

TP n°2 – Site web

Partie 1 : Le langage (X)HTML

Source : <http://lemoigno.fr>

Objectif : Apprendre à créer une page Web basique

- Les balises de base du langage (X)HTML
- Insérer des images
- Insérer des liens hypertextes
- Faire des listes
- Faire des tableaux

1.1. Les bases du (X)HTML (site w3schools.com)

Une **page web** est un fichier contenant du texte, généralement dans le langage **HTML** ou **XHTML**.

Remarque sur le XHTML : XHTML est la nouvelle version de HTML.

Pour voir cela, faites un clic droit sur une page web, par exemple <http://lemoigno.fr/Contrôleur/TP7/bases.html> et cliquez sur Code source de la page (ou Menu Afficher → Code source de la page). Une fenêtre s'ouvre : elle contient le code de la page en langage HTML. Analysez le contenu du fichier HTML.

Le **document** est la page web telle que nous la visionnons. Il faut donc 2 outils : un outil d'**édition** qui permet d'éditer le texte du fichier et un outil de **visualisation** pour voir le document final.

Il existe différents outils d'édition comme les éditeurs de texte brut (gedit gratuit utilisé en TP) ou des outils plus dédiés (comme geany ou KompoZer pour Mac) qui permettent de taper le code HTML ou d'utiliser les assistants graphiques. Les outils de visualisation correspondent aux navigateurs tels que Firefox (utilisé en TP), Internet Explorer, ...

Récupérez le fichier pp.html sur l'espace pédagogique Moodle et copiez-le dans le répertoire TP2.

Il contient certaines des principales balises (ou marqueurs) (X)HTML ainsi qu'un faux texte destiné à remplir cette page de test. Ce fichier vous servira de canevas pour cette partie du TP.

Ouvrez le fichier « pp.html » enregistré dans votre répertoire TP2, avec gedit (accessoirs → gedit) et ouvrez à nouveau ce même fichier avec le navigateur firefox (clic droit → firefox). Gedit vous montre le contenu du fichier en code HTML et le navigateur firefox affiche le résultat de son interprétation du code.

Astuce : Pour plus de facilité, vous pouvez afficher les deux fenêtres en plein écran et dans deux bureaux distincts. Pour passer d'un bureau à l'autre : Ctrl + Alt + →.

Remarque : Pour voir l'effet d'une modification du texte (document ou marqueur) dans l'affichage, il faut sauver le fichier .html modifié (CTRL + s) puis rafraîchir l'affichage du navigateur avec la touche F5. De cette façon, il n'y a pas besoin de rouvrir ni l'éditeur, ni le navigateur.

1.1.1. Les principales balises du langage

Le XHTML est un langage « à **balises** ». Un fichier XHTML a pour extension .htm ou .html et est de type « texte ». Les balises ont toujours la forme : `< nom >`. On distingue deux types de balises : les balises locales, indiquant un effet à un endroit donné, qui s'écrivent `< nom / >` (par exemple, `< br / >` pour un saut de ligne) et les balises qui ont une portée et qui ont donc un début et une fin : `< nom >.....< / nom >` (par exemple, `< h1 >...< / h1 >` pour un titre de niveau 1).

Essayez maintenant de répondre aux questions suivantes en utilisant différentes options des animations:

Remarquez la casse (majuscule ou minuscule) des différentes balises (head et body, par exemple): cette casse a-t-elle une importance?

Où retrouve-t-on ce qui est inscrit entre les marqueurs `< title >` et `< / title >`?

Quel est l'effet du marqueur `< br >`?

Que trouve-t-on à l'intérieur du marqueur `< !.... >`?

A quoi correspond `& copy;`?

Enlever les marqueurs fermants `< / html >`, `< / body >` et `< / head >`: quel est l'effet de cette suppression sur l'affichage par le navigateur ? Penser à copier-coller les lignes supprimées dans un autre fichier, afin de pouvoir les remettre lorsque vous aurez terminé.

Enlever les marqueurs ouvrants correspondants aux marqueurs fermants précédents: cela change-t-il l'affichage de la page sur le navigateur?

Enlever à présent le marqueur `< / h2 >`: comment l'affichage est-il modifié? Jusqu'où s'étend cette modification?

À quelle structure correspond le marqueur `< p >`? Que remarquez-vous pour cette balise ?

Modifiez avec la souris la largeur de la fenêtre du navigateur: que constatez-vous ?

Un langage très tolérant : On constate que le langage HTML, contrairement à un langage de programmation comme le JAVA est très tolérant et qu'un navigateur peut afficher correctement un fichier html très mal écrit. Il est bien évident cependant que dans ce cas, l'affichage est aléatoire puisqu'il dépend de la tolérance de chaque navigateur. Il est donc vivement recommandé de toujours respecter au mieux la norme du langage.

1.1.2. Les attributs des balises

Certaines balises possèdent des attributs pour indiquer des paramètres supplémentaires. Les attributs ne se mettent que dans la balise d'ouverture. Ils ont un nom (en minuscule) et une valeur (entre guillemets doubles, valeur « libre »).

Exemple : ``.

Testez les attributs bgcolor et text de la balise body. Les valeurs que peuvent prendre ces deux attributs sont des couleurs écrites sous la forme #HHHHHH, où H est un nombre hexadécimal.

Remplacer `<body>` par `<body bgcolor=#HHHHHH>`

Des attributs dépréciés : les attributs des balises sont donnés à titre de la compatibilité avec les plus anciennes versions de HTML. Elles tombent peu à peu en désuétude et leur rôle est maintenant tenu par les feuilles de style.

1.1.3. Une apparence basée sur le contenu

Si le HTML est si tolérant, c'est qu'il n'est pas un langage de programmation mais un langage de présentation de documents. Il faut toujours avoir à l'esprit quand on écrit du HTML ou plus généralement quand on crée une page Web, que tout doit être basé sur le contenu du document que l'on veut rendre public. En effet, le créateur des documents n'a qu'une influence relative sur la façon dont ses pages seront visualisées puisque cela dépend fortement du navigateur mais aussi du type de matériel de la personne qui visualise les documents.

Pour cela, il existe différents marqueurs qui vont correspondre à une fonction particulières de certaines parties du document (titres, sous-titres, listes, mise en exergue...) et qui vont chercher à mettre en évidence ces parties, par exemple en changeant la fonte (mais pas toujours).

Pour mettre en évidence les titres de différents niveaux (titres, sous-titres,...), les marqueurs sont du type `<hi>...</hi>` où i est un entier entre 1 et 6: `<h1>` correspond au niveau le plus élevé (titre principal) et `<h6>` au niveau le plus bas.

`<p>...</p>` délimitent les paragraphes

`
` introduit un saut de ligne sans saut de paragraphe

`... ` sert à mettre en évidence (« un peu » en valeur)

`...` sert à mettre « beaucoup » en valeur

Il existe également des marqueurs qui ne modifient pas l'aspect du contenu sur la base de sa fonction. Ses styles physiques sont déconseillés mais parfois bien pratiques...

`<i> </i>` sert à mettre en italique

`<u> </u>` sert à souligner

` ` sert à mettre en gras

`^{....}` met en exposant

`_{....}` met en indice

Testez ces différentes balises sur la page pp.html en ajoutant les balises de titre de différents niveaux, en mettant en évidence certaines parties du texte (em, strong, gras, italiques...), en écrivant la formule chimique $[NH_4]^+$.

1.2. Insérer des images et des liens

1.2.1. Insérer des images

Un document HTML est très souvent un document multimédia, c'est à dire qu'il comprend des images et des sons en plus du texte. Cependant, le fichier HTML proprement dit est uniquement un fichier de texte mais qui peut contenir des liens vers des fichiers multimédias.

Le marqueur qui permet d'insérer les images est ****. Ce marqueur comporte un attribut obligatoire qui indique où se trouve le fichier image correspondant, c'est l'attribut **src** = "chemin vers fichier". Ainsi, on aura toujours au moins une syntaxe de ce genre:

Il existe d'autres attributs pour ce marqueur ****, qui sont eux facultatifs:

alt = "texte descriptif" : permet d'afficher un texte décrivant l'image si le navigateur n'arrive pas à l'afficher ou lorsque la souris passe sur l'image. C'est une chaîne de 1024 caractères au maximum.

align=alignement : permet de contrôler l'alignement des images par rapport au texte environnant. Les spécifications HTML définissent 5 valeurs d'alignement: left, right, top, bottom, middle. Les deux premières positionnent l'image sur la marge gauche (left) ou droite (right) du texte qui se place donc sur la partie restante des lignes pour compléter la page. Les trois autres valeurs positionnent verticalement l'image par rapport au texte: top aligne le sommet de l'image sur le plus haut élément du texte environnant, bottom aligne le bas de l'image sur le bas du texte (c'est la valeur par défaut). La valeur middle est traitée différemment par InternetExplorer et Netscape: Netscape aligne seulement le milieu de l'image avec la ligne de base du texte tandis que IE aligne le milieu de l'image avec le milieu de l'élément (lettres ou image) le plus haut de la ligne.

hspace="nombre", **vspace**="nombre" : associés à l'attribut align, permettent de laisser un espace libre (exprimé en pixels) entre l'image et le texte.

border="nombre" : lorsqu'une image correspond à un hyperlien (on clique dessus pour appeler un autre document, voir la section 2), elle peut être signalée par une bordure bleue (de la même façon qu'un texte associé à un hyperlien est de couleur différente). L'attribut border précise la taille de la bordure bleue en pixels.

width="nombre", **height**="nombre" : permettent de fixer la largeur (width) et la hauteur (height) de l'image, soit en pixels, soit en pourcentage de la taille de la fenêtre. On peut ainsi éviter le débordement des images trop grandes mais également gagner du temps lors du chargement: le navigateur réserve directement la place demandée pour l'image au lieu de devoir la calculer et l'adapter en fonction de l'image chargée. Attention toutefois de conserver le rapport largeur/hauteur sous peine de distorsion.

Récupérez les photos photo1.jpg et photo2.jpg et insérez-les dans la page, la première dans le premier paragraphe à droite, la deuxième dans le dernier paragraphe en bas.

1.2.2 Insérer des liens hypertextes

Les liens hypertextes sont la principale caractéristique des documents HTML et sont à l'origine de la création du World Wide Web. Ils permettent la navigation à partir du document courant, soit vers d'autres documents de tous types répartis sur le réseau, soit à l'intérieur du document lui-même.

Pour ancrer un lien sur un mot ou une phrase :

```
<a href="le lien">le mot ou la phrase sur lequel on veut cliquer pour nous renvoyer sur le lien définit</a>
```

Chaque lien extérieur à la page active est défini par une adresse ou URL (Uniforme Ressource Locator). Il existe plusieurs types d'URL :

- **liens HTTP** (HyperText Transfer Protocol): ils correspondent aux adresses d'autres documents HTML sur le réseau. Ces liens s'écrivent sous la forme `http://serveur:port/chemin` avec "serveur" qui est l'adresse de la machine qui abrite le document demandé. La partie "port" est facultative : elle indique la "porte d'entrée" dans le serveur (en général 80 pour le HTML). Le reste de l'adresse ("chemin") indique la partie de l'arborescence du serveur sur laquelle se trouve la page demandée. Ce chemin peut être d'une longueur quelconque et comporter plusieurs répertoires et sous-répertoire (rep1/rep2/rep3/...). Ce chemin se termine par le nom du document. Si aucun nom de fichier n'est indiqué, le navigateur essaie d'ouvrir un fichier "index.html": s'il n'en trouve pas, il affiche la liste des fichiers du répertoire.
- **les e-mails** (courrier électronique): ces liens appellent en fait un programme de courrier électronique pour envoyer un mail vers l'adresse indiquée. Le programme de courrier est appelé par la commande: `mailto:nom@adresse`

Un lien peut également pointer vers une partie précise d'un document html (page active), si on repère ce point précis par une ancre, qui est en fait un point de repère vers lequel on peut mettre un lien hypertexte.

Une ancre est l'attribut `id="nom"` ajouté à une balise quelconque pour la repérer. Si on a posé une ancre nommée "chap_1" sur le titre principal d'une page :

```
<h1 id="chap_1"> Grand titre </h1>
```

, on pourra arriver directement à ce titre par le lien

```
<a href="#chap_1">...</a>
```

cela signifie qu'il pointe vers l'ancre chap_1 du même fichier (#) : c'est donc un moyen de créer des liens à l'intérieur d'une page un peu longue.

Dans le fichier pp.html, créez un lien vers la page principale de l'Université de Montpellier (<http://www.umontpellier.fr/>).

Faites en sorte que cette adresse soit un lien cliquable qui permet de vous envoyer un mail sur votre adresse institutionnelle.

Créer un lien depuis les images insérées dans le document vers le site Flickr (<https://www.flickr.com/>) où elles ont été récupérées (sous licence Creative Commons). Vous penserez à créer autour de l'image une bordure d'épaisseur 2 pixels.

Vous devez obtenir un aspect semblable à l'image pp.jpg (les couleurs pouvant être différentes).

1.3. Les tableaux et les listes

1.3.1. Les tableaux

En (X)HTML, les tableaux sont formés de lignes, contenant un certains nombres de cellules.

Un tableau sera compris entre les balises `<table></table>`.
 Chaque ligne est délimitée par `<tr></tr>` (pour Table Row).
 Chaque case est définie par `<td></td>` (pour Table Data) ou par `<th></th>` (pour Table Header) s'il s'agit d'une case de la ligne d'en-tête.
 Le contenu des cases peut être du texte ou des images.

La balise `<table>` admet un certains nombres d'attributs :

align: alignement du tableau par rapport au texte environnant. Il peut prendre les valeurs right, left ou center.
bgcolor : comme pour body, permet de donner la couleur de fond du tableau.
border= taille en pixels : épaisseur de la bordure du tableau (0 par défaut)
bordercolor=couleur : coloration de la bordure.
cellspacing=valeur en pixels : contrôle l'espace entre deux cellules adjacentes ou entre le bord du tableau et les cellules (2 pixels par défaut).
cellpadding=valeur en pixels : contrôle l'espace entre le bord d'une cellule et son contenu (1 pixel par défaut).
width=valeur en pixel ou en % : contrôle la largeur du tableau en % de la largeur de la page.

Récupérez le fichier table.html dans votre répertoire TP2. Affichez-le dans votre navigateur et ouvrez-le avec un éditeur de texte.

Faites-en sorte de donner à votre tableau un fond de couleur "wheat" (bgcolor) avec des bordures de 3 pixels de large (border), de couleur gris foncé (#494949). Centrez le tableau dans la page (align).

Quelle est, par défaut, la largeur du tableau? Que se passe-t-il si on impose une largeur de 120% ou de 25%?

Quelles sont les deux caractéristiques du contenu des cases th?

Que se passe-t-il si on enlève les caractères dans la première case du tableau? À quoi correspondent-ils?

Les balises `<td></td>` et `<th></th>` admettent des attributs dont :

colspan=nombre de colonnes fusionnées : fusionne les colonnes.
rowspan=nombre de lignes fusionnées : fusionne les lignes

		Préférences	
		Kumquats frais	Kumquats au sirop
sexe	Hommes de moins de 18 ans	45%	55%
	Hommes de plus de 18 ans	51%	49%
	Femmes de moins de 18 ans	58%	44%
	Femmes de plus de 18 ans	63%	37%

Utilisez ces deux attributs pour donner à votre tableau une allure voisine de celle de l'image table.jpg.

1.3.2. Les listes

Il y a deux types de listes en HTML: les listes ordonnées et les listes non ordonnées.

Comme leur nom l'indique, les **listes non ordonnées** regroupent des éléments qui n'ont pas besoin d'être présentés dans un ordre particulier (par exemple une liste de liens hypertextes vers des pages concernant un même sujet). Ce type de liste est compris entre les marqueurs `` et ``. Chaque élément de la liste est signalé par le marqueur `` et se termine par ``.
Remarque : Le marqueur `` accepte l'attribut **type**="type de puce" qui permet de choisir le motif devant chaque article de la liste entre disc, circle et square.

Par exemple

```
<ul>
  <li> <a href="url1"> adresse1 </a></li>
  <li> <a href="url2"> adresse2 </a></li>
  <li> <a href="url3"> adresse3 </a></li>
</ul>
```

donnera le résultat suivant:

- [adresse1](#)
- [adresse2](#)
- [adresse3](#)

Les **listes ordonnées** sont par exemple utilisées pour faire un index des chapitres d'un document, chaque élément de liste étant un lien vers un chapitre, dans l'ordre du texte bien entendu. Elles sont signalées par les balises `` et ``. Les éléments de listes sont, comme pour les listes non ordonnées signalés par ``.

Par exemple,

```
<ol>
  <li> Chapitre 1 </li>
  <li> Chapitre 2 </li>
  <li> Chapitre 3 </li>
</ol>
```

produira comme résultat:

1. Chapitre 1
2. Chapitre 2
3. Chapitre 3

Utilisez les ancrés vues au chapitre précédent et les listes pour ajouter un index au début du fichier pp.html qui pointera vers les débuts de chaque paragraphe.

1.4. A retenir

Le (X)HTML est un langage de présentation de données. C'est un langage utilisant des marqueurs. Il est très tolérant : en cas d'erreur ou d'oubli de certains marqueurs, il affichera tout de même quelque chose mais l'aspect obtenu sera certainement différent de celui souhaité.

Une page (X)HTML n'est qu'une proposition d'affichage : c'est le navigateur de celui ou celle qui lira la page (et son environnement) qui est responsable de l'affichage final. Il faut donc utiliser au maximum des marqueurs basés sur le sens du texte plutôt que des marqueurs physiques qui cherchent à produire un effet visuel.

Partie 2 – feuilles de style

Objectif : apprendre à utiliser les feuilles de style : la syntaxe de base, les styles contextuels, les classes de style, les divisions.

2.1. Les feuilles de style

Lorsque l'on réalise un ensemble de pages web, par exemple si on construit un site Internet, il est bien plus judicieux de séparer le contenu de la forme. La façon la plus « économique » de le faire est fournie par les **feuilles de style en cascade** (ou **CSS** pour Cascading Style Sheet). Ces feuilles CSS sont des fichiers munis de l'extension .css qui vont contenir toutes les indications de style (fond d'écran, fontes, couleurs, etc) tandis que le fond sera cantonné aux fichiers .html. En général, un fichier de style unique est utilisé pour toutes les pages d'un style: quand on veut changer celui-ci, il suffit de changer quelques lignes dans le fichier .css, sans toucher à tous les fichiers .html.

2.1.1. Principe

Pour indiquer à un navigateur que le style d'une page html sera donné dans une feuille css externe, il faut ajouter un lien vers cette dernière dans l'en-tête de la page html :

```
<head>
...
<link rel=stylesheet href= « url du fichier de style » type=text/css>
...
</head>
```

Il n'est pas nécessaire que la feuille de style soit dans le même répertoire que les pages html puisque son adresse est indiquée. On peut même en indiquer plusieurs, sous réserve qu'elles portent des noms différents (ceci peut être utile si l'on veut distinguer des styles différents pour l'affichage à l'écran, pour les mobiles, pour l'impression,...).

2.1.2. Syntaxe

La syntaxe CSS est assez simple :

```
nom_balise {
  propriété1: valeur;
  propriété2: valeur;
  ....
  propriétén: valeur;
}
```

Chaque bloc de style commence par le nom d'une balise HTML (body, h2, ul,...) suivie des propriétés de style qu'on veut donner au contenu de cette balise. Ces propriétés sont contenues entre deux accolades et se terminent par un point virgule (y compris s'il s'agit d'une propriété unique).

Remarque : même si ce n'est pas obligatoire, il est très recommandé de présenter les feuilles de style selon le modèle donné ci-dessus, en présentant une propriété par ligne: cela fait beaucoup pour la clareté de votre feuille de style...

2.1.3. Les principales propriétés

Ceci n'est qu'une liste partielle, qui permet tout de même d'obtenir une grande variété de styles. Les valeurs soulignées sont celles utilisées par défaut.

Attention, les noms de certaines propriétés sont proches du HTML mais pas identiques: par exemple background-image (css) est différent de background (HTML).

background-image : url du fichier image permet de spécifier une image de fond.

background-color : couleur permet de définir une couleur de fond.

border-style : dashed, dotted, double, groove, inset, outset, ridge, solid, none spécifie le style des bordures d'un élément. Attention: par défaut, cette propriété a la valeur none, c'est à dire qu'il n'y a pas de bordure.

border-width : largeur en pixel définit la bordure d'un élément.

border-color : couleur définit la couleur de la bordure.

color : permet de définir la couleur d'écriture du texte

float : right, left, none détermine si un élément (en particulier une image) flottera à gauche ou à droite du texte environnant.

font-family : permet de changer la police d'un texte

font-size : xx-small, x-small, small, medium, large, x-large, xx-large ou taille en pixels définit la taille de la police utilisée.

font-style : normal, italic permet de demander l'affichage du texte en italique.

font-weight : normal, bold permet de demander l'affichage du texte en gras.

height : hauteur en pixel, en pourcent, auto hauteur d'un élément.

list-style-position : inside, outside dans un élément de liste qui s'étend sur plusieurs lignes, permet de garder l'indentation de la première ligne (valeur outside par défaut) pour les lignes suivantes ou limite l'indentation à la première ligne.

list-style-type : disc, circle, square, decimal, lower-alpha, lower-roman, none, upper-alpha, upper-roman définit les marqueurs ou puces des éléments des listes non ordonnées (circle, disc, square) ou ordonnées (decimal=chiffres, roman=chiffres romains ou alpha=lettres).

margin-left : largeur en pixel ou en % de la marge à gauche.

margin-right : largeur en pixel ou en % de la marge à droite.

margin-top : largeur en pixel ou en % de la marge en haut.

margin-bottom : largeur en pixel ou en % de la marge en bas.

padding(-left, -right, -bottom, -top) : cette propriété est analogue à la propriété margin mais indique l'écart entre un élément et son contenu. Les unités sont les mêmes que pour la propriété margin.

text-align : center, right, left, justify définit l'alignement horizontal du texte dans un élément.

text-decoration : line-through, none, overline, underline définit certains effets de caractères et permet d'obtenir du texte barré (line-through), souligné (underline) ou surligné (overline).

width : largeur en pixel, en pourcent, auto largeur d'un élément (exemple une image).

a:link : couleur du lien hypertexte à visiter (définir dans le css)

a:hover : couleur du lien hypertexte lorsqu'on passe la souris dessus (définir dans le css)

a:visited : couleur du lien hypertexte lorsque celui-ci a été visité (définir dans le css)

a:active : couleur du lien hypertexte au moment du clic souris sur le lien (définir dans le css)

Pour étudier l'effet de la propriété "float", ouvrez le fichier cassoulet.html que vous aurez copié dans votre arborescence. Complétez ce fichier en insérant l'image cassoulet.jpg au début du paragraphe, juste après la liste des ingrédients.

Liez ce fichier à une feuille de style cassoulet.css que vous créerez (fichier gedit) et dans laquelle vous définirez le style de l'image, en essayant en particulier les trois valeurs de "float".

Placez à présent l'image après le texte de ce premier paragraphe et testez à nouveau l'effet des différentes valeurs de float dans la feuille de style associée.

Complétez la feuille de style de façon à obtenir un aspect qui ressemble à l'image cassoulet.jpg.

La spécification de la largeur d'un élément associé à des valeurs des marges particulières permet de centrer un élément : modifiez la feuille de style du document précédent pour centrer le tableau.

2.1.4. Les styles contextuels

Il est également possible de définir le style d'un marqueur en fonction du contexte dans lequel ce marqueur apparaît. Par exemple, pour définir le style associé à un élément d'une liste non ordonnée (une puce en forme de cercle), on emploiera

```
ul li {
  list-style:circle;
}
```

si cette liste est elle-même un élément d'une liste non ordonnée, on souhaitera que la puce soit un carré et on ajoutera dans la feuille de style

```
ul ul li {
  list-style:square;
}
```

La suite ul ul li indique que le style est celui d'un élément de liste (li) d'une liste non ordonnée (ul) elle même élément d'une liste non ordonnée (ul).

Modifier la feuille de style liée au document précédent pour que les liens hypertextes apparaissent en vert foncé (darkgreen) s'ils sont situés dans un élément de liste (li).

2.1.5. Les classes de style

Il est très souvent pratique de définir plusieurs styles pour un même marqueur (X)HTML, par exemple de définir plusieurs style d'image (marqueur) ou de paragraphe (marqueur <p>).

On différencie alors les marqueurs du document HTML par l'attribut **class=nom** où nom est choisi pour identifier une des classes associée à ce marqueur (ne pas mettre un chiffre ou nombre). Dans la feuille de style, on écrira alors :

```
nom_marqueur.nom_classe {
....
style
....
}
```

On pourra par exemple dans son document HTML définir pour les paragraphes une classe résumé, une classe introduction, etc dont les styles seront indiqués dans la feuille de style par la succession :

```
p.résumé {
....
style
....
}
p.introduction {
....
style
....
}
```

Modifiez la feuille de style cassoulet.css et le document cassoulet.html pour définir un nouveau type de paragraphe : le paragraphe de type *note* devra être écrit sur fond jaune (background-color) avec une fonte x-small (font-size) et des marges à droite et à gauche de 60 pixels (margin). Appliquez ce style pour le dernier paragraphe commençant par « poser le plat... » de cassoulet.html.

2.2. Les divisions

2.2.1. Description

La balise HTML **div** permet de séparer un document en sections logiques, appelées **divisions**. Une division est un conteneur, c'est-à-dire un élément qui peut en contenir d'autres.

Vous allez apprendre à utiliser cet élément, en conjonction avec les feuilles de style.

Remarque : la balise `body` est également un conteneur puisqu'elle peut contenir d'autres éléments comme des paragraphes, des titres... ou des `div`.

2.2.2. Bases

Copiez le fichier `base.html` dans le répertoire TP2. Ouvrez-le avec `gedit` et avec votre navigateur `firefox` pour le visualiser.

Quatre sections `DIV` ont été faites dans le fichier, chacune contenant le même texte. Dans la fenêtre d'affichage du navigateur, les quatre paragraphes (identiques) apparaissent juxtaposés, sans distinction : les divisions n'ont pour l'instant, aucun aspect particulier.

Pour appliquer des styles de mise en forme aux divisions, vous allez utiliser une feuille de style dont vous connaissez à présent le maniement.

Pour cela, liez le fichier `base.html` à une nouvelle feuille de style `base.css` que vous créerez avec `gedit`.

Pour visualiser les quatre divisions, identifiez chacune d'entre elle par un nom de classe particulier (un, deux, trois et quatre respectivement) puis, dans la feuille de style, donnez-leur une couleur particulière. Vous donnerez également une couleur gris claire au corps de la page `html` (grâce à la feuille de style).

Quelle est la taille (largeur, hauteur) par défaut d'une division ?

Comment les divisions évoluent-elles quand on module la taille de la fenêtre du navigateur ?

En utilisant des valeurs négatives pour les marges à droite et à gauche sur une des quatre divisions, trouvez la largeur par défaut (à gauche et à droite) de la marge en pixels.

Par défaut, le contenu du `DIV` est collé à ses bords: la propriété **padding** permet d'espacer le contenu avec les bordures d'un `DIV`, mais aussi d'autres conteneurs (comme `body`).

Faites apparaître un écart du contenu avec les bords de divisions de 5 pixels à gauche et à droite et de 3 pixels en haut et en bas.

Autour de la troisième division (celle identifiée par `class=trois`), créez une bordure solide, noire, de largeur 2px. Que constatez-vous?

Dans chaque `DIV`, mettez le texte entre les balises `<p>` et `</p>` : quel est l'effet produit lorsque vous visualisez la page Web ? Conclusions ?

Après vos modifications vous devez avoir un aspect qui ressemble à l'image `base.jpg`

N'oubliez pas de sauvegarder votre fichier dans votre dossier TP2

2.3. À retenir

Les feuilles de style servent à séparer le contenu (inscrit dans le fichier (X)HTML) et la forme, qui est contenue dans le fichier de style.

Elles sont particulièrement utiles pour homogénéiser le style donné à plusieurs pages HTML. Elles facilitent également la maintenance de ces pages (par exemple, celles qui forment un site Web) car il suffit de la modifier pour modifier instantanément le style de toutes les pages liées à la feuille de style.

Les divisions permettent de définir des conteneurs flottants, dont l'agencement souple permet d'améliorer le rendu des sites Web.