

Ce TP utilise très largement des ressources de l'université du Havre (Guillaume Prevost): <http://www-lih.univ-lehavre.fr/~prevost/>

## TP HTML 4.01

### HTML à la base

Le principe général d'HTML est très simple: le texte (le contenu) est structuré par des balises qui en définissent la structure (le contenant). De plus, HTML est un langage hypertexte. Cela signifie qu'il est possible de définir des liens entre plusieurs documents ou au sein d'un même document. Les documents pointés par les liens peuvent se situer sur des machines éloignées, faisant partie d'Internet.

### Conteneurs

La plupart des balises modifient l'interprétation que fait le navigateur sur leur contenu. Puisque les balises délimitent du contenu elles sont faites d'une partie ouvrante et d'une partie fermante. Les balises ouvrantes sont faites d'une instruction entre < et >. Les balises fermantes sont faites de la même instruction, mais entre </ et >. Par exemple pour mettre du texte en italique on utilise l'instruction i comme une balise ainsi: <i>texte en italique</i>.

### Marqueurs

Toutes les balises ne sont pas destinées à contenir du texte. Certaines ne sont que de simples marqueurs. Ces balises ne nécessitent pas de fermeture. Par exemple le saut de ligne peut être écrit simplement <br> (abréviation de break) en HTML 4. En revanche, pour être compatible XHTML 1.0 il faut écrire <br /> ou bien <br></br> (les deux formes sont équivalents)

### Attributs

Les attributs servent à modifier les propriétés des balises. Ils sont optionnels. Les attributs se placent après l'instruction de la balise. Ils sont suivis du signe égal =, puis d'une valeur entre doubles guillemets. Par exemple, <body bgcolor="black">.

### Exemple

Nous verrons une grande partie des balises plus tard, mais voici déjà un exemple très simple:

```
<html>
<body>
  <h1>La page la plus simple</h1>
</body>
</html>
```

**Exercice 1:** Ouvrez un éditeur de texte quelconque et entrez l'exemple ci-dessus. Sauvegardez-le dans un fichier nommé "exercice1.html". Lancez un navigateur puis chargez ce fichier afin de l'afficher (dans firefox : fichier -> ouvrir un fichier) ou bien en double cliquant sur le fichier.

Attention : le navigateur peut lire les fichiers html, mais il ne peut pas exécuter de programmes php (par exemple). Pour obtenir le même fonctionnement que celui qu'on a sur le Web, il faut installer un serveur web (apache par exemple) et regarder les fichier *au travers* de ce serveur. Sur la machine virtuelle, nous avons installé (et lancé) un serveur apache, qui tourne sur le port 80 (port http par défaut). Nous avons également mis un raccourci sur le

bureau vers le répertoire « visible » par le serveur web. Tout fichier (ou répertoire) placé dans ce répertoire pourra être lu via le Web.

Pour visualiser ces fichiers, il faut lancer un navigateur Web, puis taper l'adresse suivante : <http://localhost/> Si on a créé un fichier « mapage.html » alors pour le voir il faut taper l'URL : <http://localhost/mapage.html>

L'adresse « localhost » signifie que le navigateur Web doit contacter le serveur sur la machine « locale », ce qui est pratique dans un cas comme celui du TP où les machines ne sont pas forcément connectées à Internet.

## Structure d'une page: la recette de base

Toutes les pages HTML utilisent la même structure de base, dont voici les ingrédients:

1. Le conteneur `<html>` contient la totalité du document HTML.
2. Le conteneur `<head>` contient des informations n'apparaissant pas sur la page HTML, mais la décrivant. Ces informations sont utilisées par exemple par les moteurs de recherche pour indexer les pages Web. Ce conteneur inclut en particulier la balise de titre `<title>`. Ce titre n'est pas celui qui apparaît sur la page, mais celui qui est utilisé par les moteurs de recherche, et dans la barre de navigation.
3. Le conteneur `<body>` contient ce qui doit être affiché: texte, images, sons...

**Exercice 2:** Créez un nouveau fichier avec le notepad qui reprend l'exemple 1, et ajoutez lui une balise `<head>` dans laquelle vous placerez une balise `<title>` contenant le titre que vous désirez. Sauvegardez dans un fichier nommé `exercice2.html`. Ouvrez ce fichier avec votre navigateur et notez les (petites) différences.

## Les attributs de `<body>`

La balise `body` accepte plusieurs attributs:

- \* `text`: permet de changer la couleur du texte.
- \* `bgcolor`: permet de changer la couleur du fond de la page.
- \* `background`: permet de spécifier une image de fond.

Les couleurs peuvent être spécifiées de deux façons: par un nom prédéfini, par un triplet rouge/vert/bleu. Les noms prédéfinis sont `red`, `green`, `blue`, `yellow`, `white`, `black`... (vous trouverez une liste complète ici). Les triplets sont plus complexes et nous ne les utiliserons pas pour l'instant. Par exemple, pour changer la couleur du texte en rouge on utilisera: `<body text="red">`.

## Triplets RGB

Un triplet est spécifié par un dièse (#) suivi de trois nombres à deux chiffres. Les chiffres sont en hexadécimal (0, 1, ... 9, A, B, ... , F). Le rouge vient en premier, puis le vert, et enfin le bleu. Donc pour produire du rouge on utilise `#FF0000`, pour du bleu `#0000FF`. Si on mélange du rouge et du vert on obtient du jaune: `#FFFF00`.

**Exercice 3:** Créez un nouveau fichier qui reprend l'exercice 2, et modifiez la balise body pour que le texte soit jaune et la couleur de fond bleue. Sauvegardez-le dans un fichier nommé exercice3.html.

*Remarque:* Ces attributs, bien que fonctionnels sont obsolètes. Le W3C préconise d'utiliser des feuilles de style (CSS). Nous avons déjà donné une petite présentation en cours, que nous allons compléter par la suite.

## Formatage de texte

La majeure partie d'une page HTML est constituée de texte. Cependant, le navigateur n'interprète pas le texte comme le ferait un éditeur:

- \* les tabulations sont transformées en espaces,
- \* plusieurs espaces successifs sont regroupés en un seul,
- \* les sauts de lignes sont ignorés.

Pour mettre en forme le texte il faut donc obligatoirement utiliser les balises.

**Exercice 4:** Reprenez l'exercice 2, et insérez deux lignes après la fin de la balise h1:

```
<html>
<body>
  <h1>La page la plus simple</h1>
  ici
  et là
</body>
</html>
```

Sauvegardez-le dans un fichier nommé exercice4.html. Chargez-le dans le navigateur, que constatez-vous?

### Caractères spéciaux

Certains caractères ne sont pas directement affichables par un navigateur. Les caractères spéciaux sont introduits par un caractère & et terminés par un point-virgule ; avec entre deux le nom du caractère spécial.

Par exemple pour insérer plusieurs espaces, on utilise le caractère spécial &nbsp; ; Bien sûr il n'est pas possible non plus d'insérer directement un < ou un > puisqu'ils délimitent normalement des balises. Il faut utiliser les caractères spéciaux &lt; et &gt; ;. Idem avec le & il faut utiliser &amp; ;.

## Titres, paragraphes, etc.

Nous avons déjà vu l'une des balises de titre: h1. Les balises de h1 à h6 existent. h1 définit le titre de plus haut niveau.

Créer un paragraphe revient à placer du code dans une balise p. Pour forcer un retour à la ligne on utilise br. Enfin, pour placer une ligne séparatrice horizontale entre deux parties de la page on utilise hr (Horizontal Rule).

Les balises p et h1..h6 admettent un attribut align qui permet de définir l'alignement horizontal du texte. Il peut prendre les valeurs left (par défaut), right, center ou justify.

**Exercice 5:** Reprenez l'exercice 2, et définissez deux titres de plus bas niveaux après h1. Après chaque titre, définissez plusieurs paragraphes alignés différemment. Séparez les paragraphes par des lignes horizontales.

Les balises span et div permettent de regrouper tout type de contenu HTML pour lui appliquer des propriétés. Pour appliquer une mise en page, on utilise la balise div. Elle groupe les objets à la manière d'un paragraphe en créant un bloc distinct, et supporte donc l'attribut align. La balise span groupe des éléments sur une ou plusieurs lignes, mais ne crée pas de paragraphe.

### Gras, italique, etc.

HTML fournit trois balises très utilisées pour faire ressortir du texte:

- \* b: permet de mettre le texte en gras.
- \* i: permet de mettre le texte en italique.
- \* u: permet de souligner le texte. (obsolète, il vaut mieux utiliser em)
- \* Grâce aux balises sup et sub, il est possible de mettre du texte en exposant ou en indice.

### Lister, énumérer...

HTML permet d'arranger des éléments en listes ordonnées ou en items. On crée une liste non-ordonnée avec la balise ul (Unordered List), et une liste ordonnée avec ol (Ordered List). Chaque élément d'une liste est délimité par la balise li (List Item).

Voici une liste non ordonnée, et une liste ordonnée:

```
<ul>
  <li>item</li>
  <li>item</li>
  <li>item</li>
</ul>
<ol>
  <li>élément 1</li>
  <li>élément 2</li>
  <li>élément 3</li>
</ol>
```

**Exercice 6:** Reprenez l'exercice 5, et ajoutez des listes des deux types. Vous pouvez modifier l'apparence de la liste non ordonnée avec l'attribut type dont la valeur peut être disc, circle ou square.

### Exercice libre ...

Définissez un thème et créez une page sur ce sujet en essayant d'utiliser tous les éléments vus précédemment. Vous pouvez utiliser des images que vous insérerez dans la page grâce à la balise img ainsi: .

### Les liens

HTML est un langage hypertexte. Cela signifie qu'il est possible de rompre la linéarité du texte pour se diriger vers une autre page web, ou tout autre type de ressource informatique.

Ceci est effectué grâce à un lien: c'est à dire la mise en relation entre deux parties du World Wide Web, une source et une destination. Un lien peut pointer sur tout type d'objet informatique, pas uniquement sur une page web.

Avec HTML, un lien est introduit grâce à la balise `a`. Il existe trois types de liens:

1. Les liens à l'intérieur de la page elle-même.
2. Les liens sur des pages ou ressources d'un même site.
3. Les liens sur des pages ou ressources d'autres sites.

Pour créer la destination d'un lien à l'intérieur d'une page on utilise l'attribut `name` de la balise `a`. Par exemple `<a name="truc">`. Il est ensuite possible d'accéder à ce lien à partir d'un autre point de la page en créant un lien source ainsi: `<a href="#un_nom">`. Notez le caractère `#`.

Mais il est aussi possible de créer un lien sur une autre page dans le même répertoire que la page web en cours. Par exemple si le répertoire courant contient deux pages `A.html` et `B.html`. La page `A` peut pointer sur la page `B` si elle contient un lien de la forme `<a href="B.html">allez en B</a>`.

Pour diriger un lien vers une page arbitraire du World Wide Web, il est nécessaire, en plus du nom de la page d'ajouter le protocole, et le nom du serveur Web contenant la page. Un tel chemin d'accès est nommé URL. Ainsi la page que vous lisez à pour URL:

`http://www.akengro.t1hosting.co.uk/in111/test.html`

Une URL se décompose ainsi: au début se trouve le protocole, `http`, propre à HTML. Ensuite se trouve le nom du serveur Web `www.akengro.t1hosting.co.uk`, et enfin se trouve le chemin de la page `/in111/test.html`

Comme il est possible de faire un lien à l'intérieur d'une page elle même en un point donné de cette page, cela est aussi possible de l'extérieur. Il suffit de rajouter un `#` puis le nom d'un lien destination. Par exemple, l'URL `http://www.akengro.t1hosting.co.uk/in111/test.html#fin` pointe sur la fin de la page précédente où apparaît la balise `<a name="fin">`.

**Exercice 7:** Créez une page HTML (ou réutilisez une page que vous avez déjà créé) contenant au moins une page de texte (ou d'images), et placez au début un lien vers la fin de la page. De même à la fin de la page, placez un lien vers le début.

**Exercice 8:** Créez une deuxième page HTML du même type, et faites en sorte qu'il soit possible d'aller de la page créée dans l'exercice 1 vers cette page. Dans cette page créez un lien "retour" permettant de retourner vers la page créée dans l'exercice 1.

**Exercice 9:** Dans la page créée à l'exercice 2, ajoutez des liens vers des sites Web divers. Utilisez `www.google.fr` pour trouver des sites.

**Exercice 10 (facultatif) :** Créez une série d'une dizaine de pages web dont vous êtes le héros contenant à la fin de chaque page différents liens déterminés en fonction des choix du lecteur pour orienter l'histoire à la manière d'un jeu de rôle.

## Les tables

Les tables permettent d'aligner des éléments HTML quels qu'ils soient (même une autre table) dans une grille. Une table HTML est faite de cellules individuelles arrangées en lignes. Vous pouvez en voir un exemple au début de cette page à propos du rappel des éléments déjà vus.

Une table est créée grâce à la balise `table`. Cette balise contient ensuite une série de balises `tr` (table row) qui définissent les lignes. Enfin chaque balise `tr` contient une série de balises `td` (table data) qui définissent les cellules.

Par exemple:

```
<table>
  <tr>    <td>(0,0)</td> <td>(0,1)</td>    </tr>
  <tr>    <td>(1,0)</td> <td>(1,1)</td>    </tr>
</table>
```

Il est possible d'ajouter un titre à une table grâce à la balise `caption`. Ainsi:

```
<table>
  <caption>exemple</caption>
  <tr>    <td>(0,0)</td> <td>(0,1)</td>    </tr>
  <tr>    <td>(1,0)</td> <td>(1,1)</td>    </tr>
</table>
```

Ce titre apparaît alors centré au dessus du tableau.

Il est possible de définir certaines cellules comme étant des lignes d'entête en remplaçant la balise `td` par la balise `th`. Par exemple:

```
<table>
  <caption>exemple</caption>
  <tr>    <th>col 1</th> <th>col 2</th>    </tr>
  <tr>    <td>(0,0)</td> <td>(0,1)</td>    </tr>
  <tr>    <td>(1,0)</td> <td>(1,1)</td>    </tr>
</table>
```

## Attributs des tables

La balise `table` supporte un grand nombre d'attributs. Par exemple il est possible de spécifier la largeur d'une table avec l'attribut `width`. Sa valeur peut soit être un nombre de pixels absolu, soit être un pourcentage de la taille disponible. Par exemple:

```
<table width="80%">
  <caption>exemple</caption>
  <tr>    <th>col 1</th> <th>col 2</th>    </tr>
  <tr>    <td>(0,0)</td> <td>(0,1)</td>    </tr>
  <tr>    <td>(1,0)</td> <td>(1,1)</td>    </tr>
</table>
```

Par défaut le contenu des cellules est centré pour les entêtes et aligné à gauche pour les autres cellules. Il est possible de changer ce comportement avec l'attribut `align` des balises `td` et `th`. Par exemple:

```
<table width="80%">
```

```

<caption>exemple</caption>
<tr>
    <th align="left">col 1</th>
    <th align="left">col 2</th>
</tr>
<tr>
    <td align="center">(0,0)</td>
    <td align="center">(0,1)</td>
</tr>
<tr>
    <td align="center">(1,0)</td>
    <td align="center">(1,1)</td>
</tr>
</table>

```

Les balises `td` et `th` peuvent elles aussi prendre en argument un attribut `width`. S'il contient un pourcentage, il définit la taille d'une colonne par rapport aux autres.

```

<table width="80%">
    <caption>exemple</caption>
    <tr>    <th width="70%">col 1</th>    <th>col 2</th>    </tr>
    <tr>    <td>(0,0)</td>                <td>(0,1)</td>    </tr>
    <tr>    <td>(1,0)</td>                <td>(1,1)</td>    </tr>
</table>

```

Notez qu'il n'est pas nécessaire de spécifier `width` sur toutes les cellules d'une colonne, seule la première suffit.

Parfois le contenu d'une cellule prend plusieurs lignes de texte. Dans ce cas les cellules se trouvant sur la même ligne dans la table se trouvent élargies verticalement. Par défaut le navigateur centrera verticalement le texte dans les cellules. Il peut être pratique de changer ce comportement. Ceci est effectué grâce à l'attribut `valign` des balises `td` et `th`. Par exemple:

```

<table width="200">
    <tr>
        <td>Un long texte oblige le navigateur
        à passer à la ligne.</td>
        <td valign="top">vide</td>
    </tr>
    <tr>
        <td valign="bottom">vide</td>
        <td>Une grande cellule élargit toutes
        les autres sur sa ligne.</td>
    </tr>
</table>

```

Il est aussi possible de modifier la couleur de fond d'une table, d'une ligne ou d'une cellule grâce à l'attribut `bgcolor`. Ainsi:

```

<table width="80%" bgcolor="#8888EE">
    <caption>exemple</caption>
    <tr bgcolor="#AAAAFF">
        <th>col 1</th>
        <th>col 2</th>
    </tr>
    <tr>
        <td bgcolor="#DDDDCC">(0,0)</td>
        <td>(0,1)</td>
    </tr>
    <tr>

```

```
<td>(1,0)</td>
<td bgcolor="#DDDDCC">(1,1)</td>
</tr>
</table>
```

Puisque par défaut les tables n'ont pas de bordure et que leurs cellules peuvent contenir n'importe quel élément HTML, elles permettent de créer bien plus que des tableaux. Elles sont, par exemple, couramment utilisées pour créer des mises en page. En effet HTML ne permet pas d'agencer facilement le texte autrement qu'en paragraphes consécutifs. Il n'existe pas de balises pour créer une note dans une marge, faire courir le texte sur deux colonnes ou insérer des images d'un côté ou de l'autre d'un texte. L'utilisation de tables permet de remédier à tout ces cas.

### Exercice 11:

Le lien suivant <http://www-lih.univ-lehavre.fr/~prevost/coursinfo/STPI/img/page.gif> propose une image représentant un site HTML dont la mise en page à été faite avec des tables imbriquées. Essayez de réaliser la même mise en page avec les textes et les images de votre choix.

**Exercice 12:** Sur la base d'un tel site, créez votre propre page web, utilisant des tables pour la mise en page. Ce site devra comporter plusieurs pages accessibles par des liens.

## Introduction aux feuilles de style

HTML est à la fois un langage permettant de décrire la structure d'un texte, mais aussi de spécifier son apparence. Ce sont deux aspects différents de la création de pages HTML et la version 4 de ce langage en prend compte en introduisant l'utilisation de feuilles de style.

Une feuille de style spécifie à part l'aspect visuel (et sonore) d'une page. Le créateur de la page n'utilise alors plus HTML que pour spécifier la structure logique du texte (titres, paragraphes, etc.). Les feuilles de style utilisées par HTML suivent la norme CSS.

Nous avons vu précédemment comment créer une mise en page avec des tables HTML. CSS permet de s'affranchir en grande partie de ces astuces de formatage. Mais CSS permet aussi de produire un grand nombre d'effets qu'il était quasiment impossible de faire auparavant.

## Créer une feuille de style dans la page HTML

Il existe deux moyens d'ajouter du style dans une page HTML:

1. en plaçant les commandes CSS directement dans la page HTML,
2. en créant un fichier séparé que la page HTML référence.

La première méthode est plus simple, mais son problème majeur est que pour réutiliser le style il faut le copier à chaque fois dans toutes les pages HTML. Si on crée un site de plusieurs pages et que l'on change une partie du style, il faut alors mettre à jour toutes les pages HTML.

Voici un exemple complet avec la première méthode qui change la couleur des titres:



```

<html>
<head>
    <style type="text/css">
        h1 { color: red; }
        h2 { color: blue; }
        h3 { color: green; }
        h4 { color: purple; }
    </style>
</head>
<body>
    <h1>Titre de niveau 1</h1>
    <p>Paragraphe 1.</p>
    <h2>Titre de niveau 2</h2>
    <p>Paragraphe 2.</p>
    <h3>Titre de niveau 3</h3>
    <p>Paragraphe 3.</p>
    <h4>Titre de niveau 4</h4>
    <p>Paragraphe 4.</p>
</body>
</html>

```

Une feuille de style est placée dans la balise `head` grâce à la balise `style`. L'attribut `type="text/css"` est obligatoire.

La syntaxe de CSS est très différente de celle d'HTML. On place d'abord le nom de la balise HTML dont on veut modifier le style, puis on le fait suivre d'un bloc entre accolades. Ce bloc permet de spécifier les *attributs de style*. Ces attributs sont définis par un nom, suivi de deux points, puis de la valeur donnée à cet attribut suivie d'un point virgule:

```
baliseHTML { attribut: valeur; }
```

Il est possible de spécifier plusieurs attributs à la fois:

```
baliseHTML { attribut1: valeur1; attribut2: valeur2; }
```

Nous avons vu un premier attribut de style nommé `color` qui permet de changer la couleur du texte. Les valeurs qu'il peut prendre sont toutes les couleurs.

## Créer une feuille de style externe

Lorsque l'on veut partager le style entre plusieurs pages HTML il devient plus pratique de créer un fichier de style séparé. L'exemple suivant montre deux fichiers.

Le fichier `page.html` contient la structure du texte:

```

<html>
<head>
    <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
    <h1>Titre</h1>
    <p>Paragraphe...</p>
</body>
</html>

```

Le fichier `style.css` contient le style de la page HTML:

```
body { background-color: black; color: white; }
h1   { color: red; }
```

Cela donne une page où le texte est blanc sur fond noir et les titres de niveau 1 sont rouges

## Style local à une balise

Les feuilles de style que nous avons vu changent le style d'un type de balise pour toute la page HTML. Par exemple si on change le style de la balise `h1`, tous les titres de niveau 1 seront affichés dans ce style.

Il est possible d'assigner un style particulier à une balise HTML donnée dans la page elle-même. Par exemple, supposons que nous voulions que seul le premier titre en `h1` soit en rouge alors que les autres sont en bleu, nous pourrions écrire:

```
<html>
<head>
  <style type="text/css">
    body { background-color: black; color: white; }
    h1   { color: blue; }
  </style>
</head>
<body>
  <h1 style="color:red;">Premier titre en rouge</h1>
  <p>Paragraphe 1...</p>
  <h1>Autre titre en bleu</h1>
  <p>Paragraphe 2...</p>
  <h1>Autre titre en bleu</h1>
  <p>Paragraphe 3...</p>
  <p>etc.</p>
</body>
</html>
```

L'attribut `style` peut être placé à l'intérieur de n'importe quelle balise HTML et en modifie *localement* le style. Sa valeur est la même que le contenu des accolades dans une feuille de style.

## La mise en page

CSS définit un ensemble très important d'attributs de style. Voici les principales catégories.

### **Polices**

L'attribut CSS `font-family` permet de changer la police, par exemple,

```
<span style="font-family: fantasy;">fantasy</span>
```

donne:

*fantasy*

La taille de la police peut aussi être changée avec l'attribut `font-size`, par exemple:

```
<span style="font-size: 64;">Grand</span>
```

donne:

# Grand

Pour placer du texte en gras on utilise `font-weight: bold;` et pour le placer en italique `font-style: italic;`, ainsi:

```
<span style="font-weight: bold;
font-style: italic;">Gras & Italique</span>
```

donne:

***Gras & Italique***

## **Formatage de texte**

On peut modifier l'espacement des caractères ou l'espacement des mots avec `letter-spacing` et `word-spacing` ainsi:

```
<span style="letter-spacing: 10px;
font-weight: bold;">Espacé</span>
```

donne:

**E s p a c é**

De même il est possible de *décorer* le texte par exemple en le soulignant grâce à `text-decoration` par exemple:

```
<span style="text-decoration: underline;">Souligné</span>
```

donne:

Souligné

## **Formatage de paragraphes**

Il est possible d'ajouter une indentation sur la première ligne d'un paragraphe grâce à `text-indent`, de spécifier la distance entre les lignes grâce à `line-height`, ainsi que définir l'alignement du texte par rapport aux marges avec `text-align`.

Les marges sont définies par l'attribut `margin` qui prend en argument quatre valeurs séparées par des espaces, dans l'ordre: haut, gauche, bas, droit. Par exemple: `margin: 10 20 30 40;` place un espace de 10 pixels en haut, 20 à gauche, 30 en bas et 40 à droite.

Voici un paragraphe formaté avec une indentation de 20 pixels, une hauteur de ligne de 140% (40% de plus que la normale) et justifié (aligné le long des deux marges). Les marges sont de 20% de la page à gauche et à droite.

```
<p style="text-indent: 20px;
line-height: 140%;
text-align: justify;
margin: 0 20% 0 20%;"> La tragédie... <p>
```

**Exercice 13 :** Créez un document contenant le texte suivant, avec le style défini plus haut.

*La tragédie que nous appelons "classique" apparaît vers 1630. Une première génération, celle de Corneille, est bientôt supplantée par celle de Racine. C'est un genre extrêmement codifié : une pièce de théâtre en cinq actes et en vers, alexandrins à rime plate.*

Les marges constituent ce qu'il y a autour d'un bloc de texte, mais il est aussi possible d'ajouter un espace entre le bord du bloc de texte et le texte lui-même grâce à l'attribut `padding`. Cela ne prend usuellement un sens que lorsque le fond d'un bloc de texte ressort. L'attribut `padding` fonctionne exactement comme l'attribut `margin`.

Par exemple:

```
<p style="margin: 0 20% 0 20%
padding: 5% 5% 5% 5%;
background-color: yellow;
text-align: justify;">Mais on n'échappe...<p>
```

donne:

Mais on n'échappe pas ainsi à la tragédie, la mort d'Hippolyte vient briser ce rêve, et vient rappeler qu'il n'est pas d'issue. D'autre part, le sentiment qui unit les jeunes gens vient contraster fortement avec la passion qu'éprouve Phèdre.

Les marges sont invisibles autour du bloc jaune, le *padding* est entre le texte et la bordure, à l'intérieur du bloc jaune.

Il est aussi possible d'ajouter une bordure avec `border-width`, `border-style` et `border-color`. Par exemple:

```
<p style="margin: 0 20% 0 20%;
padding: 5% 5% 5% 5%;
background-color: yellow;
text-align: justify;
border-style: solid;
border-width: 3px;
border-color: red;">La douceur...<p>
```

donne:

La douceur s'oppose à la violence, et l'harmonie à la destruction. Ainsi, l'intrigue secondaire sert l'intrigue principale de deux manières : d'abord, d'un point de vue dramatique, puisque Phèdre apprenant les sentiments d'Hippolyte pour Aricie renonce à détromper Thésée, ensuite sur le plan symbolique, puisque l'amour tendre met davantage en valeur la démesure de l'amour-passion.

**Exercice 14 :** Créez une page HTML et une feuille de style séparée utilisant chacun des attributs vus précédemment.

## Publication d'un site web

**Exercice 15 :** Créez en une page un *journal en ligne*. Vous pouvez vous inspirer de journaux existants tels que [www.lemonde.fr](http://www.lemonde.fr) Votre page devra contenir un titre, des bandeaux, des résumés d'articles, etc.

**Exercice 16 :** Le but est de créer votre propre page Web en utilisant tous les principes vus précédemment, y compris les feuilles de style. Le sujet est libre, mais le site créé doit se composer de plusieurs pages, être navigable à l'aide liens et contenir des images.

## Formulaires

**Exercice 17 :** Réalisation d'un formulaire.

Créer un formulaire possédant les caractéristiques suivantes :

- id = « Form1 »
- method = « GET »
- action = « affiche1.php »

Réaliser plusieurs champs dans ce formulaire :

- trois de type « text » (zones textuelles) : « nom », « prenom » et « age »
- un de type « checkbox » : « accord »
- deux (liés) de type « radio » : le nom sera « role » et la valeur « prof » ou « eleve »
- un bouton de type « submit »

En plaçant au même endroit le fichier affiche1.php fourni, il est possible de voir les réponses.

***VOTRE SITE SERA NOTE A LA FIN DU SEMESTRE !***

La note prendra en compte la présentation, l'ergonomie, les fonctionnalités et le contenu du site.

Vous devez obligatoirement utiliser des feuilles de style pour la mise en page.

Vous pouvez également utiliser PHP comme nous le verrons dans les cours à venir.