, n'hésitez pas à en abuser.

Afficher un titre

Il existe 6 balises pour afficher des titres en XHTML : de **<h1>** à **<h6>**

Ces 6 balises correspondent à des importances de titre :

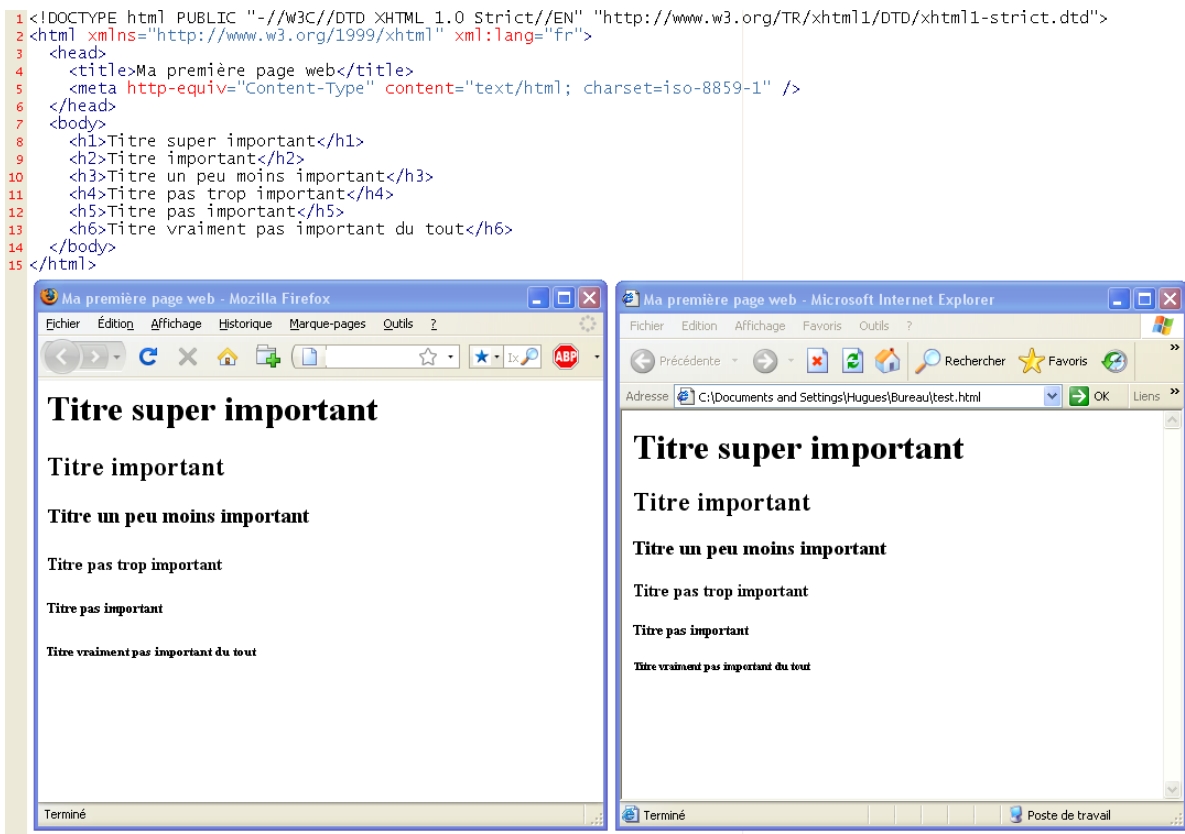
<h1> = Gros titre très important

...

<h6> = Petit titre à peine mis en avant

Il n'y a pas d'ordre : Vous pouvez utiliser un **<h3>** sans avoir mis de **<h1>** ou de **<h2>** avant ...

Un exemple de page web affiché avec Firefox 3 (à gauche) et IE 6 (à droite)



Et si je veux un titre centré écrit en rouge et souligné ?

Nous verrons cela avec les CSS. Ce qui est important, c'est de retenir que **<h1>** ne signifie pas "Times New Roman, taille 16 pt", mais "Titre important".

Ce sont les CSS, qui dirons que "Les titres importants sont centrés, rouges et soulignés"

Mettre, un peu, en valeur le texte :

La balise (pour embedded) permet de mettre un peu en avant une partie du texte.
Par défaut (sans CSS) le texte est juste en italique.

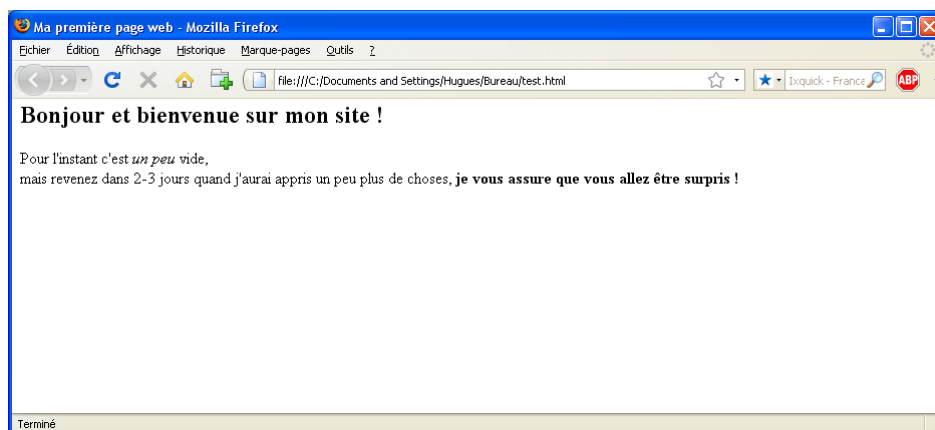
Mettre, beaucoup, en valeur le texte :

La balise permet de mettre plus en avant une partie du texte.
Par défaut (sans CSS) le texte est juste en gras.

Ex :

```
<!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>Ma première page web</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <h2>Bonjour et bienvenue sur mon site !</h2>
    <p>
      Pour l'instant c'est <em>un peu</em> vide, <br />
      mais revenez dans 2-3 jours quand j'aurai appris un peu plus de choses,
      <strong>je vous assure que vous allez être surpris !</strong>
    </p>
  </body>
</html>
```

Donne :



Mettre en exposant ou en indice <sup> et <sub>

<sup> pour mettre en indice. ex 1^{er} donne : 1^{er}

<sub> pour mettre en exposant. ex CO₂ donne : CO₂

Les citations <q> et <blockquote>

<q> pour quote (Citation) et <blockquote> pour Citation de bloc ...

En XHTML, on distingue 2 types de balises :

- Les balises de type inline : ce sont des balises que l'on utilise à l'intérieur d'un paragraphe. C'est le cas notamment des balises , , <q> etc...
- Les balises de type bloc : ces balises servent à structurer la page en plusieurs "blocs". C'est le cas notamment des balises <p>, <h1>, <h2>, <blockquote>... Chacune de ces balises constitue un bloc séparé.

Par défaut (Sans CSS) <q> met le texte entre guillemets et <blockquote> crée un retrait de quelques centimètres à gauche.

Les acronymes

En français un acronyme est une suite d'initiales qui est devenu un nom (Radar, ovni, laser, sida ...).

En anglais c'est – juste – un mot composé d'initiales, comme par exemple XHTML, SNCF, CD ...

La balise **<acronym>** permet de fournir une aide instantanée sur un mot.

<acronym> fonctionne avec un attribut (title) :

<p>Le **<acronym title="eXtensible HyperText Markup Language">**XHTML**</acronym>** est le langage qui permet de créer un site web.**</p>**

Le XHTML est le langage qui permet de créer un site web.

eXtensible HyperText Markup Language

Si vous laissez la souris sur XHTML, la description de l'acronyme apparaît.

Vous noterez encore une petite différence entre IE et Firefox : Firefox souligne en pointillés l'acronyme, tandis qu'IE... ne fait rien

Les liens Hypertext

Les liens internes à la page

C'est la balise **<a>** qui gère tous les types de liens

Pour faire, par exemple, un lien pointant vers le bas de la page, il faut deux choses.

La cible

En bas de la page (ou en haut ...), on ajoute l'attribut **id=** à une balise qui encadre le texte ciblé **<p id="nom">** (par exemple, mais on aurait put utiliser un **<h2 id="nom">**)

Le lien.

Autre part dans le document, on encadre le texte qui fera lien entre **** et ****

Exemple :

****Aller en bas****

<p id="fondpage">Fond**</p>**

Les liens internes au site

C'est le même fonctionnement, mais encore plus simple car il n'y a pas de NAME à préciser (puisque la page-cible sera simplement désignée par son nom de fichier.)

****Passer à la page suivante****

Il faut juste s'assurer que le fichier soit bien à l'endroit spécifié, le plus simple étant de mettre toutes les pages du site dans un même répertoire.

Attention : Contrairement à Windows, Linux est sensible à la case. PageDeux.html et Pagedeux.html sont deux fichiers différents.

Attention : Si les fichiers ne se trouvent pas dans le même dossier :

Il faudra préciser le chemin relatif : "dossier/cible.html"

Ne confondez pas avec les antislash (\) de Windows : sur Internet, on utilise TOUJOURS des slash (/)

Les liens externes

Fonctionne comme les liens internes, mais au lieu de référencer un fichier sur le même serveur, il faut préciser une URL.

```
<a href="http://www.whitehouse.gov/">Pour s'entraîner au hacking</a>
```

On précise TOUJOURS le protocole dans un lien (http://).

Si vous faites un lien vers un site qui comporte une adresse avec des paramètres (&), comme :
`http://www.site.com/?data=15&name=toto&source=ms`

Vous devrez remplacer tous les (&) par (&#amp;) comme ceci :
`http://www.site.com/?data=15&#amp;name=toto&#amp;source=ms`

Cela est nécessaire pour que votre page web respecte les normes du XHTML.

Les liens vers un e-mail

Pour lancer automatiquement le programme de Mail par défaut et remplir quelques champs, il suffit d'utiliser la syntaxe suivante :

```
<a href="mailto:xxxxxxx@xxxxxx.xx">texte</a>
```

Pour proposer un sujet par défaut, il faut libeller le lien ainsi :
`<a href="mailto:president@elysee.fr?subject=Salut Nico">Ici</a>`

Les liens qui ouvrent une nouvelle page

Même si beaucoup de sites le font, en XHTML il n'est plus possible d'ouvrir les liens dans une nouvelle fenêtre.

C'est une décision volontaire : Ceci afin de ne pas gêner la navigation du visiteur qui n'a pas forcément envie de voir 50 fenêtres de son explorateur ouvertes.

C'est au visiteur de décider lui-même s'il veut ouvrir le lien dans une nouvelle fenêtre. Il fera Maj + Clic sur le lien (fonctionne sur IE et Firefox), ou encore mieux : Ctrl + Clic (pour ouvrir dans un nouvel onglet dans Firefox).

Enfin, chose importante : les personnes non-voyantes⁴ qui surfent sur le web sont perturbées par l'ouverture d'une nouvelle fenêtre de leur navigateur. Quand une nouvelle fenêtre s'ouvre, il n'est plus possible de faire "Précédente".

Lien vers une ancre située dans une autre page

L'idée, c'est de faire un lien qui ouvre une nouvelle page ET qui amène directement à une ancre située plus bas sur cette page.

En pratique c'est assez simple : il suffit faire suivre le nom de la page d'un dièse (#), suivi du nom de l'ancre.

Par exemple :

```
<a href="cible.html#rollers">
```

vous amènera sur la page cible.html, directement au niveau de l'ancre appelé "rollers".

... Si cette ancre existe.

⁴ Valable aussi pour les handicapés qui ont un écran de 4 pouces (Smartphones ...)

Les images

Les formats d'image

- Le **jpeg** (ou jpg) convient très bien pour les photos.
- Le **png** est le plus récent. Ce format est adapté "à tout ce qui n'est pas une photo". Le PNG gère très bien la transparence.
- Le **gif** est un format assez vieux (et propriétaire) mais qui reste très utilisé par habitude. Aujourd'hui, le PNG est bien meilleur que le GIF : les images sont plus légères et la transparence est meilleure.
Le format GIF est limité à 256 couleurs (alors que le PNG peut aller jusqu'à plusieurs millions de couleurs). Le format GIF conserve quand même un avantage que le PNG n'a pas : il peut être animé. (C'est l'image qui est animée en elle même, pas de traitements particuliers au niveau du xhtml)

Le **bmp** est à bannir : car non compressés, donc trop gros, donc pas adaptés au web.

La balise

C'est une balise de type seule (comme **
**).

Elle peut prendre plusieurs attributs, et 2 d'entre eux sont indispensables :

- **src** : il indique où se trouve l'image que l'on veut insérer. Le nom de fichier est un nom relatif. Ainsi, si votre image est dans un sous-dossier "images" vous devrez taper :
src="images/fleur.png"
- **alt** : pour "texte alternatif". On doit toujours indiquer un texte qui décrit ce que contient l'image. Ce pour 4 raisons :
 - Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée
 - Ce texte sera affiché à la place de l'image sur les navigateurs de personnes handicapées (non-voyants)
 - Ce texte est indexé par les moteurs de recherche (pour leur section Image)
 - Sinon, votre page ne sera pas "XHTML valide".par exemple : **alt="Une fleur"**

Note : On peut, pour l'attribut src indiquer une url (**src="http://www.site.com/fleur.png"**), mais cela ralentit l'affichage (Le navigateur doit se connecter au site une fois par image).

Voici un exemple d'insertion d'image :

```
<p>Voici une photo que j'ai prise lors de mes dernières vacances :<br />
  
</p>
```

Note : Évitez les accents, majuscules et espaces dans vos noms de fichiers et de dossiers. Si votre image ne s'affiche pas, c'est très généralement parce que le chemin est incorrect !

Note : les images se mettent obligatoirement à l'intérieur d'un paragraphe (**<p></p>**). Si vous ne le faites pas, comme pour "alt" la page s'affichera peut-être correctement mais elle ne sera alors plus valide.

Les liens via une image

Pour avoir une image qui sert de lien hypertexte, il suffit de combiner la balise **<a>** et la commande **** :

```
<p>Cliquez sur la vache pour nous rejoindre</p>
<a href="pagedeux.Html"></a>
```

Facile non ?

Les infobulles sur les images

L'attribut fait pour afficher une bulle d'aide est le même que pour les liens, c'est title. Cet attribut est facultatif.

```
<p>Voici une photo que j'ai prise lors de mes dernières vacances :<br />
    
</p>
```

A nouveau Internet Explorer ne suit pas les règles du W3C : En l'absence de title, il utilise alt pour afficher un infobulle. Ce n'est pas normal.

Premiers CSS

Le xhtml sert à gérer le contenu (ce que l'on écrit).

Les CSS servent à gérer la présentation (comment on l'écrit).

Ce principe de séparation de la forme et du fond est à la base de la création de page web.

Pour commencer, un peu de motivation :

Allez jeter un oeil au site zengarden : www.csszengarden.com/tr/francais/

Le principe de ce site : C'est une même page xhtml qui est conjugué avec des css différents.

(Cliquez sur la colonne de droite pour en changer)

Impressionnant non ?

(Le fichier xhtml sans css : www.csszengarden.com/zengarden-sample.html)

Ou écrire les CSS ?

Dans un fichier .css

le plus souvent on écrit du code CSS dans un fichier spécial ayant l'extension .css (contrairement aux fichiers XHTML qui ont l'extension .html).

C'est – de loin - la méthode la plus pratique car :

Elle permet proposer facilement plusieurs designs différents à vos visiteurs.

Elle permet de regrouper les styles pour toutes les pages de votre site.

Si vous utilisez Notepad++, pensez à aller dans le menu "Langage / CSS" pour activer la coloration du CSS. Vous penserez à enregistrer votre fichier en .css au lieu de .html.

Concrètement : Il faudra ajouter une ligne dans votre fichier .html entre les balises **<head>** **</head>** comme ceci :

```
<!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>Exemple d'utilisation de CSS externe</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css" title="Design"
        href="design.css" />
  </head>
  <body>
```

```
<p>Cette page comporte une feuille de style externe. C'est la meilleure méthode
à utiliser quand on fait du CSS.</p>
</body>
</html>
```

La balise **<link />** comporte plusieurs attributs. Vous pouvez en modifier deux d'entre eux :

- title : c'est le nom que vous donnez à votre feuille de style (Pas de contraintes)
- href : c'est l'emplacement où se trouve votre feuille de style sous forme de lien relatif. Dans cet exemple le CSS se trouve dans le même dossier, mais je vous recommande de le mettre dans un sous-dossier "styles/" par exemple.

Dans la section <head>

Il est aussi possible de taper du CSS directement à l'intérieur même du fichier XHTML, entre les balises **<head>** **</head>**. Vous devrez y mettre une balise **<style>** **</style>** à l'intérieur, comme ceci :

```
<style type="text/css">
/* Vous taperez votre code CSS ici */
</style>
```

Directement dans la balise

Cette méthode consiste à ajouter un attribut style sur les balises pour leur appliquer un style particulier. Cet attribut fonctionne sur toutes les balises. Par exemple sur un titre :

```
<h1 style="/* Votre style pour cette balise ici */">Titre</h1>
```

Je vous rappelle que seule la première méthode vous permettra d'avoir de n'écrire qu'une fois chaque style.

Appliquer un style à des balises

Le contenu d'une feuille de style CSS ressemble à ça :

```
balise1
{
    propriete: valeur;
    propriete: valeur;
}
balise2
{
    propriete: valeur;
    propriete: valeur;
    propriete: valeur;
    propriete: valeur;
}
balise3
{
    propriete: valeur;
}
```

- Des noms de balises dont on veut modifier l'apparence. Par exemple, pour modifier l'apparence de tous les titres **<h1>**, écrire h1
- Des propriétés CSS : Par exemple la propriété **color** qui permet d'indiquer la couleur du texte, **font-size** pour la taille du texte ...
- Les valeurs : à chaque propriété CSS on doit indiquer une valeur. Par exemple, pour la couleur, il faut indiquer le nom de la couleur. Pour la taille, il faut indiquer quelle taille on veut ...

Chaque propriété est suivie du symbole "deux-points" (:) puis de sa valeur correspondante. Enfin, chaque ligne se termine par un point-virgule (;)

Exemple :

```
p
{
    color: blue;
    font-size: 18px;
}
```

Cela signifie : "Tous les paragraphes sont en bleu avec une taille de 18 pixels."

Note : Pas d'espace entre "18" et "px", sinon le code CSS ne fonctionnera pas.

Appliquer un style à plusieurs balises

Si les styles de 2 balises doivent avoir la même présentation. Il suffit de séparer les noms des balises par des virgules, comme ceci :

```
h1, h2
{
    color: red;
}
```

Commentaires dans les css

Pour ouvrir un commentaire /*

Pour fermer un commentaire */

Exemple :

```
/* -----
essai.css
Ma première feuille de style
-----*/
h1, h2
{
    color: red; /* tout les titres sont en rouge */
}
```

L'attribut class

C'est le cœur des css. Il sert à personnaliser des présentation.

Ce que je vous ai montré jusqu'ici a un défaut : cela implique par exemple que TOUS les paragraphes soient écrits par exemple en bleu / 18 pixels.

Il existe, bien sur, un attribut spécial qui fait que seulement certains paragraphes (ou certains titres, ou certaines citations) soient écrits d'une manière différente : L'attribut class

Il fonctionne sur toutes les balises :

```
<h1 class="" > </h1>
<p class=""></p>
<img class="" />
```

Comme valeur, on met ... ce qu'on veut.

A condition de mettre le pendant de cette valeur dans le fichier css.

Ex : (Coté xhtml)

```
<!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>Exemple d'utilisation de CSS externe</title>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" media="screen" type="text/css" title="Design"
      href="design.css" />
</head>
<body>
  <h1>Découverte du CSS</h1>
  <p>
    Bonjour<br />
    Je suis une page XHTML <em>apparemment</em> banale, mais je sers en fait
    de test de fichier <acronym title="Cascading Style Sheets">CSS</acronym> pour les
    tutoriels d'<a href="http://www.arrobe.fr">Arrobe</a>
  </p>
  <h2>Ce n'est que du blabla</h2>
  <p>
    Comme l'a dit Neil Armstrong un 20 juillet 1969 : <q
    class="important">C'est un petit pas pour l'homme, un grand pas pour
    l'humanité</q><br />
    Il a également déclaré un 12 février 1978 : <q>Tiens, j'ai envie de frites
    ...</q><br />
    Bizarrement, cette seconde phrase n'est pas passé à la postérité.
  </p>
  <p class="important">
    Quoi, c'est n'importe-quoi ce que j'écris ? On s'en fiche, c'est pour
    tester le fichier .css !
  </p>
</body>
</html>

```

Ex : (Coté css)

```

/* -----
Feuille de style
-----*/
h1, h2
{
    color: blue;
}

.important
{
    color: red;
    font-size: 14px;
    text-decoration: blink;
}

```

Note : Pour monter qu'il s'agit d'une classe et nom d'une balise, on le précède d'un point.

Les balises universelles

Il arrivera parfois que vous ayez besoin d'appliquer une class à certains mots qui ne sont pas entourés par des balises. (class étant un attribut, vous ne pouvez le mettre que sur une balise.)

On a inventé les balises-qui-ne-servent-à-rien :

- **** : c'est une balise de type inline (une balise qui se met à l'intérieur d'un paragraphe, comme , , <q>).
- **<div>** : c'est une balise de type bloc. (Comme <p>, <h1> etc...) Elle crée un nouveau "bloc" dans la page, et provoque donc obligatoirement un retour à la ligne.

Exemple :

Code : HTML

```
<p>Comme l'a dit <span class="nom">Neil Armstrong</span> un certain 20 juillet 1969...</p>
```

Code : CSS

```
.nom
{
    color: blue;
}
```

Imbrications de balises

Il est possible d'indiquer l'ordre d'imbrication des balises dans le CSS.

Si, par exemple, nous souhaiterions définir un style uniquement pour les balises **** qui se trouvent à l'intérieur d'un titre **<h3>**. On doit écrire ceci :

```
h3 em /* => tous les em situés à l'intérieur d'un h3 */
{
    color: blue;
}
```

Il suffit de séparer les 2 noms de balises par un espace, et non pas par une virgule comme on l'a fait plus haut.

L'ordre est important ! Si l'on avait mis "em h3" cela aurait voulu dire "Tous les **<h3>** situés à l'intérieur de balises ****".

Autre exemple (même css):

```
<h3>L'imbrication, c'est <em>pratique</em></h3>
<p>Cet exemple vous montre comment fonctionne l'imbrication.<br />Le mot "pratique"
dans le titre sera écrit en bleu, mais pas celui-ci : <em>pratique</em>.</p>
```

Donnera :

L'imbrication, c'est *pratique*

Cet exemple vous montre comment fonctionne l'imbrication.

Le mot "pratique" dans le titre sera écrit en bleu, mais pas celui-ci : *pratique*.

On peut aussi mélanger des balises et des class :

```
p .important
```

Ce qui signifie : "Toutes les class **"important"** contenues dans des balises **<p>**".

Dernier exemple (tordu) :

```
blockquote p strong, h1 .important
```

Attention : La virgule "coupe" la ligne en deux, elle signifie "ET".

Soit : "Toutes les balises **** contenues dans un **<p>** elles-mêmes contenues dans un **<blockquote>** ET toutes les class **"important"** contenues dans un titre **<h1>**".

Formatage du texte en CSS

On a déjà vu que la propriété CSS qui permet de changer la taille du texte est font-size.

En pixels

On a déjà vu comment indiquer la taille du texte en pixels :

Un pixel c'est un « point écran ». c'est une mesure à la fois extrêmement précise et un peu rigide (Un affichage en pixels)

Pour avoir un texte de 16 pixels de hauteur, Écrire : `font-size: 16px;`

```
p
{
    font-size: 14px; /* Paragraphes de 14 pixels */
}
h1
{
    font-size: 22px; /* Titres de 22 pixels */
}
```

En centimètres

Si vous êtes allergique aux pixels, sachez qu'il est aussi possible d'indiquer une taille en centimètres ("2cm" par exemple) et en millimètres ("14mm" par exemple).

Note : Indiquer la taille du texte en pixels, centimètres et millimètres est certes très pratique et très précis. Mais cela ne tiens pas compte de la taille de l'écran de l'utilisateur, ni des préférences (certains préfèrent avoir du texte plutôt gros, d'autres plutôt petit etc...)

En relatif

... c'est-à-dire en écrivant "gros", "très gros", "petit", "minuscule". (en anglais, bien sûr).

Liste des différentes valeurs :

- xx-small : minuscule
- x-small : très petit
- small : petit
- medium : moyen
- large : grand
- x-large : très grand
- xx-large : euh... gigantesque

Exemple (css)

```
.rikiki
{
    font-size: xx-small;
}
.trespetit
{
    font-size: x-small;
}
.petit
{
    font-size: small;
}
.moyen
{
    font-size: medium;
}
.grand
{
    font-size: large;
}
.balaize
{
    font-size: x-large;
}
```

```

}
.supermegagigatresgrand
{
    font-size: xx-large;
}

```

La taille du texte n'est pas exactement identique selon le navigateur... mais ce n'est pas grave, car dans tous les cas "xx-small" correspond à un texte très petit, et "xx-large" à un texte très grand.

En pourcentage

C'est une autre façon d'indiquer de manière relative la taille du texte.

On doit préciser la taille du texte par rapport à la taille normale de la police. Soit : "100%" signifie "Taille normale". Si vous mettez "130%", le texte aura une taille 1,3 fois plus grande. De même pour réduire : "80%" et votre texte aura une taille 0,8 fois plus petite.

Note : On peut aussi utiliser les « em » avec $1\text{em} = 100\%$ $1.3\text{em} = 130\%$...

Les Polices

L'une des conséquences du fonctionnement des pages web est que, comme c'est le navigateur qui calcule l'aspect de la page, si la police demandée n'est pas sur l'ordinateur ...

Le navigateur se débrouille avec les polices par défaut.

Pour une fois, Firefox et IE se sont entendus pour rendre standards sept fonts, qui sont donc Arial, **Arial Black**, Comic Sans MS, Courier New, **Impact**, Times New Roman (qui est la fonte par défaut), et Verdana.

La propriété CSS qui permet d'indiquer la police est font-family :

```
font-family: police;
```

Pour éviter qu'il n'y ait de problème si l'internaute n'a pas la même police que vous, on met en général plusieurs noms de police, séparés par des virgules :

```
font-family: police1, police2, police3, police4;
```

Le navigateur essaiera d'abord de mettre la police1. S'il ne l'a pas, il essaiera de mettre la police2 et ainsi de suite.

Si le nom de la police comporte des espaces, vous devez l'entourer de guillemets, Ex "Arial Black".

Alignement

On utilise la propriété text-align, et on indique l'alignement désiré :

- left : le texte sera aligné à gauche (c'est le réglage par défaut).
- center : le texte sera centré.
- right : le texte sera aligné à droite.
- justify : le texte sera "justifié". (C'est à dire faire en sorte qu'il prenne toute la largeur possible sans laisser d'espace blanc à la fin des lignes)

Exemple :

```
h1
{
```

5 1.3em pas 1,3em.

```

    text-align: center; /* Pour centrer le titre */
    font-family: "Arial Black", Arial, Verdana, serif;
}
blockquote
{
    text-align: justify; /* La citation sera justifiée */
}
.signature
{
    text-align: right; /* Pour aligner à droite ma signature */
    font-family: "Comic Sans MS", Georgia, "Times New Roman", serif;
    font-size: 80%;
}

```

Note : Vous ne pouvez pas modifier l'alignement du texte d'une balise inline . L'alignement ne fonctionne que sur des balises de type bloc (p, div, blockquote, h1, h2, ...).

L'indentation

l'indentation c'est la mise en retrait du texte. Exemple de paragraphe indenté :

Bien que cela :
plus en plus inquié
telles hostilités ven

On utilise la propriété **text-indent**. On doit indiquer quelle est la taille du retrait en pixels, en centimètres, en millimètres...

```

p
{
    text-indent: 30px; /* Les paragraphes commenceront 30 pixels sur la droite */
    text-align: justify; /* Ils seront justifiés */
    font-size: large;
}

```

L'avantage en CSS, est qu'il suffit de dire une fois pour toutes : "Je veux que tous mes paragraphes soient indentés de 30 pixels sur la droite" et à chaque nouveau paragraphe, celui-ci sera automatiquement mis en retrait. A noter 2 propriétés similaires :

- **word-spacing** : l'espace entre les mots (en pixels).
- **letter-spacing** : l'espace entre les lettres (en pixels).

Effets de style

Mettre en italique

On utilise **font-style**, qui peut prendre 2 valeurs :

- **italic** : le texte sera mis en italique.
- **normal** : le texte sera normal (par défaut). Cela vous permet d'annuler une mise en italique. (Par exemple, si vous voulez que les textes entre ne soient plus en italique)

Mettre en gras

La propriété CSS pour mettre en gras est **font-weight**, et prend les valeurs suivantes :

- **bold** : le texte sera en gras.
- **normal** : le texte sera écrit normalement (par défaut).

Les majuscules en CSS (font-variant)

La propriété **font-variant**, prend uniquement 2 valeurs différentes :

- **small-caps** : le texte sera écrit en petites capitales.
- **normal** : le texte sera écrit normalement (par défaut).

Les majuscules en CSS (text-transform)

Il y a une seconde propriété CSS qui travaille sur les majuscules : **text-transform** :

- **uppercase** : tout le texte sera écrit en majuscules.
- **lowercase** : tout le texte sera en minuscules.
- **capitalize** : la première lettre de chaque mot sera en majuscule.
- **none** : pas de transformation (par défaut).

Exemple (xhtml) :

```
<h1>Je suis très en colère...</h1>
<p>...mais je vais essayer de rester calme. Enfin, si possible, mais je promets
rien hein...<br />
<span class="pousser_une_gueulante">ahhhh !!! non cette fois je craque ! qui c'est
qui a mis de la mayo dans mes frites !??<br />sacrilège !</span></p>
<p class="chuchoter">CECI EST UN PARAGRAPHE QUE JE CHUCHOTE, ALORS QUE POURTANT
DANS LE CODE XHTML IL EST ECRIT EN MAJUSCULES</p>
```

Exemple (CSS) :

```
h1
{
    text-align: center;
    font-family: Arial, "Times New Roman", Verdana, serif;
    text-transform: capitalize; /* Les premières lettres des mots du titre seront en
    majuscules */
}
.pousser_une_gueulante
{
    text-transform: uppercase; /* Si j'ai envie de me faire entendre, je mets en
    majuscules */
}
.chuchoter
{
    text-transform: lowercase;
    font-style: italic; /* Le texte chuchoté sera en minuscules et italique */
}
```

Résultat :

Je Suis Très En Colère...

...mais je vais essayer de rester calme. Enfin, si possible, mais je promets rien hein...
 AHHHH !!! NON CETTE FOIS JE CRAQUE ! QUI C'EST QUI A MIS DE LA MAYO DANS MES FRITES !??
 SACRILÈGE !

ceci est un paragraphe que je chuchote, alors que pourtant dans le code xhtml il est écrit en majuscules

Un peu de déco ?

Une propriété CSS porte bien son nom : **text-decoration**

Voici les différentes valeurs qu'elle peut prendre :

- **underline** : souligné.
- **line-through** : barré.
- **overline** : ligne au-dessus.
- **blink** : clignotant. (Ne marche pas sous Internet Explorer).
- **none** : normal (par défaut).

Les couleurs

On a plusieurs possibilités pour indiquer une couleur.

La propriété qui permet de changer la couleur du texte est **color**.

La méthode la plus simple pour choisir une couleur est de taper son nom (in english, oeuf corse).

Le défaut est qu'il n'existe que 16 couleurs dites "standard". D'autres couleurs officielles existent, mais elles ne fonctionneront pas forcément pareil sur tous les navigateurs.

Pour utiliser les autres couleurs (Les navigateurs en reconnaissent 16 millions ...) on utilise la notation hexadécimale.

Un nom de couleur en hexadécimal, ça ressemble à ça :

color: #FF5A28;

On doit toujours commencer par écrire un dièse (#), suivi de 6 lettres ou chiffres allant de 0 à 9 et de A à F.

Les couleurs s'écrivent sous la forme "#RRVVB"

Elles sont au format RVB, c'est à dire que les deux premiers chiffres hexadécimaux commandent le niveau de rouge, les suivants le vert et les derniers le bleu.

- "#FFFFFF" correspond à Blanc (tout au max.)
- "#000000" à noir
- "#FF0000" à rouge
- "#C0C0C0" à un gris clair
- ...

Les logiciels de dessin, comme The Gimp, vous indiquent les couleurs en hexadécimal. Il vous sera alors facile de copier-coller le code hexadécimal d'une couleur dans votre fichier CSS.

Couleur	Aperçu
white	
silver	
gray	
black	
red	
maroon	
lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

Le fond

Contrairement à ce qu'on pourrait croire, le fond ne désigne pas forcément le fond de toute une page web. On peut aussi appliquer un fond uniquement aux titres, ou aux paragraphes, ou encore à certains mots d'un paragraphe.

Il faut tout d'abord savoir qu'il existe 2 types de fonds :

- Les fonds comportant une couleur
- Les fonds comportant une image de fond

La couleur de fond

Pour indiquer une couleur de fond, on utilise la propriété CSS **background-color**. Elle s'utilise de la même manière que la propriété **color**, c'est-à-dire que vous pouvez taper le nom d'une couleur ou l'écrire en notation hexadécimale.

Pour indiquer la couleur de fond de la page web, il faut travailler sur la balise **<body>**. Eh oui, **<body>** correspond à toute la page web, c'est donc en modifiant sa couleur de fond que l'on changera la couleur de fond de la page web.

Exemple :

```
body /* On travaille sur la balise body, donc sur TOUTE la page */
{
  background-color: black; /* Le fond de la page sera noir */
  color: white; /* Le texte de la page sera blanc */
}
```

Resultat :



Vous remarquez que tous les paragraphes **<p>** et titres **<h1>** sont devenus blancs.

Ce phénomène s'appelle l'héritage : En CSS, si vous appliquez un style à une balise, toutes les balises qui se trouvent à l'intérieur de cette balise prendront le même style.

Voici un exemple :

```
body
{
  background-color: black;
  color: white; /* Toutes les balises contenues dans body verront leur texte
               coloré en blanc... */
}
h1
{
  color: red; /* ... sauf si je demande expressément de changer la couleur par la
               suite */
}
.surligne /* Un style qui permet par exemple de surligner un texte */
{
  background-color: yellow;
  color: black; /* Le texte surligné sera écrit en noir, parce que le blanc sur
               fond jaune on voit rien ;o) */
}
```

L'image de fond

Tout comme pour la couleur de fond, l'image de fond ne s'applique pas forcément à la page entière, on peut tout aussi bien mettre une image de fond aux titres, paragraphes, citations etc...

La propriété permettant d'indiquer une image de fond est **background-image**. Exemple :

```
background-image: url("fond.png");
```

Attention lorsque vous mettez une adresse en relatif dans le fichier CSS ! L'adresse de l'image doit être indiquée par rapport au fichier .css et non pas par rapport au fichier .html.

Il existe par ailleurs une propriété CSS qui permet de "fixer" le fond, pour ne pas qu'il bouge en même temps que le texte. C'est **background-attachment** :

- **fixed** : l'image de fond reste fixe.
- **scroll** : l'image de fond défile avec le texte (par défaut).

Exemple :

```
body
{
  background-image: url("../images/neige.png");
  background-attachment: fixed; /* Le fond restera fixe */
}
```

la répétition de l'image de fond

Cette propriété s'appelle **background-repeat** et peut prendre ces valeurs :

- **no-repeat** : le fond ne sera pas répété. L'image sera donc unique sur la page.
- **repeat-x** : le fond sera répété uniquement sur la première ligne, horizontalement.
- **repeat-y** : le fond sera répété uniquement sur la première colonne, verticalement.
- **repeat** : le fond sera répété (par défaut).

La position de l'image de fond

On peut indiquer où doit se trouver l'image de fond avec **background-position**. Cette propriété n'est intéressante que si vous avez mis "**background-repeat: no-repeat;**" (un fond qui ne se répète pas).

Vous devez donner à **background-position** 2 valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe si vous appliquez le fond à un paragraphe) :

background-position: 30px 50px;

Il est aussi possible de mettre ces valeurs en anglais :

- **top** : en haut.
- **bottom** : en bas.
- **left** : à gauche.
- **center** : centré.
- **right** : à droite.

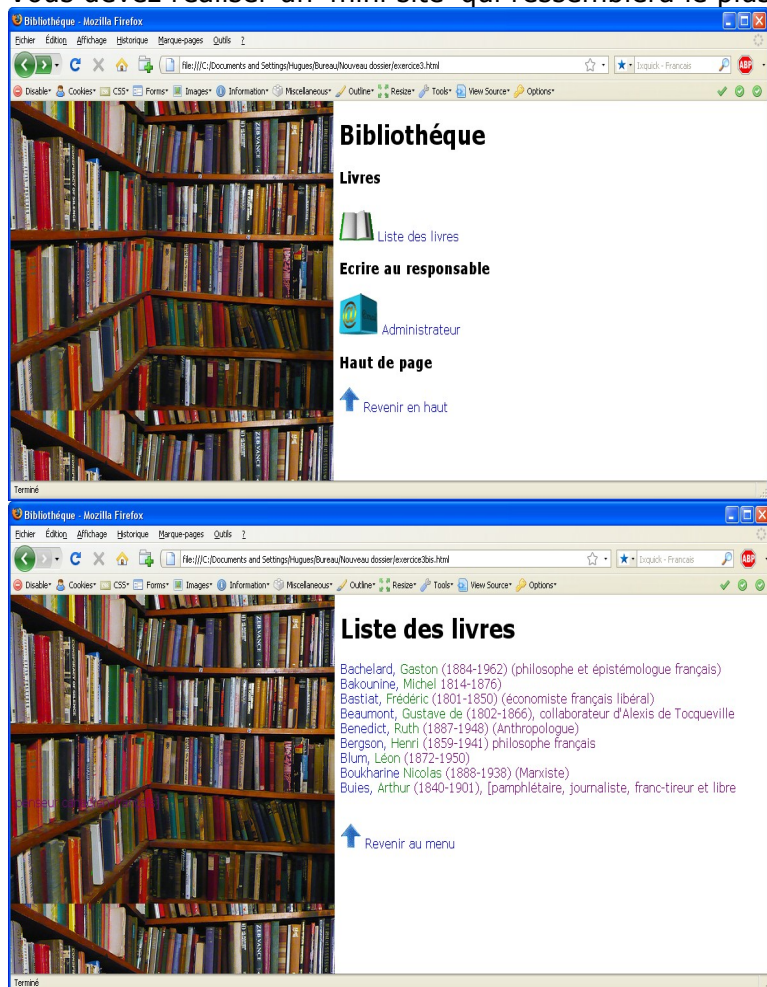
Il est possible de combiner ces mots. Par exemple, pour aligner une image en haut à droite:

background-position: top right;

Exercice récapitulatif

Cahier des charges

Vous devez réaliser un 'mini site' qui ressemblera le plus possible à ceci :



Contraintes :

- La première page s'appelle « exercice3.html »
- Le fait de cliquer sur le livre ou sur « Liste des livres » amène sur la page « exercice3bis.html »
- Le fait de cliquer sur Le cube ou sur « Administrateur » ouvre le logiciel de messagerie et crée un nouveau courriel ayant pour destinataire « administrateur@monsite.fr »
- Le fait de cliquer sur la flèche ou sur « Revenir en haut » aligne l'affichage sur le livre.
- Le xhtml des fichiers exercice3.html et exercice3bis.html doit être valide.
- La feuille de style, commune aux deux pages, s'appelle exercice3.css et doit être valide.

Mathos

Vous trouverez les images sur www.arrobe.fr/xhtml/

Pour valider vos pages xhtml : <http://validator.w3.org/>

Pour valider vos css : <http://jigsaw.w3.org/css-validator/>

Les pseudo-formats

Les pseudos-formats CSS permettent des présentations dynamiques

C'est une information que l'on rajoute après le nom de la balise (ou de la class) dans le CSS

Au passage de la souris :hover

Hover pour « hand over ». C'est donc une présentation qui ne s'activera qu' « au survol du pointeur de la souris »

Exemple :

```
a
{
    text-decoration: none; /* Les liens ne seront plus soulignés */
    color: red; /* Les liens seront en rouge au lieu de bleu */
    font-style: italic; /* Les liens seront en italique */
}
a:hover /* Quand le visiteur pointe sur le lien */
{
    text-decoration: underline; /* Le lien sera souligné quand on pointera dessus */
    color: green; /* Le lien sera écrit en vert quand on pointera dessus */
}
```

La bonne nouvelle, c'est que vous pouvez appliquer le pseudo-format ":hover" à n'importe quelle balise.

La mauvaise nouvelle, c'est que pour Internet Explorer 6 le ":hover" ne marche que sur les liens. Sous Internet Explorer 7, il marche en théorie sur toutes les balises, mais d'après mes tests il y a quand même quelques bugs.

Encore un exemple.

Ce CSS ci sert à mettre en avant le paragraphe survolé par la souris.

```
p:hover
{
    background-color: #CFE1EB;
    text-indent: 20px;
}
```

Au moment du clic :active

Le pseudo-format `:active` permet d'appliquer un style particulier au moment du clic. Le lien gardera cette apparence très peu de temps : juste pendant que vous cliquez avec le bouton de la souris. En clair, ce n'est pas forcément toujours bien visible.

Note : On peut « factoriser » plusieurs pseudos-formats :

```
a:hover, a:active
{
    background-color: #FFCC66;
}
```

Lorsque l'objet est actif :focus

Par principe, le focus est exclusif : Un seul des objets graphique (Lien hypertexte, champ de formulaire, bouton, ...) peut avoir le focus.

Si un nouvel objet acquiert le focus, le précédent le perd.

La touche Tabulation permet de faire passer le focus d'un objet à l'autre.

Quand la page a déjà été vue :visited

Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue. Par défaut, le navigateur colore le lien en un violet immonde.

Note : La liste des liens visités est géré par l'historique du navigateur.

La première lettre :first-letter

First-letter sert à modifier automatiquement l'apparence de la première lettre du texte contenu dans une balise.

La première ligne :first-line

Un autre pseudo-format intéressant permet de modifier cette fois la première ligne.

Attention : La notion de 1er ligne dépend de la taille de l'écran ... et de la taille de la fenetre dans l'écran.

```
p
{
    text-indent: 20px;
}
p:first-letter /* La première lettre de chaque paragraphe */
{
    font-weight: bold; /* En gras */
    font-size: 1.2em; /* Ecrit légèrement plus gros que la normale */
    color: blue; /* En bleu */
}
p:first-line /* La première ligne de chaque paragraphe */
{
    font-variant: small-caps; /* En petites capitales */
}
```

Avant / Après :before & :after

! Ne sont pas supportés par Internet Explorer

Une démonstration vaut mieux qu'un discours

Coté xhtml :

```

<h1>Q.C.M</h1>
<p>Ceci est une démonstration des pseudos-formats :before et :after</p>
<p class="question">Que signifie html</p>
<p class="reponse">ce sont des lettres tirés au sort</p>
<p class="reponse"><strong>h</strong>ey <strong>t</strong>hat's <strong>m</strong>y
    <strong>l</strong>unch</p>
<p class="reponse"><strong>h</strong>ello <strong>t</strong>o <strong>m</strong>y
    <strong>l</strong>ove</p>
<p class="reponse"><strong>h</strong>yper <strong>t</strong>ext
    <strong>m</strong>ark-up <strong>l</strong>anguage</p>
<p class="question">Que signifie css</p>
<p class="reponse">ce sont des lettres tirés au sort</p>
<p class="reponse"><strong>c</strong>ontent
    <strong>s</strong>crambling<strong>s</strong>ytem</p>
<p class="reponse"><strong>c</strong>atholic <strong>s</strong>ocial
    <strong>s</strong>ervices</p>
<p class="reponse"><strong>c</strong>ascading <strong>s</strong>tyle
    <strong>s</strong>heets</p>

```

Coté css :

```

.question:before
{
    content: "Question : ";
    font-weight: bold; /* "Question" sera en gras */
    color: blue; /* "Question" sera en bleu */
}
.question:after
{
    content: url("../images/question.gif"); /* Mettre un fichier image de "?" */
}
.reponse:before
{
    content: url("../images/check.gif");
}
strong
{
    color: red;
}

```

Coté résultat :

Q.C.M

Ceci est une démonstration des pseudos-formats :before et :after

Question : Que signifie html ?



ce sont des lettres tirés au sort



hey that's my lunch



hello to my love



hyper text mark-up language

Question : Que signifie css ?



ce sont des lettres tirés au sort



content scrambling system



catholic social services



cascading style sheets

Attention : N'oubliez pas que le chemin relatif vers l'image se fait à partir de la position du fichier CSS et non pas du fichier XHTML.

Les listes à puces

Coté xhtml

Liste non ordonnée

On vas faire court. Ceci :

- Fraises
- Framboises
- Cerises

S'obtient avec cela :

```
<ul>
  <li>Fraises</li>
  <li>Framboises</li>
  <li>Cerises</li>
</ul>
```

Avec **** pour Unordored list (Liste non ordonnée) et **** pour chaque élément de cette liste.

Si on imbrique plusieurs niveaux; la présentation suit :

```
<ul>
  <li>Lundi</li>
  <ul>
    <li>matin</li>
    <li>midi</li>
    <li>soir</li>
  </ul>
  <li>Mardi</li>
  <ul>
    <li>matin</li>
    <li>midi</li>
    <li>soir</li>
  </ul>
</ul>
```

- Lundi
 - matin
 - midi
 - soir
- Mardi
 - matin
 - midi
 - soir

Liste ordonnée

On remplace juste le **** par un **** :

```
<ol>
  <li>janvier</li>
  <li>février</li>
  <li>mars</li>
  <li>avril</li>
</ol>
```

1. janvier
2. février
3. mars
4. avril

Liste de définitions

On combine 3 types de balises :

<dl> (c'est l'abréviation de "Definition List") pour le type de liste

<dt> (Definition table) pour le mot à définir

<dd> (Définition data) pour la définition proprement dite

Ce qu'il faut bien comprendre, c'est qu'on doit mettre d'abord le mot (<dt>), ensuite sa définition (<dd>). Si vous voulez mettre une seconde définition, il faut recommencer : un dt et ensuite un dd.

<pre><dl> <dt>Geek</dt> <dd>c'est un mec qui croit que dans 1 km, il y a 1024 mètres.</dd> <dt>Nerd</dt> <dd>Le nerd n'aime pas grand chose , à part ce qu'il fait</dd> <dt>No-life</dt> <dd>Acronyme de Névrosé, Obsédé, Lol, Imbécile, Fainéant, Emmanché </dd> </dl></pre>	Geek c'est un mec qui croit que dans 1 km, il y a 1024 mètres. Nerd Le nerd n'aime pas grand chose , à part ce qu'il fait No-life Acronyme de Névrosé, Obsédé, Lol, Imbécile, Fainéant, Emmanché
---	--

Coté CSS

Représentation de la puce :list-style-type

La propriété list-style-type peut prendre de nombreuses valeurs. Certaines d'entre elles concernent uniquement les listes non ordonnées, d'autre concernent uniquement les listes ordonnées :

- Pour les listes non ordonnées () :
 - disc : un disque noir (par défaut).
 - circle : un cercle.
 - square : un carré.
 - none : aucune puce ne sera utilisée.
- Pour les listes ordonnées () :
 - decimal : des nombres 1, 2, 3, 4, 5... (par défaut)
 - decimal-leading-zero : des nombres commençant par zéro (01, 02, 03, 04, 05...). **Ne fonctionne pas sur Internet Explorer**
 - upper-roman : numérotation romaine, en majuscules (I, II, III, IV, V...)
 - lower-roman : numérotation romaine, en minuscules (i, ii, iii, iv, v...)
 - upper-alpha : numérotation alphabétique, en majuscules (A, B, C, D, E...)
 - lower-alpha : numérotation alphabétique, en minuscules (a, b, c, d, e...)
 - lower-greek : numérotation grecque. **Ne fonctionne pas sur Internet Explorer**

Retrait des listes :list-style-position

- inside : la liste n'est pas mise en retrait.
- outside : la liste est mise en retrait (par défaut).

- Liste à puces *outside*
- Liste à puces *inside*



Le texte n'est pas mis en retrait

Changer l'image de la puce :list-style-image



S'obtient par :

```
ul /* Ma liste aura des puces en forme de dossiers */
{
```

```
list-style-image: url("dossier.png");
}
```

Mise en forme des blocs

La taille

2 propriétés CSS :

- **width**: c'est la largeur du bloc. A exprimer en pixels (px) ou en pourcentage (%).
- **height**: c'est la hauteur du bloc. Idem.

Par défaut, et c'est très important à savoir, un bloc prend 100% de la largeur disponible.

Interlude

Le « lorem ipsum »

Généralement, on utilise un texte en faux latin (le texte ne veut rien dire, il a été modifié), le Lorem ipsum, qui permet donc de faire office de texte d'attente. L'avantage de le mettre en latin est que l'opérateur sait au premier coup d'œil que la page contenant ces lignes n'est pas valide, et surtout l'attention du client n'est pas dérangée par le contenu, il demeure concentré seulement sur l'aspect graphique.

Ce texte aurait originellement été tiré d'un Cicéron, (De Finibus Bonorum et Malorum) dont l'une des premières phrases est : « Neque porro quisquam est qui **dolorem ipsum** quia **dolor sit amet, consectetur, adipisci velit...** »⁶

Le texte à copier/coller :

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit.

Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis. Curabitur aliquet pellentesque diam. Integer quis metus vitae elit lobortis egestas. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi vel erat non mauris convallis vehicula. Nulla et sapien. Integer tortor tellus, aliquam faucibus, convallis id, congue eu, quam. Mauris ullamcorper felis vitae erat. Proin feugiat, augue non elementum posuere, metus purus iaculis lectus, et tristique ligula justo vitae magna.

Aliquam convallis sollicitudin purus. Praesent aliquam, enim at fermentum mollis, ligula massa adipiscing nisl, ac euismod nibh nisl eu lectus. Fusce vulputate sem at sapien. Vivamus leo. Aliquam euismod libero eu enim. Nulla nec felis sed leo placerat imperdiet. Aenean suscipit nulla in justo. Suspendisse cursus rutrum augue. Nulla tincidunt tincidunt mi. Curabitur iaculis, lorem vel rhoncus faucibus,

⁶ Il n'existe personne qui aime la souffrance pour elle-même, ni qui la recherche ni qui la veuille pour ce qu'elle est...

felis magna fermentum augue, et ultricies lacus lorem varius purus. Curabitur eu amet.

Exercice : Lorem ipsum - 1

Faites une page xhtml qui affiche le lorem ipsum sur seulement 50 % de la largeur de la fenêtre.

Minimum et maximum

Il vous sera aussi très pratique de demander à ce qu'un bloc ait une taille minimale ou maximale. C'est là qu'entrent en jeu les 4 propriétés CSS suivantes :

- **min-width**: largeur minimale
- **min-height**: hauteur minimale
- **max-width**: largeur maximale
- **max-height**: hauteur maximale

Limiter une taille de bloc overflow:

Supposons que vous vouliez limiter un paragraphe à 250px de large et 100px de haut.

Code : CSS

```
p{
  width: 250px;
  height: 100px;
  text-align: justify;
}
```

Or, cette manœuvre seule ne limite le texte qu'en largeur, pas en hauteur.

Si vous voulez "couper" votre paragraphe pour qu'il soit d'une dimension exacte de 250x100, vous allez devoir utiliser la propriété CSS **overflow**:

Les valeurs que peut prendre overflow sont les suivantes :

- **visible** (par défaut) : si le texte dépasse les limites de taille, le bloc est agrandi de manière à ce que tout soit visible.
- **hidden** : si le texte dépasse les limites, il sera tout simplement coupé. On ne pourra pas voir tout le texte.
- **scroll** : Le navigateur met en place des barres de défilement pour qu'on puisse voir tout le texte.
- **auto** : En gros, le navigateur décide ou pas de mettre des barres de défilement (si nécessaire).

Note : Il existe une balise HTML, **<iframe>**, qui donne à peu près le même résultat. Toutefois, cette balise n'existe plus en XHTML. Il vous faudra donc utiliser un overflow à la place.

Exercice : Lorem ipsum - 2

Faites une page xhtml qui affiche les trois premier paragraphe du lorem ipsum comme suit : (250px * 100px)

Les bordures border:

Border: est la propriété qui gère la bordure des blocs. C'est une méga-propriété, c'est à dire que vous pouvez lui mettre plusieurs valeurs (ici 3). Vous n'êtes pas obligés de mettre toutes les valeurs, et l'ordre dans lequel vous mettez ces valeurs n'a aucune importance.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam lacinia leo ac nunc. Nulla aliquet. Nullam quam leo, volutpat vitae, venenatis eget, blandit eu, augue. Proin ac libero nec magna laoreet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam lacinia leo ac nunc. Nulla aliquet. Nullam quam leo, volutpat vitae,

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam lacinia leo ac nunc. Nulla aliquet. Nullam quam leo, volutpat vitae, venenatis eget, blandit eu, augue. Proin

- La largeur : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px), ou utilisez un des mots-clé suivants :
 - **thin** : bordure fine
 - **medium** : bordure moyenne
 - **thick** : bordure épaisse
- La couleur : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur ("black", "red"...), soit une valeur hexadécimale (#FF0000)
- Le type de bordure : Voici les différentes valeurs disponibles :
 - none : pas de bordure (par défaut)
 - solid : un trait simple.
 - dotted : pointillés.
 - dashed : tirets.
 - double : bordure double.
 - groove : en relief.
 - ridge : effet 3D.
 - inset : autre effet 3D (on a l'impression que le bloc forme un creux).
 - outset : encore un autre effet 3D (on a l'impression que le bloc est surélevé).

Exemples :

```
p
{
    width: 300px; /* Tous les paragraphes font 300px */
    text-align: justify; /* Tous les paragraphes sont justifiés */
}

.rien
{
    border: none;
}

.solid
{
    border: 2px solid black;
}

...
```

Note : On n'est même pas obligé de mettre la même bordure des 4 cotés :

- **border-top**: bordure en haut.
- **border-bottom**: bordure en bas.
- **border-left**: bordure à gauche.
- **border-right**: bordure à droite.

```
.portnawak
{
    border-top: 3px dotted blue;
    border-left: 2px solid green;
    border-right: thin groove green;
    border-bottom: medium ridge pink;
}
```

Balises inline

Ca marche aussi sur des balises inline !

Par exemple, pour enlever ce moche cadre autour des images cliquables :

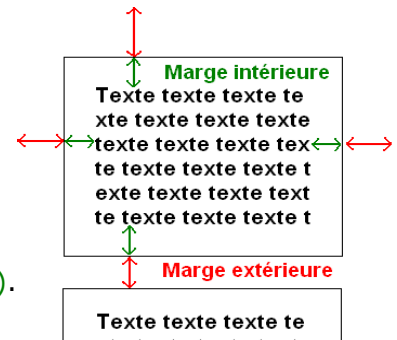
```
a img /* Toutes les images contenues dans un lien */
{
  border: none; /* Pas de bordure */
}
```

Les marges

Il existe 2 types de marges :

- Les marges intérieures
- Les marges extérieures

L'espace entre le texte et la bordure est la marge intérieure (en vert).
L'espace entre la bordure et le prochain bloc est la marge extérieure (en rouge).



En CSS, on peut modifier la taille des marges avec les 2 propriétés suivantes :

- **padding**: indique la taille de la marge intérieure. A exprimer en pixels (px).
- **margin**: indique la taille de la marge extérieure. En pixels aussi.

Note : A nouveau, on peut affiner et ne préciser que le coté concerné avec, pour margin :

- margin-top : marge extérieure en haut.
- margin-bottom : marge extérieure en bas.
- margin-left : marge extérieure à gauche.
- margin-right : marge extérieure à droite

Et padding :

- padding-top : marge intérieure en haut.
- padding-bottom : marge intérieure en bas.
- padding-left : marge intérieure à gauche.
- padding-right : marge intérieure à droite.

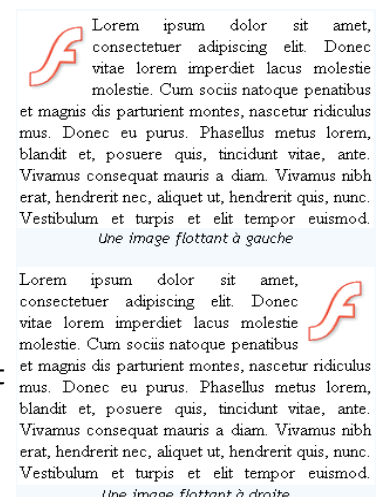
Centrer des blocs margin:auto

Il est tout à fait possible de centrer des blocs, c'est même très pratique pour réaliser un design centré quand on ne connaît pas la résolution du visiteur.

Attention : Cela ne fonctionne que si vous avez indiqué une largeur précise (width) au bloc.

Code : CSS

```
p
{
  width: 350px; /* Comme on a indiqué une largeur ... */
  border: 1px solid black;
  text-align: justify;
  padding: 12px;
  margin: auto; /* ... On peut demander à ce que le bloc
soit centré */
  margin-bottom: 20px;
}
```



Les flottants float:

Une propriété permet d'accrocher une image à un bloc de texte. C'est **float**:

Elle admet deux valeurs :

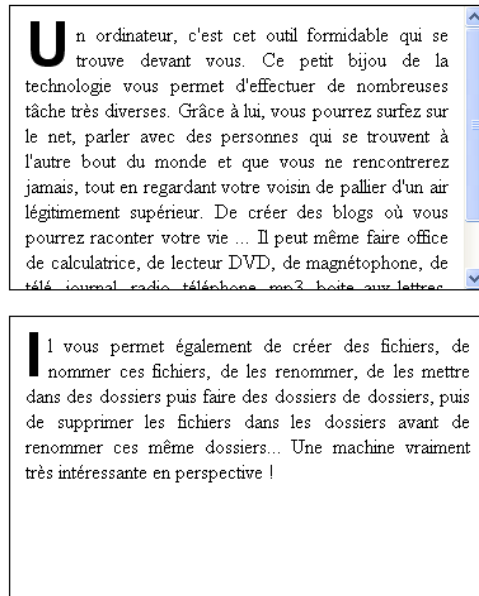
- **left** : l'élément flottera à gauche.
- **right** : l'élément flottera... à droite

Le xhtml qui vas avec :

```
<p></p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus.</p>
```

Exercice : Créer une lettrine

Ecrivez le CSS pour afficher cela :



A partir du code XHTML suivant, que vous n'avez pas le droit de changer :

```
<p>Un ordinateur, c'est cet outil formidable qui se trouve devant vous. Ce petit
bijou de la technologie vous permet d'effectuer de nombreuses tâche très diverses.
Grâce à lui, vous pourrez surfer sur le net, parler avec des personnes qui se
trouvent à l'autre bout du monde et que vous ne rencontrerez jamais, tout en
regardant votre voisin de pallier d'un air légitimement supérieur. De créer des
blogs où vous pourrez raconter votre vie ... Il peut même faire office de
calculatrice, de lecteur DVD, de magnétophone, de télé, journal, radio, téléphone,
mp3, boîte aux lettres, calendrier, horloge, stylo, grille-pain, micro-onde, canapé
et bien plus encore !</p>
```

```
<p>Il vous permet également de créer des fichiers, de nommer ces fichiers, de les
renommer, de les mettre dans des dossiers puis faire des dossiers de dossiers, puis
de supprimer les fichiers dans les dossiers avant de renommer ces même dossiers...
Une machine vraiment très intéressante en perspective ! </p>
```

Positionnement absolu, fixe et relatif

il existe 3 façons de positionner un bloc en CSS :

- Le positionnement absolu : il permet de placer un bloc n'importe où sur la page (en haut à gauche, en bas à droite, tout au centre etc...)
- Le positionnement fixe : c'est pareil que le positionnement absolu, mais cette fois le bloc reste toujours visible, même si on descend plus bas dans la page. Un peu comme le background-attachment:fixed;
- Le positionnement relatif : Il permet de déplacer un bloc par rapport à sa position normale.

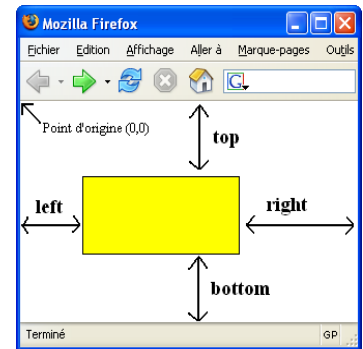
Positionnement absolu

Css :

```
position: absolute;
```

Il faut préciser où on veut que le bloc soit positionné sur la page grâce aux 4 propriétés suivantes :

- **left**: position par rapport à la gauche de la page.
- **right**: position par rapport à la droite de la page.
- **top**: position par rapport au haut de la page.
- **bottom**: position par rapport au bas de la page.



On peut leur donner une valeur en pixels, comme "14px", ou bien une valeur en pourcentage, comme "50%".

Ex :

Coté xhtml :

```
<p>
```

Ceci est un paragraphe normal.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer vel libero. Cras dolor. Quisque quis odio eget justo pulvinar aliquet. Morbi luctus mi. Fusce leo. Integer eleifend condimentum felis. Phasellus vitae nibh. Mauris pellentesque porta magna. Quisque at turpis. Praesent semper odio eget risus. Praesent bibendum imperdiet massa. Quisque ac arcu ac augue dapibus vestibulum. Pellentesque gravida. Mauris turpis. Aenean molestie. Praesent at quam et ligula pellentesque luctus.

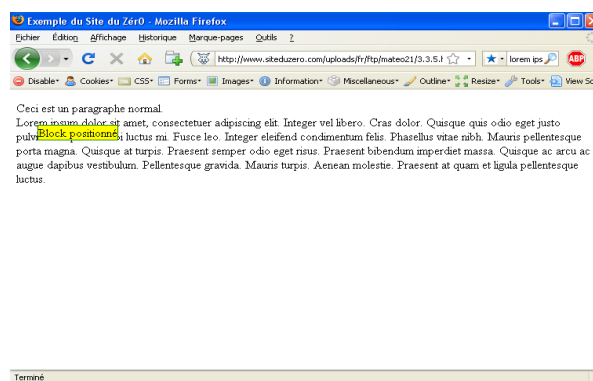
```
<div>Block positionné</div>
```

Coté CSS :

```
div
```

```
{  
  background-color: yellow;  
  border: 1px solid green;  
  
  position: absolute;  
  right: 50%;  
  bottom: 20px;  
}
```

Résultat :



Attention : Si vous positionnez un bloc A en absolu, et qu'à l'intérieur de ce bloc vous positionnez un autre bloc B en absolu, ce positionnement-là ne se fera plus par rapport au coin en haut à gauche de la fenêtre mais par rapport au coin en haut à gauche du bloc A.

Le positionnement fixe

Le fonctionnement est **exactement le même** que pour le positionnement absolu, sauf que cette fois le bloc reste fixe, même si on descend plus bas dans la page.

Le positionnement fixe fonctionne sur tous les navigateurs ... sauf sur Internet Explorer 6

Le positionnement relatif

Plus délicat, le positionnement relatif sert généralement à faire des "ajustements".

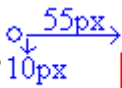
Pas de doute,  si on veut comprendre corr

Si vous faites un `position: relative` et que vous appliquez une des propriétés `top`, `bottom`, `left`, `right`, le texte sur fond rouge va se déplacer par rapport à sa position d'origine (le point 0,0 précisé sur le schéma)

```
strong
{
  background-color: red;
  color: yellow;

  position: relative;
  left: 55px;
  top: 10px;
}
```

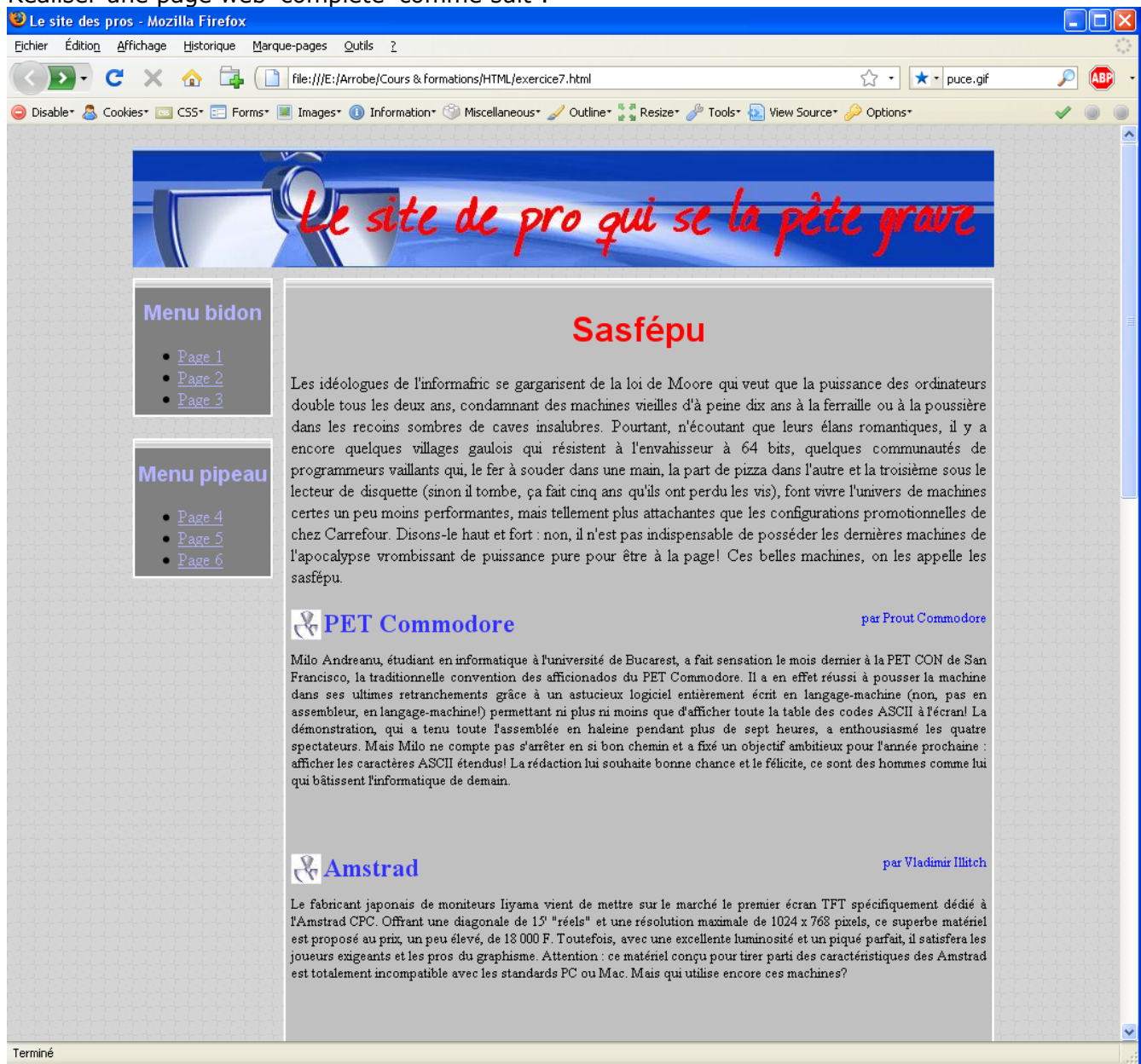
Le texte s'est décalé par rapport à sa position initiale, comme ceci :

Pas de doute,  si on veut com

Travaux dirigés

But

Réaliser une page web 'complète' comme suit :



Cette page sera d'un type classique. Soit 800 pixels de large et centré à l'affichage.

La structure générale sera la suivante :

En haut, l'en-tête (aussi appelé "header"). Cet en-tête contient en général le titre de votre site, sous forme de bannière (image).

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.

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

Matériel

Récupérez les images suivantes sur www.arrobe.fr/xhtmll :

- banniere.jpg
- banniere_2.png
- puce.jpg
- bordure.jpg
- fond.png

Organisation

La page s'appellera exercice7.html

Les images sont dans un sous répertoire image

La feuille de style dans un sous répertoire style, et s'appellera techno.css

Pour chacune des 4 zones (header, menu, corps & footer) on prévoit un div spécifique.

étape 1

```
<!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>Le site des pros</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" media="screen" type="text/css"
        title="Style High Tech" href="style/techno.css" />
  </head>
  <body>
    <div class="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>
  </body>
</html>
```

étape 2 : L'en-tête

Alors ça, ça va être rapide.

En général un en-tête c'est juste une bannière (une image) voire, pour les nuls en dessin, un titre h1.

Ici, on va utiliser l'image banniere.jpg :



Pour mettre cette image dans le <div>; On a 2 possibilités :

- Soit on crée une balise **** à l'intérieur, afin d'insérer notre bannière.
- Soit on ne met rien à l'intérieur du **<div class="en_tete">** et on mettra la bannière sous forme d'image de fond tout à l'heure dans le CSS.

Nous allons choisir la seconde solution, car elle permettra de changer le design du site plus facilement.

Donc, on ne met donc rien entre le **<div>** et le **</div>** (juste un espace à la rigueur)

étape 3 : Menu et sous-menus

Dans le bloc du menu (en bleu marine), on va avoir à créer plusieurs sous-blocs pour séparer les différents éléments du menu.

il contient 2 sous-blocs (cadres verts) qui vont servir de sous-menus.

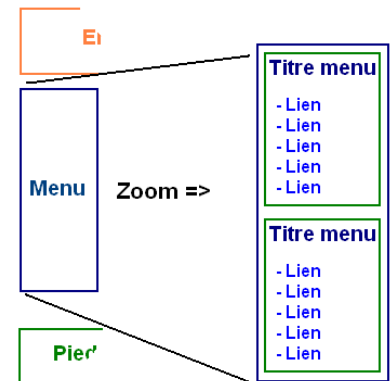
```
<div class="menu"> <!-- Cadre en bleu marine sur le schéma -->
  <div class="element_menu"> <!-- Cadre en vert sur le schema -->
    Texte du premier menu
  </div>
  <div class="element_menu">
    Texte du second menu
  </div>
</div>
```

Ajoutons maintenant le texte à l'intérieur du menu.

On va mettre dans chaque "element_menu" un titre de menu (<h3>).

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

Et chaque élément du menu sera un lien vers une des pages du site.



```
<div class="menu">
  <div class="element_menu">
    <h3>Menu bidon</h3>
    <ul>
      <li><a href="page1.html">Page 1</a></li>
      <li><a href="page2.html">Page 2</a></li>
      <li><a href="page3.html">Page 3</a></li>
    </ul>
  </div>
  <div class="element_menu">
    <h3>Menu pipeau</h3>
    <ul>
      <li><a href="page4.html">Page 4</a></li>
      <li><a href="page5.html">Page 5</a></li>
      <li><a href="page6.html">Page 6</a></li>
    </ul>
  </div>
</div>
```

étape 4 : Le corps

La ... c'est à vous de jouer.

Pour mon exemple, j'ai mis le texte suivant :

```
<h1>Sasfêpu</h1>
```

<p class="chapeau">Les idéologues de l'informafric se gargarisent de la loi de Moore qui veut que la puissance des ordinateurs double tous les deux ans, condamnant des machines vieilles d'à peine dix ans à la ferraille ou à la poussière dans les recoins sombres de caves insalubres. Pourtant, n'écoutant que leurs élans romantiques, il y a encore quelques villages gaulois qui résistent à l'envahisseur à 64 bits, quelques communautés de programmeurs vaillants qui, le fer à souder dans une main, la part de pizza dans l'autre et la troisième sous le lecteur de disquette (sinon il tombe, ça fait cinq ans qu'ils ont perdu les vis), font vivre l'univers de machines certes un peu moins performantes, mais tellement plus attachantes que les configurations promotionnelles de chez Carrefour. Disons-le haut et fort : non, il n'est pas indispensable de posséder les dernières machines de l'apocalypse vrombissant de puissance pure pour être à la page! Ces belles machines, on les appelle les sasfépu.

</p>

<h2>PET Commodore</h2>

<div class="auteur">par Prout Commodore</div>

<p class="texte">Milo Andreanu, étudiant en informatique à l'université de Bucarest, a fait sensation le mois dernier à la PET CON de San Francisco, la traditionnelle convention des aficionados du PET Commodore. Il a en effet réussi à pousser la machine dans ses ultimes retranchements grâce à un astucieux logiciel entièrement écrit en langage-machine (non, pas en assembleur, en langage-machine!) permettant ni plus ni moins que d'afficher toute la table des codes ASCII à l'écran! La démonstration, qui a tenu toute l'assemblée en haleine pendant plus de sept heures, a enthousiasmé les quatre spectateurs. Mais Milo ne compte pas s'arrêter en si bon chemin et a fixé un objectif ambitieux pour l'année prochaine : afficher les caractères ASCII étendus! La rédaction lui souhaite bonne chance et le félicite, ce sont des hommes comme lui qui bâtissent l'informatique de demain.

</p>

<h2>Amstrad</h2>

<div class="auteur">par Vladimir Illitch</div>

<p class="texte">Le fabricant japonais de moniteurs Iiyama vient de mettre sur le marché le premier écran TFT spécifiquement dédié à l'Amstrad CPC. Offrant une diagonale de 15" "réels" et une résolution maximale de 1024 x 768 pixels, ce superbe matériel est proposé au prix, un peu élevé, de 18 000 F. Toutefois, avec une excellente luminosité et un piqué parfait, il satisfera les joueurs exigeants et les pros du graphisme. Attention : ce matériel conçu pour tirer parti des caractéristiques des Amstrad est totalement incompatible avec les standards PC ou Mac. Mais qui utilise encore ces machines?

</p>

<h2>Amiga</h2>

<div class="auteur">par Fred</div>

<p class="texte">La rolls des sasfépuguères n'a décidément pas dit son dernier mot. Rappel des faits : Commodore, qui avait investi 5 milliards de dollars dans le développement de l'Amiga, avait fait faillite et 92, vendant la marque et les brevets Amiga à Digital Equipment, subdivision de Sumitomo Ltd, pour 2 milliards de dollars. Toutefois, frappée par la crise, DE n'eut bientôt plus les moyens d'assumer les frais de développement et surtout le financement d'une nouvelle usine pour le nouvel Amiga 32 bits, et avait dû revendre le tout à la société Escom, créée pour l'occasion par des anciens de commodore, pour la modique somme de 600 millions de dollars. Un ambitieux plan marketing était prêt pour remettre l'ordinateur familial sur les rails quand patatras! Alan Zbojnowsky, PDG d'Escom, s'enfuit aux bahamas avec la caisse, la paye des employés et 4500 tickets restau. Après la faillite d'Escom, c'est Bull qui rachète l'Amiga pour 100 millions de dollars, pour le revendre six mois plus tard à Sony (montant de la transaction secret, mais probablement voisin de 12 millions de dollars). Cependant, la firme japonaise, occupée à lancer la PlayStation, n'avait que faire d'une machine concurrente, et la revend discrètement à un consortium Israélo-Libanais, Zahwara Investments, en paiement d'une créance de \$ 830 000. On retrouve ensuite trace de l'Amiga en Iran, puis dans les républiques musulmanes de l'ex-URSS. Et puis plus rien depuis deux ans.
On croyait l'Amiga définitivement enterré quand, le 12 juillet dernier, la dépêche tombe sur les téléscripteurs : le combinat Kolossal Molossol Uralski, propriétaire entre autre des mines d'uranium de Tcheliabinsk, venait de revendre les droits de notre ordinateur préféré à Petrograd Silicon, une start-up menée tambour battant par Ivan Romanenko, jeune entrepreneur de Saint-Petersbourg, frais émoulu de l'université et neveu d'un ancien directeur du KGB du temps de Gorbatchev. Ce jeune loup vient de faire l'affaire du siècle en rachetant les licences pour la modique somme de 137 roubles 20 kopecks, soient 0,17 centimes. Affaire à suivre.

</p>

<h2>Amstrad encore</h2>

<div class="auteur">par Vladimir Illitch</div>

<p class="texte">Drame au siège de la société Iiyama à Kobé (Japon) : Nous apprenons que monsieur Noriyuki Sato, PDG de la société, a été interpellé alors qu'il tentait de prendre la fuite après avoir roué de coups puis défenestré du treizième étage messieurs Shintaro Tanaka et Jingoro Matsuda, respectivement directeur technique et directeur commercial de son entreprise. De source proche de la police, il semblerait que le démarrage plutôt lent du modèle CPCTFT-01, l'écran TFT pour Amstrad CPC dans lequel la société avait mis tous ses espoirs, ait été à l'origine du différends. Affaire à suivre.

</p>

<h2>Thomson</h2>

<div class="auteur">par Stan Huibits</div>

<p class="texte">Thomson va-t-elle reprendre la micro après quinze ans de silence? On pourrait le croire en considérant la curieuse réaction du service relations publiques de Thomson Multimédia, à qui nous avons téléphoné récemment pour leur demander des nouvelles de la micro "made in France".
"Quoi? Thomson a fait de la micro?" nous a répondu sans rire la demoiselle au demeurant fort gentille. "C'est bien la première fois que j'en entends parler", continue l'employée, visiblement briefée par sa direction. Et de faire obstinément l'ingénue pour dissiper les soupçons... Un tel silence, une telle manipulation, ne peuvent avoir qu'une seule explication : Thomson va reprendre la

micro! Courage, possesseurs de MO et TO, votre patience va bientôt être récompensée, et nous pourrons bientôt nous remettre fièrement à nos écrans CGA 320x200 en 4 couleurs pour épater les PCistes.

```
</p>
<h2>Abaque</h2>
<div class="auteur">par Nielle Brasfort</div>
<p class="texte">"L'abaque a un avenir brillant devant lui" se réjouit le mathématicien Lucien Chapendu, président d'honneur du Abacus Research Group. "Jamais les perspectives de développement de notre système informatique n'ont été aussi riches" continue ce passionné, chiffres à l'appui. Il est vrai que la puissance de calcul des micro-ordinateurs modernes est largement sous-exploitée, notamment dans les applications bureautiques, et qu'en dehors des interfaces graphiques ("des fioritures pour attirer les gogos", d'après le professeur Chapendu), rien ne justifie une telle débauche de MIPS et de méga-octets. A ce titre, l'abaque, de technologie robuste, éprouvée par les siècles et peu chère, constitue une solution avantageuse, notamment pour les PME des pays émergents, mais aussi pour de grandes entreprises occidentales (on parle notamment du français Bull) qui se sont d'ores et déjà montrées intéressées. Dans cette optique, l'Abacus Research Group oriente ses travaux à la mise au point d'une architecture réseau à base de gobelets en PVC reliés par un backbone ficelle. "Nous avons encore quelques problèmes de débit", confie le professeur.<br />
  Toutefois, système "A" pourrait à terme se voir concurrencé par le développement d'une nouvelle technologie, le "Finger Computing", qui intègre dans un même appareil l'unité de traitement arithmétique et l'interface d'impression. L'avenir nous dira qui aura le dernier mot.
</p>
</div>
```

étape 5 : Pied de page

On fait court :

```
<div class="pied_de_page">
  <p>Copyleft "Port'nawak team" 2008, faites en ce que vous voudrez</p>
</div>
```

étape 6 : Centrer le design

On passe maintenant au CSS.

On veut un cite centré de 800 pixels de large.

C'est la balise body qui vas servir à cela.

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

Enfin, je met une image de fond (background-image) à la page pour faire un peu moins vide : utilisez l'image fond.gif.

```
body
{
  width: 800px;
  margin: auto; /* Pour centrer la page */
  margin-top: 20px; /* Pour éviter de coller avec le haut de la fenêtre du
navigateur. */
  margin-bottom: 20px; /* Idem pour le bas du navigateur */
  background-image: url("../image/fond.gif"); /* Notez le chemin : "../" -> je
remonte d'un répertoire; "image/" -> je descend dans image; ... */
}
```

étape 7 : L'image d'en tête

Les valeurs qui suivent doivent être mises pour le div « en_tete » soit

```
.en_tete
{
  ...
}
```

La première chose à faire, si on veut que la bannière soit visible, ce sera d'agrandir la taille du bloc "en_tete" afin qu'il puisse contenir la bannière entière.

Comme notre bannière fait 800x112 pixels, on va définir une largeur et une hauteur de ces dimensions :

```
width: 800px;
```

height: 112px;

Ensuite, on va indiquer quelle image de fond on veut mettre à notre grand bloc qui est pour l'instant vide :

background-image: url("../image/banniere.jpg");

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;

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;

étape 8 : Le menu

Le menu doit être à gauche du corps de la page, alors que pour l'instant le menu se trouve au-dessus.

Pour cela, le plus rapide, c'est d'utiliser la propriété CSS **float**.

On va aussi donner une taille de 130 pixels au menu, pour qu'il ne soit pas trop grand.

Pour la couleur du fond : un gris clair (#808080)

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 white.

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.

étape 9 : Les éléments du menu

Une série de styles vont permettre de mettre en relief les menus :

```
.element_menu h3 /* Tous les titres h3 des menus */
{
    color: #B3B3FF; /* bleu gris */
    font-family: Arial, "Arial Black", serif;
    text-align: center;
}

.element_menu ul /* Toutes les listes à puces se trouvant dans un menu */
{
    padding: 0px; /* Tous les côtés ont une marge intérieure de 0 pixels */
    padding-left: 40px; /* ... mais on modifie ensuite la marge de gauche, donc
    celle-là fera finalement 20 pixels */
    margin: 0px; /* Idem pour margin, ça nous évite d'avoir à en écrire 4 (margin-
    left, margin-right...) */
    margin-bottom: 5px; /* Même chose que tout à l'heure, on modifie ensuite juste
    margin-bottom, mais tous les autres sont à 0px */
}

.element_menu a /* Tous les liens se trouvant dans un menu */
{
    color: #B3B3FF;
}

.element_menu a:hover /* Quand on pointe sur un lien du menu */
```

```
{
background-color: #B3B3FF;
color: black;
}
```

étape 10 : Le positionnement du 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. C'est un comportement normal car on a fait un **float:left** sur le menu.

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

En couleur de fond, un gris moyen : #c0c0c0

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.

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...

```
background-image: url("../image/bordure.jpg");
```

```
background-repeat: repeat-x; /* Une petite image de fond qui se répètera
horizontalement en haut */
```

```
border: 2px solid white; /* Une bordure pour bien marquer les limites du corps et
pour faire joli */
```

étape 11 : La mise en page du corps

Le titre h1 :

Arial, Rouge et Centré

Les titre h2 :

4 choses

- 1 – écrire en bleu gris (#3030ff) aligné à gauche
- 2 – Ajouter une image de fond (puce.jpg) mais non répétée.
- 3 – Décaler le texte de 30 pixels pour éviter le chevauchement avec l'image
- 4 – Forcer la hauteur du titre à 30 pixels pour que l'image tienne en entier.

Le premier paragraphe (« Les idéologues de l'informafric .. ») à une classe qui lui est propre : chapeau. On le met en police moyenne, noire et justifiée.

Les autres paragraphes (class="texte") seront en noir, justifié et petite police.

Les auteurs (class="auteur") sont en bleu, aligné à droite et petite police.

Par contre, pour « tasser » un peu le tout, on va utiliser un positionnement relatif **négatif**.

Les auteurs seront remonté de 50 pixels par :

```
position: relative;
top: -50px;
```

Il faut faire la même manipulation sur les paragraphes (class="texte") pour ne pas perdre l'espace ainsi gagné. (seulement -40px)

étape 12 : Le pied de page

Rien de spécial :

```
.pied_de_page
{
    padding: 5px;

    text-align: center;

    color: #B3B3B3;
    background-color: #626262;
    background-image: url("../image/bordure.jpg");
    background-repeat: repeat-x;

    border: 2px solid white;
}
```

Et voilà .. c'est fini.

N'oubliez pas de valider le xhtml : <http://validator.w3.org/>

Et le CSS : <http://jigsaw.w3.org/css-validator/>

étape 13 : les feuilles de styles alternatives

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>` :

```
<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**"). Si le visiteur dispose d'un navigateur suffisamment récent (c'est-à-dire grosso modo "s'il n'utilise pas Internet Explorer"), il pourra choisir le design qui lui plaît.

Il suffit d'aller dans le menu "Affichage > Style de la page" et de choisir le design que vous voulez.

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.

```
<head>
    <title>Presque CSSZenGarden ... Presque</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_mecanique.css" />
    <!-- Ci-dessous les designs alternatifs que vous proposez -->
    <link rel="alternate stylesheet" media="screen" type="text/css"
        title="Rose pour les filles" href="design_poney_rose.css" />
    <link rel="alternate stylesheet" media="screen" type="text/css"
        title="Sombre et terrifiant" href="sombre.css" />
    <link rel="alternate stylesheet" media="screen" type="text/css"
        title="Aquatique" href="aquatique.css" />
</head>
```

Réalisez au moins un CSS alternatif. (vous pouvez, par exemple, vous inspirer de banniere_2.png)

Les tableaux

Retour au XHTML.

Un tableau simple

<table> **</table>** : Balise qui indique le début et la fin d'un tableau.

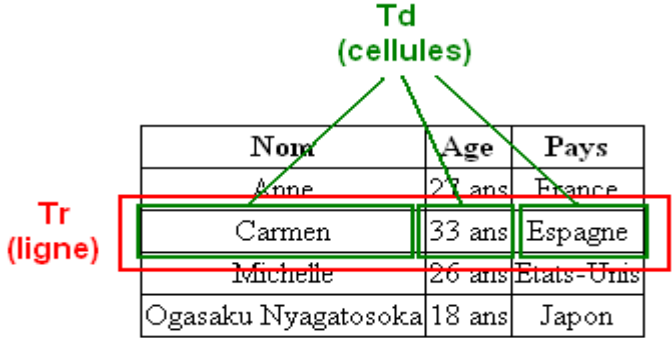
<tr> **</tr>** : (Table Row) indique le début et la fin d'une ligne du tableau.

<td> **</td>** : (Table Data) indique le début et la fin du contenu d'une cellule.

En xhtml, le tableau se construit ligne par ligne et case par case.

On ne gère pas de colonnes.

Exemple .:

<pre><table> <tr> <td>Anne</td> <td>27 ans</td> <td>France</td> </tr> <tr> <td>Carmen</td> <td>33 ans</td> <td>Espagne</td> </tr> <tr> <td>Michelle</td> <td>26 ans</td> <td>Etats-Unis</td> </tr> <tr> ... </tr> </table></pre>	
--	--

Si vous tapez l'exemple ci dessus, le texte s'affichera à la suite, sans les cases du tableau.

Il faut préciser, en CSS :

```
td /* Toutes les cellules des tableaux... */
{
  border: 1px solid black; /* ... auront une bordure de 1px */
}
```

En fait c'est pas encore tout à fait ça : A ce stade, le tableau ressemble à ça :

Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

Il existe une propriété CSS spécifique aux tableaux : **border-collapse**, qui signifie "coller les bordures entre elles".

Cette propriété peut prendre 2 valeurs :

- **collapse** : les bordures seront collées entre elles, c'est l'effet qu'on recherche.
- **separate** : les bordures seront dissociées (valeur par défaut)

Le CSS est donc :

```
table
{
  border-collapse: collapse; /* Les bordures du tableau seront collées */
}
```

```
td
{
border: 1px solid black;
}
```

La ligne d'en-tête

Maintenant, on va rajouter la ligne d'en-tête du tableau. Dans notre exemple, les en-têtes sont "Nom", "Age" et "Pays".

La ligne d'en-tête se crée avec un **<tr>** comme on a fait jusqu'ici, mais les cellules à l'intérieur sont cette fois des **<th>** (table header) et non pas des **<td>** !

Le xhtml devient :

```
<table>
  <tr>
    <th>Nom</th>
    <th>Age</th>
    <th>Pays</th>
  </tr>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>Etats-Unis</td>
  </tr>
</table>
```

Et le CSS devient :

```
table
{
border-collapse: collapse;
}
td, th /* Mettre une bordure sur les td ET les th */
{
border: 1px solid black;
}
```

Nom	Age	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

Résultat :

Titre du tableau

Ajouter **<caption>**Passagers du vol 377**</caption>** entre **<table>** et le premier **<tr>**

Résultat :

Passagers du vol 377		
Nom	Age	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

Fusionner des cellules

Pour réaliser le tableau suivant :

Lundi	libre	
mardi	8h	10h
	bureau	RDV Mr X

Il y a 2 opérations « spéciales » à faire.

Sur la 1er ligne, il faut fusionner les cases 2 et 3

Pour cela on supprime la case 3, et on précise que la case 2 « s'étend » sur 2 cases de large :

```

<tr>
  <td>Lundi</td>
  <td colspan="2">libre</td>
</tr>

```

Sur la 1er cellule de la 2eme ligne, on précise que celle-ci s'étend sur 2 lignes :

```

<tr>
  <td rowspan="2">mardi</td>
  <td>8h</td>
  <td>10h</td>
</tr>

```

Attention : La troisième ligne comporte maintenant 4 cases (Les 3 d'origine plus la case « mardi » qui déborde sur la 3eme ligne); il faut donc en supprimer une pour avoir un tableau de 3 * 3.

Soit :

```

<table>
  <tr>
    <td>Lundi</td>
    <td colspan="2">libre</td>
  </tr>
  <tr>
    <td rowspan="2">mardi</td>
    <td>8h</td>
    <td>10h</td>
  </tr>
  <tr>
    <td>bureau</td>
    <td>RDV Mr X</td>
  </tr>
</table>

```

Présentation des tableaux

Rien qu'en réutilisant des propriétés déjà connues :

```

caption /* Titre du tableau */
{
  margin: auto; /* Centre le titre du tableau */
  font-family: Arial, Times, serif;
  font-weight: bold;
  font-size: 1.2em;
  color: #009900;
  margin-bottom: 20px; /* Pour éviter que le titre ne soit trop collé au tableau
    en-dessous */
}

table /* Le tableau en lui-même */
{
  margin: auto; /* Centre le tableau */
  border: 4px outset green; /* Bordure du tableau avec
    effet 3D (outset) */
  border-collapse: collapse; /* Colle les cellules entre
    elles */
}

th /* Les cellules d'en-tête */
{
  background-color: #006600;
  color: white;
  font-size: 1.1em;
  font-family: Arial, "Arial Black", Times, serif;
}

td /* Les cellules normales */

```

Passagers du vol 377

Nom	Age	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis
François	43 ans	France
Martine	34 ans	France
Jonathan	13 ans	Australie
Xu	19 ans	Chine
Nom	Age	Pays

```

{
  border: 1px solid black;
  font-family: "Comic Sans MS", "Trebuchet MS", Times, "Times New Roman",
  serif;
  text-align: center; /* Tous les textes des cellules seront centrés*/
  padding: 5px; /* Petite marge intérieure aux cellules pour éviter que le texte
  touche les bordures */
}

```

Démo :

CSS propre aux tableaux

Les principales :

caption-side: indique où doit se trouver le titre du tableau (ne fonctionne pas sur Internet Explorer). Prend les valeurs suivantes :

- **top** : le titre sera placé en haut du tableau (par défaut).
- **bottom** : le titre sera placé en bas du tableau.
- **left** : le titre sera placé à gauche du tableau.
- **right** : le titre sera placé à droite du tableau.

border-collapse: Indique que les bordures des cellules seront collées entre elles.

- **separate** : les bordures seront séparées les unes par rapport aux autres (par défaut).
- **collapse** : les bordures seront collées entre elles.

vertical-align: alignement vertical du contenu d'une cellule. Si une cellule a une hauteur importante, il est possible de placer son contenu en haut, en bas ou au milieu de la cellule

top : le contenu sera aligné en haut de la cellule (par défaut).

middle : le contenu sera aligné au milieu de la cellule

bottom : le contenu sera aligné en bas de la cellule

Les formulaires

Les formulaire c'est ce qui permet à un utilisateur de saisir une information.

Par exemple, ça c'est un formulaire :



La balise form

Qui dit formulaire, dit envoi d'informations.

Qui dit envoi d'informations dit programme pour les traiter.

... Et ça, c'est le prochain cours : PHP.

Pour ce cour, on vas se contenter d'envoyer les formulaires par mail, de façon non formaté.

La balise qui commence (et termine) un formulaire c'est **<form>** et **</form>**

La balise ouvrante reçoit plusieurs paramètres, mais nous les verrons plus tard (cours PHP)

Pour l'instant on utilisera :

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
</form>
```

Zone de texte à une ligne

C'est ça :

Pour une zone de texte à une ligne, on doit taper : **<input type="text" />**

On retrouvera souvent la balise input. Selon la valeur du paramètre type, elle aura des aspects très différents.

Mais ce n'est pas tout ! Il manque un attribut qui sera très important : c'est le nom de la zone de texte.

En effet, cela vous permettra plus tard (dans le langage PHP) de reconnaître que tel texte est le pseudo du visiteur, tel texte est son mot de passe etc...

Il faut donc donner un nom à cette zone de texte, grâce à l'attribut name. Ici, on va supposer qu'on demande au visiteur de rentrer son pseudo :

```
<input type="text" name="pseudo" />
```

Note : Pensez à entourer votre balise **<input />** par une balise block comme **<p></p>** car sinon votre page web ne sera pas valide

Les labels

Cette zone de texte est bien jolie, mais si votre visiteur tombe dessus il ne sait pas trop ce qu'il doit mettre dedans. On va justement lui expliquer qu'il doit rentrer son pseudo.

Pour indiquer ce que signifie une zone de texte au visiteur, on utilise la balise **<label>** qui entoure le libellé, comme ceci :

```
<label>Votre pseudo</label> <input type="text" name="pseudo" />
```

Mais ça ne suffit pas. Il faut lier le label avec la zone de texte.

Pour ce faire, il faut donner un nom à la zone de texte, non pas avec l'attribut name mais avec l'attribut id (que l'on peut utiliser sur toutes les balises).

Truc : Ne voyous embêtez pas; donnez le même nom aux id et aux name.

Soit :

```
<label id="pseudo">Votre pseudo</label>
<input type="text" name="pseudo" id="pseudo" />
```

Le code sera :

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
  <p>
    <label id="pseudo">Votre pseudo</label>
    <input type="text" name="pseudo" id="pseudo" />
  </p>
</form>
```

Quelques attributs supplémentaires

Il est possible de donner une valeur par défaut à une zone de texte.

Pour faire cela, ajouter l'attribut **value** à **<input />**.

Vous pouvez modifier la largeur de votre zone de texte ainsi que le nombre maximal de caractères que l'on peut mettre dedans.

La largeur se définit avec **size**.

Le nombre maximal de caractères se définit avec **maxlength**.

Exemple :

```
<input type="text" name="pseudo" id="pseudo" value="toto" size="30" maxlength="10" />
```

Zone de mot de passe

Pour que la zone de texte se comporte comme une zone "mot de passe", c'est-à-dire une zone où on ne voit pas à l'écran ce qui est tapé dedans : Changer l'attribut type de **<input />**. Mettez **type="password"**, et le tour est joué !

Je complète mon formulaire. Il demande maintenant au visiteur son pseudo ET son mot de passe :

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
  <p>
    <label id="pseudo">Votre pseudo</label> :
    <input type="text" name="pseudo" id="pseudo" />
    <br />
    <label id="pass">Votre mot de passe</label> :
    <input type="password" name="pass" id="pass" />
  </p>
</form>
```

Zone de texte multiligne

La zone de texte multiligne est une balise existant par paire (contrairement à **<input />**). C'est la balise **<textarea></textarea>**

Il faut lui donner un nom avec name, et utiliser un label qui explique de quoi il s'agit.

Pour modifier la taille du textarea : 2 façons.

- En CSS : il suffit d'appliquer les propriétés CSS width et height au textarea. Malheureusement cette méthode ne marche pas sur Internet Explorer.
- Avec des attributs : on peut ajouter les attributs **rows** et **cols** à la balise **<textarea>**. Le premier indique le nombre de lignes du textarea, et le second le nombre de colonnes.

Pour pré-remplir le textarea on ne peut pas utiliser value= (impossible de mettre plusieurs lignes !). il faut utiliser l'espace entre les balises **<textarea>** et **</textarea>**

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
  <p>
    <label id="ameliorer">Comment améliorer ce cours ?</label> :
    <textarea name="ameliorer" id="ameliorer" rows="10" cols="50" />
      Améliorer ce cours ?
      Mais enfin ! Il est impossible d'améliorer la perfection !
    </textarea>
  </p>
</form>
```

</form>

Les cases à cocher

C'est rapide : `<input type="checkbox" />`

Pour obtenir :

- ☐ Frites
- ☐ Steak haché
- ☐ Epinards
- ☐ Huitres

Il faut :

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
  <p>
    Cochez les aliments que vous aimez manger :<br />
    <input type="checkbox" name="frites" id="frites" />
    <label for="frites">Frites</label><br />
    <input type="checkbox" name="steak" id="steak" />
    <label for="steak">Steak haché</label><br />
    <input type="checkbox" name="epinards" id="epinards" />
    <label for="epinards">Epinards</label><br />
    <input type="checkbox" name="huitres" id="huitres" />
    <label for="huitres">Huitres</label>
  </p>
</form>
```

Vous pouvez faire en sorte qu'une case soit déjà cochée par défaut. Pour faire cela, il faut rajouter l'attribut `checked="checked"`

Les boutons radio

A la différence des cases à cocher, il y a toujours au moins et au plus un bouton radio de sélectionné.

La balise à utiliser est toujours un `<input />`, avec cette fois la valeur `"radio"` pour l'attribut `type`.

La grosse différence avec les cases à cocher, c'est que les zones d'options fonctionnent par "groupe". Tout un groupe d'options a le même nom, mais un attribut `value` différent à chaque fois.

```
<p>
  Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :<br />
  <input type="radio" name="age" value="moins15" id="moins15" />
  <label for="moins15">Moins de 15 ans</label><br />
  <input type="radio" name="age" value="1525" id="1525" checked="checked" />
  <label for="1525">15-25 ans</label><br />
  <input type="radio" name="age" value="2540" id="2540" />
  <label for="2540">25-40 ans</label><br />
  <input type="radio" name="age" value="plus40" id="plus40" />
  <label for="plus40">Encore plus vieux que ça ?</label><br />
</p>
```

Donne :

- Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :
- ☐ Moins de 15 ans
 - ☒ 15-25 ans
 - ☐ 25-40 ans
 - ☐ Encore plus vieux que ça ?!

Les listes déroulantes

Pour obtenir ça :

Dans quel pays habitez-vous ?



On utilise la balise `<select>` `</select>`.

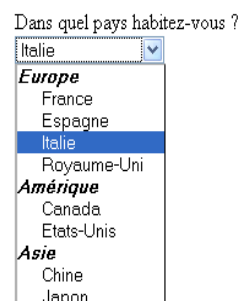
Comme ceci :

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
  <p>
    <label for="pays">Dans quel pays habitez-vous ?</label><br />
    <select name="pays" id="pays">
      <option value="F">France</option>
      <option value="E">Espagne</option>
      <option value="I" selected="selected">Italie</option>
      <option value="GB">Royaume-Uni</option>
      <option value="C">Canada</option>
      <option value="USA">Etats-Unis</option>
      <option value="CN">Chine</option>
      <option value="J">Japon</option>
    </select>
  </p>
</form>
```

Note : On peut créer des groupes d'options à l'intérieur de la liste, grâce à la balise `<optgroup>` `</optgroup>`. Vous devez lui ajouter l'attribut `label` qui permet de donner un nom au groupe (à ne pas confondre avec la balise `<label>` !).

Exemple :

```
<form method="post" action="mailto:hugues.levasseur@arrobe.fr">
  <p>
    <label for="pays">Dans quel pays habitez-vous ?</label><br />
    <select name="pays" id="pays">
      <optgroup label="Europe">
        <option value="F">France</option>
        <option value="E">Espagne</option>
        <option value="I" selected="selected">Italie</option>
        <option value="GB">Royaume-Uni</option>
      </optgroup>
      <optgroup label="Amerique">
        <option value="C">Canada</option>
        <option value="USA">Etats-Unis</option>
      </optgroup>
      <optgroup label="Asie">
        <option value="CN">Chine</option>
        <option value="J">Japon</option>
      </optgroup>
    </select>
  </p>
</form>
```



Note : Si on ajoute l'attribut **size="n"** à la balise **<label>** on obtient, non pas une combo box, mais une liste déroulante de n lignes de haut.

Note : Si on ajoute l'attribut **multiple="multiple"** à la balise **<label>** on peut, en maintenant la touche contrôle enfoncée, sélectionner plusieurs valeurs.

Définir un ordre de tabulation

C'est un des points censé faciliter la navigation.

On peut se déplacer dans un formulaire uniquement grâce à la touche "Tab" (tabulation).

A chaque fois qu'on appuie sur Tab, on va au champ suivant.

A chaque fois qu'on fait Maj + Tab, on retourne au champ précédent.

On va utiliser l'attribut **tabindex** qui peut se rajouter sur toutes les balises du formulaire.

On doit lui mettre un nombre pour valeur. Chaque champ du formulaire doit avoir un nombre différent.

Note : Vous n'êtes pas obligés de mettre des nombres qui se suivent. Il est même conseillé de laisser des "espaces" entre les nombres.

Organiser le formulaire en zones

Si vous avez un formulaire assez gros (et en général ça sera le cas), il est facile que le visiteur se perde dans la masse d'informations qu'il a à entrer. Il est possible en XHTML de grouper plusieurs champs ayant un thème entre eux.

On utilisera la balise **<fieldset></fieldset>** pour délimiter un groupe de champs.

A l'intérieur de cette balise, vous mettrez vos champs (vos **<input />** entre autres...) ainsi qu'une autre balise : **<legend></legend>**. Celle-ci permet de donner le nom du groupe.

Vos coordonnées

Quel est votre nom ?

Quel est votre prénom ?

Quel est votre e-mail ?

Votre souhait

Faites un souhait que vous voudriez voir exaucé :

☐ Etre riche
☐ Etre célèbre
☐ Etre **encore** plus intelligent
☐ Autre...

Si "Autre", veuillez préciser :

Ex :

Les CSS

Pas de nouvelles propriétés, vous connaissez déjà tout.

La feuille de style suivante transformera l'exemple du dessus en ça :

```
input, textarea
{
    font-family: "Times New Roman", Times, serif; /* On modifie la police du texte
tapé l'intérieur des champs */
}
```

```

input:focus, textarea:focus /* Quand le curseur est sur un champ (ne marche pas sur
IE) */
{
    background-color: #FFFF99;
}

label
{
    color: blue; /* Colorer en bleu tous les labels (bah pourquoi pas ?) */
}

legend /* On met un peu plus en valeur les titres des fieldset */
{
    font-family: Arial, "Arial Black", Georgia, serif;
    color: #FF9933;
    font-weight: bold;
}

fieldset
{
    margin-bottom: 15px; /* Une marge pour séparer les fieldset */
    background-color: #FFFACC;
}

```

Les boutons d'action

Nous y sommes presque.

Nous avons vu la plupart des éléments que l'on peut intégrer dans un formulaire, mais il manque le plus important : le bouton de validation !

Il en existe 3 sortes :

- Le bouton d'envoi : il déclenche l'envoi du formulaire. Le visiteur se retrouve automatiquement transporté à la page indiquée dans l'attribut action du formulaire (ou dans un mail, dans notre cas).
Un bouton d'envoi se crée avec l'attribut **type="submit"**.
Vous pouvez lui ajouter un attribut **value** pour changer le texte à l'intérieur du bouton :
<input type="submit" />
- Le bouton de remise à zéro : il remet à leur valeur initiale, automatiquement, toutes les valeurs du formulaire. On doit utiliser cette fois l'attribut **type="reset"**
<input type="reset" />

- Le bouton qui sert à rien : c'est un bouton "générique" qui n'effectue aucune action particulière. Le formulaire n'est pas envoyé, il n'est pas remis à zéro : rien ne se passe. Ça vous servira principalement à lancer des scripts en Javascript.

`<input type="button" value="Je sers à rien" />`

Exercice

Reproduire le formulaire suivant le plus fidèlement possible

Commande - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

Exercice_6.html OK

Passez commande en ligne :

Civilité	☐ Mademoiselle ☒ Madame ☐ Monsieur
Nom / Prénom	
Adresse	Rue No Cidex ...
Code postal / Localité	Grenoble
Pays	France ▼
Système(s)	☒ Windows ☒ Macintosh ☐ Unix
Application(s)	Bureauutique DAO Statistiques SGBD Internet

Soumettre formulaire Effacer

Terminé

Récapitulatif CSS

formatage de texte.

Police, taille et décorations

Type	Nom	Valeurs possibles
Nom de police	font-family	font-family: "Arial Black", Arial, Verdana, serif; Indiquer les noms de polices possibles par ordre de préférence : Si le visiteur a la police 1, il l'utilisera. Sinon, il regarde s'il a la police 2, puis la police 3 etc. Utilisez des guillemets si le nom de la police comporte des espaces.
Taille du texte	font-size	Indiquez la taille du texte. Plusieurs unités sont possibles : - px (pixels) % (pourcentage, 100% = normal) - em (taille relative, 1.0 = normal) - ex (taille relative à la hauteur de la lettre "x". 1.0 = normal) - nom de taille :

Type	Nom	Valeurs possibles
		- xx-small : très très petit - x-small : très petit - small : petit - medium : moyen - large : grand - x-large : très grand - xx-large : très très grand
Gras	font-weight	bold : gras bolder : plus gras lighter : plus fin normal : pas gras (par défaut)
Italique	font-style	italic : italique oblique : autre façon de mettre en italique normal : normal (par défaut)
Décoration	text-decoration	underline : souligné overline : ligne au-dessus line-through : barré blink : clignotant none : normal (par défaut)
Petites capitales	font-variant	small-caps : petites capitales normal : normal (par défaut)
Capitales	text-transform	uppercase : tout mettre en majuscules lowercase : tout mettre en minuscules capitalize : début des mots en majuscules none : normal (par défaut)
Méga-propriété de police	font	Indiquez dans n'importe quel ordre des valeurs possibles pour font-weight, font-style, font-size, font-variant, font-family. Exception : le nom de la police (font-family) doit être placé en dernier dans la liste dans tous les cas. font : bold , 16px, Arial;

Alignement

Type	Propriété	Valeurs possibles
Alignement horizontal	text-align	left : à gauche (par défaut) center : centré right : à droite justify : texte justifié (prend toute la largeur de la page)
Alignement vertical	vertical-align	A utiliser dans des cellules de tableau top : en haut middle : au milieu bottom : en bas
Hauteur de ligne	line-height	Indiquer une valeur en pixels (px) ou en pourcentage (%)
Alinéa	text-indent	Indiquez une valeur en pixels (px) pour définir un retrait pour les paragraphes (peut être négatif). normal : le passage à la ligne est automatique (par défaut) nowrap : pas de passage à la ligne automatique, à moins qu'une balise XHTML comme ne soit présente.
Césure	white-space	pre : le passage à la ligne se fait tel que le texte a été saisi dans le code source (comme la balise XHTML <pre>)

Propriétés de couleur et de fond

Couleur

Type	Propriété	Valeurs possibles
Couleur de texte	color	Indiquer une couleur avec l'une des méthodes suivantes : - En tapant le nom de la couleur en anglais (black, blue, green, white, red...). - En indiquant la couleur en hexadécimal (#CC48A1) - En indiquant la couleur en RGB : rgb(128, 255, 0)
Couleur de fond	background-color	Même fonctionnement que color. Cela définit cette fois la couleur de fond du texte

Image de fond

Type	Propriété	Valeurs possibles
Image de fond	background-image	Indiquer l'url de l'image (notation absolue ou relative) background-image : <code>url("images/fond.png")</code> ; background-image : <code>url("http://www.site.com/images/fond.png")</code> ;
Fond fixé	background-attachment	fixed : le fond reste fixe quand on descend plus bas sur la page scroll : le fond défile avec le texte (par défaut) repeat : le fond se répète (par défaut)
Répétition du fond	background-repeat	repeat-x : le fond ne se répète que sur une ligne, horizontalement repeat-y : le fond ne se répète que sur une colonne, verticalement no-repeat : le fond ne se répète pas, il n'est affiché qu'une fois
Position du fond	background-position	2 façons de faire : - En notant une distance en px ou %, par rapport au coin en haut à gauche. - En utilisant des valeurs prédéfinies, une pour la verticale et une pour l'horizontale : top : en haut, verticalement center : au milieu, verticalement bottom : en bas, verticalement left : à gauche, horizontalement center : au centre, horizontalement right : à droite, horizontalement

Propriétés des boîtes

Dimensions

Type	Propriété	Valeurs possibles
Largeur	width	Valeur en px, %, ou encore "auto" (valeur par défaut, la largeur dépendra du texte à l'intérieur)
Hauteur	height	Idem
Largeur minimale	min-width	Indiquer une valeur, en pixels par exemple.
Largeur maximale	max-width	Idem
Hauteur minimale	min-height	Idem
Hauteur maximale	max-height	Idem

Marges extérieures

Type	Propriété	Valeurs possibles
Marge en haut	margin-top	Indiquer une valeur comme 20px, 1.5em...
Marge à gauche	margin-left	Idem
Marge à droite	margin-right	Idem
Marge en bas	margin-bottom	Idem
Méga-propriété de marge	margin	Indiquez de 1 à 4 valeurs à la suite. Selon le nombre de valeurs que vous mettez, la signification change : 1 valeur : ce sera la marge pour le haut, le bas, la gauche et la droite 2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde pour la gauche et la droite 3 valeurs : la première correspond à la marge du haut, la seconde aux marges à gauche et à droite, la troisième à la marge du bas 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche.

Marges intérieures

Type	Propriété	Valeurs possibles
Marge intérieure en haut	padding-top	Indiquer une valeur comme 20px, 1.5em...
Marge intérieure à gauche	padding-left	Idem
Marge intérieure à droite	padding-right	Idem
Marge intérieure en bas	padding-bottom	Idem
Méga-propriété de marge intérieure	padding	Indiquez de 1 à 4 valeurs à la suite. Selon le nombre de valeurs que vous mettez, la signification change : 1 valeur : ce sera la marge pour le haut, le bas, la gauche et

Type	Propriété	Valeurs possibles
		la droite
		2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde pour la gauche et la droite 3 valeurs : la première correspond à la marge du haut, la seconde aux marges à gauche et à droite, la troisième à la marge du bas 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche.

Bordures

Type	Propriété	Valeurs possibles
Epaisseur de la bordure	border-width	Indiquer une valeur en px.
Couleur de la bordure	border-color	Indiquer une valeur de couleur. none : pas de bordure (par défaut) hidden : bordure cachée solid : ligne pleine double : ligne double (nécessite une taille de bordure de 3px minimum) dashed : en tirets dotted : en pointillés inset : effet 3D "enfoncé" outset : effet 3D "surélevé" ridge : autre effet 3D
Type de bordure	border-style	Indiquer la couleur, l'épaisseur et le type de bordure pour la bordure gauche. L'ordre n'a pas d'importance. Exemple : <code>border-left: 2px inset blue;</code>
Bordure à gauche	border-left	Idem
Bordure en haut	border-top	Idem
Bordure à droite	border-right	Idem
Bordure en bas	border-bottom	Idem
Méga-propriété de bordure	border	Indiquera l'apparence des bordures en haut, à droite, en bas et à gauche.

Propriétés de positionnement et d'affichage

Affichage

Type	Propriété	Valeurs possibles
Type d'élément	display	none : l'élément ne sera pas affiché block : l'élément devient de type "block" (bloc, comme <p>) inline : l'élément devient de type "inline" (en ligne, comme) list-item : l'élément devient de type "élément de liste à puce" (comme)
Affichage	visibility	hidden : masqué visible : visible (par défaut) Attention : display:none; fait complètement disparaître l'élément, tandis que visibility:hidden; masque l'élément, qui continue quand même à prendre de la place sur l'écran.
Afficher seulement une partie	clip	Indiquer 4 valeurs comme ceci : <code>clip: rect(valeur1, valeur2, valeur3, valeur4);</code> Cela permet de n'afficher qu'une partie d'un élément. rect() permet d'indiquer les coordonnées du rectangle qui sera affiché. Les valeurs 1 à 4 correspondent respectivement aux coins haut, droite, bas et gauche du rectangle.
Limiter les dimensions	overflow	visible : tout l'élément sera affiché (par défaut). hidden : l'élément sera coupé s'il dépasse les limites définies par height et width. On ne pourra pas voir la partie du texte coupée. scroll : tout comme hidden, l'élément sera coupé s'il dépasse les limites. Toutefois, cette fois le navigateur ajoutera des barres de défilement pour qu'on puisse voir la suite du texte. auto : c'est le navigateur qui décide d'ajouter des barres de défilement ou pas en fonction des cas. Bien souvent, utiliser cette valeur revient à utiliser la valeur "scroll".

Positionnement

Type	Propriété	Valeurs possibles
Flottant	float	left : flottant à gauche right : flottant à droite

Type	Propriété	Valeurs possibles
Stopper un flottant	clear	none : pas de flottant (par défaut) left : supprime l'effet d'un flottant à gauche précédent right : supprime l'effet d'un flottant à droite précédent both : supprime l'effet d'un flottant précédent, qu'il soit à gauche ou à droite none : pas de suppression de l'effet du flottant (par défaut)
Type de positionnement	position	absolute : position absolue par rapport au coin en haut à gauche fixed : position fixe (fonctionne comme la position absolue). L'élément reste à sa position même quand on descend plus bas dans la page. relative : position relative, par rapport à la position "normale" de l'élément static : positionnement normal (par défaut)
Position par rapport au haut	top	Valeur en px, %, em... A utiliser pour un positionnement absolu, fixe ou relatif.
Position par rapport au bas	bottom	Idem
Position par rapport à gauche	left	Idem
Position par rapport à droite	right	Idem
Ordre d'affichage	z-index	En cas de positionnement absolu par exemple, si 2 éléments se chevauchent, z-index permet d'indiquer quel élément doit être affiché au-dessus de l'autre. Indiquez un nombre. Plus ce nombre est élevé, plus l'élément sera affiché en avant. Par exemple, si vous avez 2 éléments positionnés en absolus avec un z-index de 10 pour l'un et de 20 pour l'autre, c'est celui qui a un z-index de 20 qui sera affiché par-dessus.

Propriétés des listes

Type	Propriété	Valeurs possibles
Type de liste	list-style-type	- Pour les listes non ordonnées () : disc : un disque noir (par défaut). circle : un cercle. square : un carré. none : aucune puce ne sera utilisée. - Pour les listes ordonnées () : decimal : des nombres 1, 2, 3, 4, 5... (par défaut) decimal-leading-zero : des nombres commençant par zéro (01, 02, 03, 04, 05...). Ne fonctionne pas sur Internet Explorer upper-roman : numérotation romaine, en majuscules (I, II, III, IV, V...) lower-roman : numérotation romaine, en minuscules (i, ii, iii, iv, v...) upper-alpha : numérotation alphabétique, en majuscules (A, B, C, ...) lower-alpha : numérotation alphabétique, en minuscules (a, b, c, ...) lower-greek : numérotation grecque. Ne fonctionne pas sur Internet Explorer
Position en retrait	list-style-position	inside : sans retrait outside : avec retrait (par défaut)
Puce personnalisée	list-style-image	Indiquer l'url de l'image qui servira de puce. Exemple : list-style-image : <code>url("images/puce.png")</code> ;

Propriétés des tableaux

Type	Propriété	Valeurs possibles
Type de bordure	border-collapse	collapse : les bordures du tableau et des cellules sont mélangées. separate : les bordures du tableau et des cellules sont séparées (par défaut).
Cellules vides	empty-cells	show : les bordures des cellules vides sont affichées. collapse : les cellules vides sont masquées (par défaut). Indique la position du titre du tableau, défini via la balise <caption>
Position du titre	caption-side	top : en haut du tableau bottom : en bas du tableau left : à gauche du tableau right : à droite du tableau

Autres propriétés

Type	Propriété	Valeurs possibles
		auto : curseur automatique (par défaut) default : curseur standard pointer : curseur en forme de main, comme quand on pointe sur un lien text : curseur utilisé quand on pointe sur du texte wait : curseur utilisé pour indiquer une attente (sablier) progress : curseur utilisé pour indiquer une tâche de fond (curseur avec sablier) help : curseur en forme de point d'interrogation, indiquant une aide move : curseur en forme de croix, indiquant un déplacement possible
Curseur de souris	cursor	n-resize : flèche vers le nord ne-resize : flèche vers le nord-est e-resize : flèche vers l'est se-resize : flèche vers le sud-est s-resize : flèche vers le sud sw-resize : flèche vers le sud-ouest w-resize : flèche vers l'ouest nw-resize : flèche vers le nord-ouest url : curseur personnalisé, de type .cur ou .ani

Récapitulatif xhtml

Balises de premier niveau

Balises	Description
<html>	Balise principale de toute page web. On lui donne généralement 2 attributs : - xmlns : la liste des balises xhtml existantes (appelée espace de noms). - xml:lang : la langue utilisée sur votre page web. Utilisez "fr" pour un document en français.
<head>	En-tête de la page
<body>	Corps de la page

Le code minimal d'une page XHTML

```
<!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>Titre page</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <p>Contenu page</p>
  </body>
</html>
```

Balises d'en-tête

Entre <head> et </head>

Balise	Description
	Cette balise permet d'indiquer certaines informations sur la page web. - Inclure une feuille de style CSS : <link rel="stylesheet" media="screen" type="text/css" title="Mon design" href="design.css" /> - Feuille de style alternative <link /> <link rel="alternate stylesheet" media="screen" type="text/css" title="Design rose pour les filles" href="design_poney_rose.css" /> - Spécifier une page d'accueil alternative : <link rel="start" title="Accueil" href="index.html" /> - Spécifier une page d'aide pour le site : <link rel="help" title="Accessibilité" href="accessibilite.html" />

Balise

Description

- Spécifier un « favicon » (c'est ne mini-icône qui s'affiche dans les onglets)

```
<link rel="shortcut icon" type="image/x-icon" href="favicon.ico" />
```

Cette balise permet de définir les propriétés de la page web.

- Auteur de la page

```
<meta name="author" content="arrobe.fr" />
```

- Description de la page

```
<meta name="description" content="La page personnelle de Jean Dupont" />
```

- Mots-clés de la page

```
<meta name="keywords" content="expériences, laboratoire, chimie" />
```

- Adresse de contact (Ou le meilleur moyen pour se faire spammer)

```
<meta name="reply-to" content="monadresse@email.com" />
```

- Empêcher la mise en cache de la page par le navigateur

```
<meta /> <meta http-equiv="pragma" content="no-cache" />
```

- Table de caractères

```
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
```

- Redirection automatique au bout de 5 secondes

```
<meta http-equiv="refresh" content="5; URL=http://www.arrobe.fr" />
```

- Pour empêcher l'indexation de la page par les robots

```
<meta name="robots" content="noindex" />
```

- Pour empêcher les robots d'indexation de suivre les liens de la page

```
<meta name="robots" content="nofollow" />
```

A propos de **Description** et de **Keywords** : La description et les mots-clés de la page ont été créés pour favoriser l'indexation par les moteurs de recherche. Mais ça fait bien longtemps qu'ils n'en tiennent plus compte ! Inutile de surcharger ces balises avec 45 mots clés...

Permet de placer un script (ex : Javascript) :

- Version « dans le corps du texte »

```
<script type="text/javascript">
/* Votre script ici */
</script>
```

- Version « externe »

```
<script src="fichier.js" type="text/javascript"></script>
```

Permet de définir du code CSS directement dans la page. (pour la version « fichier inclus, voir <link>)

```
<style type="text/css">
/* Votre css ici */
</style>
```

Titre de la page web.

```
<title>Mon super cours de xhtml</title>
```

Choisissez bien votre titre car les moteurs de recherche donnent de l'importance aux mots qui se trouvent dans le titre.

Balises de contenu

Entre <body> et </body>

Balises de structuration du texte

Balise

Type

Description

<acronym>	Inline	Sert à définir des acronymes : Le texte entre les balises apparaît; et le titre apparaît en infobulle au survol. <pre><acronym title="Central Intelligence Agency">C.I.A.</acronym></pre>
<blockquote>	Block	Citation longue. Vous devez obligatoirement mettre une balise de paragraphe à l'intérieur du blockquote. <pre><blockquote> <p> Texte de la citation </p> </blockquote></pre>
<cite>	Inline	Citation (moyenne)
<q>	Inline	Citation (courte)
<sup>	Inline	Mise en exposant
<sub>	Inline	Mise en indice
	Inline	Mise en valeur (forte) (Généralement mis en gras).

Balise	Type	Description
	Inline	Mise en valeur (faible) (Généralement mis en italique.)
<h6>	Block	Titre de niveau 6
<h5>	Block	Titre de niveau 5
<h4>	Block	Titre de niveau 4
<h3>	Block	Titre de niveau 3
<h2>	Block	Titre de niveau 2
<h1>	Block	Titre de niveau 1
	Inline	Insère une image. Attributs obligatoire : src (pour indiquer l'adresse de l'image) alt (pour indiquer un texte de remplacement) <code></code>
<a>	Inline	Lien hypertexte. (Indiquez l'url de destination grâce à l'attribut href) <code>Rendez-vous sur l'autre page</code>

	Inline	Retour à la ligne
<p>	Block	Paragraphe
<hr />	Block	Crée une ligne de séparation horizontale
<address>	Block	Permet d'indiquer une adresse, ou éventuellement l'auteur d'un document. (Le texte est généralement mis en italique.)
	Inline	Permet d'indiquer un texte qui a été supprimé. (Le texte est généralement barré.)
<ins>	Inline	Permet d'indiquer un texte qui a été inséré. (Le texte est généralement souligné.)
<dfn>	Inline	Permet d'indiquer une définition.
<kbd>	Inline	Permet d'indiquer un code que doit taper le visiteur.
<pre>	Block	Le texte à l'intérieur de la balise <pre> sera affiché tel qu'il a été tapé dans le code (espaces et entrées compris). Une police de taille fixe est utilisée.

Balises de liste

Balise	Type	Description
	Block	Liste à puces non numérotée. Vous devez mettre un <code></code> par élément de la liste. Exemple : <code></code> <code>Un élément</code> <code>Un autre élément</code> <code></code>
	Block	Liste à puces numérotée. Vous devez mettre un <code></code> par élément de la liste. Exemple : <code></code> <code>Elément n°1</code> <code>Elément n°2</code> <code></code>
	list-item	Permet de créer un élément de liste. Le type de la balise est particulier car elle n'est ni block ni inline. On dit qu'elle est de type list-item.
<dl>	Block	Liste de définitions. Vous devez alterner chaque terme <code><dt></code> par sa définition <code><dd></code> . Exemple : <code><dl></code> <code><dt>Porte</dt></code> <code><dd>Ouverture dans un mur permettant d'entrer et de sortir</dd></code> <code><dt>Théâtre</dt></code> <code><dd>Lieu où l'on représente des ouvrages dramatiques</dd></code> <code></dl></code>
<dt>	Block	Terme à définir
<dd>	Block	Définition du terme

Balises de tableau

Balise	Type	Description
<table>	Block	Délimite un tableau.
<caption>	-	Titre du tableau
<tr>	-	Ligne de tableau
<th>	-	Cellule d'en-tête du tableau (généralement mise en gras)
<td>	-	Cellule du tableau
<thead>	-	Balise non obligatoire permettant d'insérer l'en-tête du tableau.

Balise	Type	Description
		Si vous choisissez d'utiliser <thead>, <tfoot> et <tbody>, vous devez les mettre dans l'ordre suivant dans votre code source : 1. <thead> 2. <tfoot> 3. <tbody>
<tbody>	-	Balise non obligatoire permettant d'insérer le corps du tableau
<tfoot>	-	Balise non obligatoire permettant d'insérer le pied du tableau

Balises de formulaire

Balise	Type	Description
		Délimite un formulaire. 2 attributs obligatoires : method : indique la méthode d'envoi du formulaire (get ou post). action : URL de la page appelée par le bouton Submit.
<form>	Block	
<fieldset>	Block	Permet de regrouper plusieurs éléments d'un formulaire. Pour donner un titre à votre groupe, utilisez la balise <legend>
<legend>	Inline	Titre d'un groupe dans un formulaire. A utiliser à l'intérieur d'un <fieldset>
<label>	Inline	Titre d'un élément de formulaire. Mettre l'attribut for sur cette balise pour indiquer l'id de l'élément auquel correspond le label.
		Champ de formulaire. : - Zone de texte d'une ligne <input type="text" /> - Mot de passe (le texte est caché) <input type="password" /> - Envoi de fichier <input type="file" /> - Case à cocher <input type="checkbox" /> - Bouton d'option <input type="radio" /> - Bouton d'envoi <input type="submit" /> - Bouton de remise à zéro <input type="reset" /> - Champ caché <input type="hidden" /> Nomer les champs grâce à l'attribut name
<input />	Inline	
<textarea>	Inline	Zone de saisie multiligne. Utiliser rows et cols (nombre de lignes et colonnes) pour fixer la taille ou bien le faire en CSS grâce aux propriétés width et height . Liste déroulante. : Utilisez la balise <option> pour créer chaque élément de la liste : <select name="pays"> <option value="F">France</option> <option value="E">Espagne</option> <option value="I">Italie</option> </select>
<select>	Inline	
<option>	Block	Élément d'une liste déroulante
<optgroup>	Block	Groupe d'éléments d'une liste déroulante. A utiliser dans le cas d'une grande liste déroulante. Vous devez utiliser l'attribut label pour donner un nom au groupe.

Balises génériques

Les balises génériques (aussi appelées balises universelles) sont des balises qui n'ont pas de sens sémantique.

En effet, toutes les autres balises XHTML ont un **sens** : <p> signifie "Paragraphe", <h2> signifie "Sous-titre" etc.

Il y a 2 balises génériques : l'une est inline, l'autre est block.

Balise	Type	Description
	Inline	Balise générique de type inline
<div>	Block	Balise générique de type block

Ces balises ont un intérêt uniquement si vous leur donnez un attribut class, id ou style :

- **class** : indique le nom de la classe CSS à utiliser.
- **id** : donne un nom à la balise. Ce nom doit être unique sur toute la page.
- **style** : cette balise vous permet d'indiquer directement le code CSS à appliquer.

Ces 3 attributs ne sont pas réservés aux balises génériques : vous pouvez aussi les mettre sur la plupart des autres balises sans aucun problème