

# **M1106**

## **Cahier de TD/TP**

- TP1 : le texte et sa présentation
- TP2 : les formulaires
- TP2 (suite) : les liens et els images
- TP3 : CSS
- TP4 : HTML5 et CSS3

# Le texte et sa présentation

---

## Objectifs

- Créer une page HTML simple,
- Gérer la présentation (titre, paragraphe...),
- Le formatage du texte.

## Exercice 1 – Texte simple

Le texte d'une page Web s'encode tout simplement dans la partie appelée le corps du document, soit entre les balises <body> et </body>.

A l'aide de l'éditeur de code Web Brackets (<http://www.brackets.io/>), saisissez les lignes suivantes :

```
1 <html>
2 <head><title>oh ! ma page...</title></head>
3 <body>
4 Hello World !
5 </body>
6 </html>
```

Après avoir enregistré le document sous une extension **.htm** ou **.html**, ouvrez la page (en local) dans le navigateur Firefox.

## Exercice 2 – Les titres

Il est parfois utile de prévoir différentes tailles de caractères pour les titres du contenu textuel (par exemple, les noms de chapitre).

Le langage Html propose jusqu'à 6 grandeurs de caractères pour les titres. Quand on sait que titre en anglais se dit « **heading** », l'élaboration de la balise devient évidente ; <h pour heading suivi du numéro du niveau 1, 2, 3, 4, 5, 6>. La notation <hx> sera utilisée, où x est un niveau de 1 à 6.

```
1 <html>
2 <head><title>oh ! ma page...</title></head>
3 <body>
4 <h1>Hello World !</h1>
5 <h2>Hello World !</h2>
6 <h3>Hello World !</h3>
7 <h4>Hello World !</h4>
8 <h5>Hello World !</h5>
9 <h6>Hello World !</h6>
10 </body>
11 </html>
```

Malgré la simplicité de cette balise, plusieurs remarques s'imposent :

- Par défaut, avec la balise <hx> le texte est mis en gras.
- Par défaut également, la balise ajoute une ligne vide entre le texte de la balise et la suite du document.
- Pour obtenir des caractères plus grands que ceux proposés par le niveau 6, il faut utiliser les feuilles de styles ou construire des images.
- Les balises de titre sont des éléments importants pour structurer de façon efficace le contenu textuel de la page.
- L'usage des balises de titres doit être réservé à leur fonction première, c'est-à-dire de déterminer des titres de niveaux différents. Il importe de ne pas les détourner de leur but premier pour introduire, par exemple, une mise en forme (gras) ou une taille de caractères déterminée.

## Exercice 3 – Les paragraphes de texte et le retour à la ligne

Pour délimiter un paragraphe, la balise est `<p> ... </p>`.

```
1  <html>
2  <head><title>oh ! ma page...</title></head>
3  <body>
4  <h1>Hello World !</h1>
5  <h2>Hello World !</h2>
6  <h3>Hello World !</h3>
7  <h4>Hello World !</h4>
8  <h5>Hello World !</h5>
9  <h6>Hello World !</h6>
10     <p>premier paragraphe</p>
11     <p>second paragraphe</p>
12 </body>
13 </html>
```

### Remarques

- Par défaut, la balise `<p>` ajoute une ligne vide entre le texte de la balise et la suite du document.
- On a, pendant des années, présenté la balise de fermeture de la balise de paragraphe comme facultative. Avec les évolutions récentes des règles du HTML réclamant plus de rigueur dans la conception du code, il est impératif de respecter scrupuleusement la fermeture des balises.
- Avec les balises de titre, la division en paragraphes contribue à la bonne structuration du contenu.

Dans un document, il est parfois nécessaire de passer à la ligne sans pour cela créer un paragraphe et la ligne vide qu'il ajoute : la balise `<br>` est prévue à cet effet.

```
9  <h6>Hello World !</h6>
10     <p>premier paragraphe</p>
11     <p>second paragraphe</p>
12  Vive le HTML<br>
13     et le<br>
14     XHTML
15 </body>
```

### Remarques

- La balise `<br>` appartient au groupe des balises pour lesquelles il n'y a pas de balises de fermeture.
- Ces balises sans balise de fermeture sont appelées des balises vides ou balises uniques.
- La balise `<br>` remplace les passages à la ligne de votre éditeur de texte.

## Exercice 4 – Le formatage du texte

Il est possible de faire ressortir le texte en le mettant en gras ou en italique. Pour mettre du texte en gras, on peut utiliser la balise `<b> ... </b>` (b pour bold) et le texte en italique, on utilise la balise `<i>... </i>` (i pour italic).

```
<h6>Hello World !</h6>
    <b><p>premier paragraphe</p></b>
    <i><p>second paragraphe</p></i>
Vive le HTML<br>
    et le<br>
    XHTML
```

### Remarque

- Il est possible d'appliquer plusieurs balises à un élément de la page. Dans ce cas, veuillez à respecter scrupuleusement l'ordre inverse pour les balises de fermeture. On parlera par la suite de balises correctement imbriquées.

## Exercice 5 – L'alignement

L'alignement du texte, ainsi que de tout élément du document comme par exemple les tableaux ou les images, est réalisé par l'attribut **align**="type" où "type" peut prendre la valeur :

- **left** pour un alignement à gauche (par défaut),
- **center** pour un alignement centré,
- **right** pour un alignement à droite,
- **justify** pour un alignement justifié.

Le terme d'attribut est un élément de code qui vient s'ajouter à une balise.

```
4 <h1>Hello World !</h1>
5 <h2 align="left">Hello World !</h2>
6 <h3 align="right">Hello World !</h3>
7 <h4 align="center">Hello World !</h4>
8 <h5 align="justify">Hello World !</h5>
9 <h6>Hello World !</h6>
```

## Exercice 6– La taille, la couleur et la police

### La taille

Pour modifier la taille des caractères, on peut utiliser la balise **<font size="x"> ... </font>** où x peut prendre, au choix, les valeurs de 1 à 7 ou de -3, -2, -1, 0, +1, +2, +3.

```
1 <html>
2 <head><title>Des exemples</title></head>
3 <body>
4   <font size=8>Hello World !</font><br>
5   <font size="+3">Hello World !</font>
6 </body>
7 </html>|
```

Le HTML 4.0 préconise l'usage des feuilles de style CSS pour la taille des caractères. Les possibilités offertes par ces dernières sont nettement plus variées que celles proposées par le HTML.

### La couleur

Chaque couleur est définie par une proportion de trois couleurs de base. C'est le système appelé **RVB** : Rouge, Vert et Bleu (RGB en anglais). Ce système est utilisé dans de nombreuses applications graphiques. Le rouge aurait pu être défini par exemple avec les repères 255,0,0 comme en RVB.

En HTML, au lieu d'exprimer ces chiffres de 0 à 255, on a préféré les exprimer en valeur hexadécimale soit de 00 à FF (soit deux positions au lieu de trois). Ce qui donne pour le rouge la valeur FF0000. Comme cette base est peu utilisée, cette notation peut paraître assez sibylline. Elle présente quand même l'avantage de pouvoir exprimer en 6 chiffres quelques 16,7 millions de couleurs différentes.

Par la suite, le W3C a introduit une liste de 16 noms de couleur qui peuvent également être utilisés soit *black, maroon, green, olive, navy, purple, teal, gray, silver, red, lime, yellow, blue, fuchsia, aqua et white*. Les firmes comme Microsoft et Netscape ont étendu cette liste mais certains noms de couleur ne sont pas communs aux deux firmes et l'usage du nom de couleurs (hors les 16 couleurs officielles) n'est pas dénué de risques. Les concepteurs de sites professionnels restent d'ailleurs fidèles à la notation hexadécimale. Un convertisseur hexadécimal en ligne : <http://www.proftnj.com/RGB3.htm>

Pour changer la couleur d'un texte :

- **<font color="#\$\$\$\$\$"> ... </font>** pour la notation hexadécimale ou
- **<font color="nom"> ... </font>** pour la notation par nom

```
4      <font size=8>Hello World !</font><br>
5      <font size="+3">Hello World !</font><br>
6      <font color="#FF0000">Rouge</font><br>
7      <font color="navy" size=6>Marine</font>
```

#### Remarques

- La couleur par défaut du texte de la page est le noir.
- Cette couleur par défaut peut être modifiée par l'attribut `text="#$$$$$"` appliqué à la balise `<body>`.

## La police de caractères

Il est possible de changer la police définie par défaut dans le navigateur de l'utilisateur, généralement Times New Roman, par la balise `<font face="nom de la police"> ... </font>`

Si la police de caractères n'est pas installée sur l'ordinateur de votre visiteur, son navigateur applique la police par défaut. Il est ainsi inutile d'utiliser des polices de caractères exotiques, en conception de pages Web, qui n'existeront peut-être pas sur l'ordinateur de votre visiteur du bout du Web. Votre bel effet de présentation sera ainsi réduit à néant.

Il est même recommandé par précaution de proposer deux ou trois polices. Le navigateur regarde alors s'il peut afficher le texte dans la première police spécifiée, sinon il passe à la deuxième et ensuite à la troisième. Ce n'est que lorsqu'il a épuisé les deux ou trois propositions qu'il retient la police définie par défaut.

La balise se note alors, par exemple `<font face="Verdana, Arial, Helvetica">`

```
7      <font color="navy" size=6>Marine</font>
8      <h2><font face="arial">Bonjour</font></h2>
9      <h2><font face="verdana">Bonjour</font></h2>
```

## Exercice 7– Les caractères spéciaux

Les anglo-saxons se retrouvent parfaitement à l'aise avec l'ASCII standard utilisé pour les documents HTML. En effet, comme ils n'utilisent pas les accents, les 128 caractères de l'ASCII leur suffisent largement. Heureusement pour nous européens, dont beaucoup de langues sont d'essence latine, le HTML reconnaît la **norme ISO 8859-1** dite de l'alphabet latin-1.

#### Exemples

- le é s'encode avec `&eacute;`;
- le è s'encode avec `&eagrave;`;
- le ê s'encode avec `&ecirc;`;
- le à s'encode avec `&agrave;`;
- le € s'encode avec `&euro;`;
- l'espace insécable avec `&nbsp;`;

```
10      Les étudiants rêvassent... <br>
11      Les &eacute;tudiants r&ecirc;vassent...
```

La liste est ici <http://lorem-ipsum.studiovitamine.com/caracteres-speciaux,287,fr.html>

Référence des entités de caractères dans HTML 4 du W3C <http://www.la-grange.net/w3c/html4.01/sgml/entities.html>

La syntaxe exacte comporte bien le point-virgule final. Même si Internet Explorer affiche correctement le caractère sans le point-virgule final, il n'en est pas de même avec d'autres navigateurs comme Firefox.

## Exercice 8– Autres balises de texte

Les balises `<strike>` et `<u>` permettent de tracer une ligne au travers ou en dessous d'un texte.

Si vous encodez des codes informatiques ou d'autres caractères que vous désirez différencier du texte, vous pouvez les formater au moyen d'une police à pas fixe de style courrier. Bien que le résultat visuel soit identique, voici quelques balises :

- `<code> ... </code>` signale du code informatique.
- `<kbd> ... </kbd>` signale une saisie clavier.
- `<samp> ... </samp>` signale un exemple.
- `<tt> ... </tt>` signale du texte de télétype.

Pour mettre des chiffres ou du texte en exposant ou un indice, on dispose en HTML de la balise `<sup> ... </sup>` (exposant) et de `<sub> ... </sub>` (indice).

La balise `<pre> ... </pre>` affiche le texte préformaté (en police à pas fixe) soit avec les espaces, les retraits, les sauts de ligne tels qu'ils apparaissent à l'écran dans l'éditeur de texte.

La balise `<blockquote> ... </blockquote>` a été conçue pour introduire une citation. Le texte de celle-ci s'affiche avec un léger retrait par rapport au texte normal.

```

1 <html>
2 <head><title>Autres balises...</title></head>
3 <body text="#2ABE81">
4 Le <strike>CSS2</strike> CSS3 est...<br>
5 texte normal<br>
6 <code>code informatique</code><br>
7 <kbd>saisie clavier</kbd><br>
8 <samp>exemple</samp><br>
9 <tt>Exemple de teletype</tt><br>
10 y = ax<sup>2</sup> + bx + c<br>
11 H<sub>2</sub>O<br>
12 <pre>
13 1      2      3
14 4      5      6
15 7      8      9
16 </pre><br>
17 <blockquote>Microsoft n'est pas le diable, il fait simplement de tr&egrave;s mauvais syst&egrave;mes d'exploitation.</blockquote>
18 Linus Torvald
19 </body>
20 </html>

```

## Exercice 9– Les listes

### Les listes numérotées

Une liste numérotée est délimitée par les balises `<ol>` et `</ol>` entre lesquelles chaque élément de la liste est précédé de la balise `<li>`.

La balise `<ol>` comporte des attributs :

- `type="x"` où x correspond à :
  - A pour des majuscules,
  - a pour des minuscules,
  - I pour des chiffres romains majuscules,
  - i pour des chiffres romains minuscules,
  - 1 (par défaut) pour des nombres.
- `start="x"` où x représente la valeur (en nombre) de début de la numérotation.

#### Remarques

- La balise `<ol>` comme pour les autres listes introduit un léger retrait du texte.
- N'oubliez pas la balise de fermeture de la liste `</ol>` car dans ce cas le retrait continue à être appliqué par le navigateur.
- Il y a quelques années, la balise de fermeture de liste `</li>` était présentée comme facultative. Avec la rigueur réclamée par le W3C dans l'écriture du code Html, cette balise de fermeture est bien obligatoire.

## Les listes à puces

Une liste non numérotée est définie de la même façon, sauf qu'elle est encadrée par les balises <UL> et </UL>.

La balise <ul> comporte un attribut type="forme" où forme correspond au type de puces. En HTML, vous avez le choix entre trois formes de puces :

- **disc** pour une puce ronde et pleine,
- **circle** pour une puce ronde et vide
- **square** pour une puce carrée.

```
19      Exemple de liste &agrave; puces
20      <ul type="circle">
21      <li>Le texte simple</li>
22      <li>Les titres</li>
23      <li>Les paragraphes</li>
24      </ul>
```

### Remarques

- La balise <ul> comme pour les autres listes introduit un léger retrait du texte.
- N'oubliez pas la balise de fermeture de la liste </ul>, sinon le retrait continue à être appliqué par le navigateur.
- La balise de fermeture </li> est bien obligatoire.

## Exercice 10– Les séparateurs horizontaux

Le trait horizontal est un outil fort pratique pour structurer le contenu. Le Html incorpore un élément graphique pour le réaliser. C'est la balise <hr>.

Cette balise comporte quelques attributs :

- |                                       |  |
|---------------------------------------|--|
| ▪ size="x"                            | détermine la hauteur du trait en pixels.                   |
| ▪ width="x" ou width="x%"             | détermine la largeur du trait en pixels ou en pourcentage. |
| ▪ align="left" ou "center" ou "right" | pour l'alignement du trait.                                |
| ▪ Noshade                             | pour créer un trait plein sans effet d'ombrage.            |
| ▪ color="couleur"                     | détermine la couleur.                                      |

```
5      <body>
6      Chapitre 1
7      <hr size="5" width="60%" align="center" noshade color="red">
8      Chapitre 2
9      </body>
```

### Remarques

- La balise <hr> ne comporte pas de balise de fermeture.
- Les concepteurs préfèrent remplacer ce trait, assez simpliste, par une image, graphiquement plus élaborée.
- L'attribut **color** utilisé dans l'exemple, ne fait pas partie de la norme HTML 4.0.

## Exercice 11– Synthèse

Réaliser la page HTML ci-dessous en respectant les consignes suivantes :

- Titre de la page : Une galerie photo avec Piwigo !
- Titre balise H1, texte centré
- Introduction, couleur 2BA696, citation,
- Les sous-titres, balise H2, couleur D02A00,
- Paragraphe présentation générale : taille 4, texte justifié et en italique,
- ...

## Une galerie photo avec Piwigo

*Piwigo est une application de publication de photos pour le web. A la fois élégante, performante et riche en fonctionnalités, Piwigo reste simple d'utilisation. Parmi les possibilités offertes, vous trouverez l'organisation par catégories sur vos photos.*

### Présentation générale

Les photos sont classées par catégories, au sein desquelles il est possible de les trier automatiquement ou manuellement par glisser/déposés telles que le titre, la description, l'auteur, la date et les tags. Piwigo peut exploiter les métadonnées EXIF/IPTC pour renseigner ces propriétés.

### Piwigo.com : Piwigo en solution hébergée

Lancé en 2010 par le fondateur de Piwigo, le site Piwigo.com est une offre d'hébergement dédié à Piwigo. Piwigo.com propose une unique offre à 39 euros par an, laquelle comprend :

- une **galerie** Piwigo préinstallée ;
- un **espace** permettant le stockage d'un nombre illimité de photos ;
- un **support** dédié ;
- l'automatisation des mises à jour sur la dernière version du logiciel ;
- un système de sauvegardes automatiques ;
- l'absence de publicité ;
- la confidentialité des données personnelles et l'absence d'exploitation commerciale de celles-ci.

## Exercice 12– Les tableaux

Un tableau (<table>) comporte des lignes (<tr>). Ces lignes comprennent plusieurs cellules (<td>). Soit le tableau suivant :

1	2
3	4

Dans la logique du Html, le tableau se construit ainsi :

- On déclare que l'on ouvre un tableau avec la balise <table>.
- Ce tableau comporte une première ligne soit <tr>.
- Cette ligne comporte une première cellule <td>.
- Soit 1 comme contenu de la cellule.
- Fermeture de cette première cellule </td>.
- Vient une seconde cellule <td>2</td>.
- Fin de la première ligne </tr>.
- On passe à la ligne suivante soit à nouveau <tr>.
- Cette ligne comporte les cellules <td>3</td> et <td>4</td>.
- Fin de la seconde ligne </tr>.
- Et fin du tableau avec la balise de fermeture </table>.

La balise <table> possède de nombreux attributs pour améliorer la présentation des tableaux :

- ajouter une bordure par l'attribut **border= "x"** où x est un nombre exprimé en pixels.
- donner une couleur à la bordure par l'attribut **bordercolor=couleur**
- aligner le tableau à gauche (par défaut), au centre et à droite par l'attribut **align="left"** ou **"center"** ou **"right"**.
- gérer l'espace entre les cellules du tableau (par défaut à 2 pixels). L'attribut **cellspacing="x"** permet de modifier cet espacement entre les cellules.
- gérer la marge à l'intérieur des cellules (espace qui sépare les bords de la cellule et le contenu de celle-ci, par défaut 1 pixel). L'attribut **cellpadding="x"** permet de modifier cet espacement entre les bords de la cellule.
- déterminer la largeur du tableau par l'attribut **width="x"** ou **"x%"** où x est exprimé en pixels et x% correspond à un pourcentage de la fenêtre.



## TP1 – Le texte et sa présentation

```
1  <html>
2  <body>
3  <table width="40%" align="center" cellspacing="20" cellpadding="10" border="2">
4  <tr>
5  <td>1</td><td>2</td>
6  </tr>
7  <tr>
8  <td>3</td><td>4</td>
9  </tr>
10 </table>
11 </body>
12
13 </html>|
```

# Les formulaires

---

## Objectifs

- Mettre en place un formulaire,
- Créer des zones de texte, combobox, boutons...
- Ajouter des boutons d'envoi.

Avec les formulaires, le HTML vous ouvre les portes de l'interactivité et vous permet de recevoir des informations provenant directement de l'utilisateur et structurées selon vos souhaits.

## Exercice 1 – Création d'un formulaire simple

```
<form>
... les éléments du formulaire ...
</form>
```

Cette balise comporte des attributs :

### L'attribut name

Il est parfois utile, pour un traitement ultérieur de ces formulaires, de leur prévoir un nom surtout si la page en comporte plusieurs. Utilisez l'attribut **name="nom"**.

### L'attribut method

Les données d'un formulaire seront souvent transmises à un serveur. Cette transmission peut s'effectuer selon la méthode GET ou la méthode POST. La différence entre ces deux méthodes repose sur la façon dont les données seront transmises au serveur et exploitées par celui-ci. Avec le temps, la méthode **POST** s'est imposée car elle apparaît plus efficace et permet le traitement d'une quantité plus importante de données.

### L'attribut action

Lorsque vous donnez l'ordre au navigateur de transmettre vos données, il a besoin de savoir ce qu'il doit faire (action). En fait, il faut qu'il connaisse l'adresse de l'agent ou du "programme" à qui il doit transmettre les données du formulaire.

- soit une adresse d'un script de traitement de ces données en CGI, Perl, PHP,... situé sur un serveur, par exemple, `action="http://www.serveur.net/cgi-bin/ma_cgi.pl"`.
- soit une adresse électronique pour récupérer simplement les données. On utilise alors le protocole mailto: suivi de l'adresse électronique de destinataire (généralement votre adresse e-mail, par exemple, `action="mailto:mon_email@serveur.net"`).

L'attribut action est obligatoire en Html 4.0. Afin de reproduire un code correct, nous utiliserons le subterfuge `action=""`.

### L'attribut enctype

L'attribut **enctype** va spécifier sous quel format (informatique) seront transmises les données du formulaire. Par défaut, le format utilisé est le format **application/x-www-form-urlencoded**, dont la forme est difficilement lisible. Pour se faciliter la lecture, surtout pour un envoi vers une adresse électronique avec mailto, on utilise alors **enctype="text/plain"**. Cet attribut ne peut être utilisé qu'accompagné par la méthode POST.

Simple mais fort utile, la zone de texte à une seule ligne permet de recevoir l'encodage de toutes sortes de données (lettres ou chiffres) comme le nom, l'adresse ou le code postal.

```
<body>
<form action="">
<input type="text" name="nom" size="40" maxlength="35" value="saisir votre nom ici">
</form>
</body>
```

### La zone de texte à une seule ligne

- L'attribut **name**  
Un nom d'identification peut se révéler nécessaire pour le traitement de votre formulaire par un script ou une application. L'attribut prend la forme de **name="nom"**.
- L'attribut **size**  
L'attribut **size="x"** où x est la largeur de la zone de texte, définit la taille visible de la ligne de texte. La valeur par défaut de size est de 20.
- L'attribut **maxlength**  
L'attribut **maxlength="x"** définit le nombre maximal de caractères que l'utilisateur peut encoder dans la zone de texte. Cet attribut est particulièrement utile pour les données de longueur définies comme par exemple 5 chiffres pour un code postal français.
- L'attribut **value**  
Il est possible de donner une valeur par défaut à la zone de texte par l'attribut **value="texte"** par exemple `<input type="text" value="Votre nom ici !">`.

```
7 <body>
8 <form action="">
9 Nom : <input type="text" value="votre nom" name="nom" size="40" maxlength="35">
10 </form>
11 </body>
```

### La zone de texte à plusieurs lignes

Dans certains cas, il faut prévoir plus d'espace à l'utilisateur pour encoder par exemple ses remarques et suggestions. On utilise dans ce cas la balise **<textarea>...</textarea>**.

1. Créer une zone de texte de 4 lignes et 25 colonnes.
2. Quels sont les attributs de cette balise ? (<http://www.w3.org/TR/html4/interact/forms.html#edef-TEXTAREA>)

### Le menu déroulant

Le menu déroulant ou la liste de choix est un autre élément de formulaire.

```
7 <body>
8 <form action="">
9 <select>
10 <option>moins de 20 ans</option>
11 <option>de 20 à 40 ans</option>
12 <option>de 40 à 60 ans</option>
13 <option>plus de 60 ans</option>
14 </select>
15 </form>
16 </body>
17 </html>
```

Pour les attributs, voir <http://www.w3.org/TR/html4/interact/forms.html#edef-OPTION>

### Les boutons de choix unique (bouton radio)

Les boutons d'option, aussi appelés boutons radio, ont comme particularité qu'une seule option à la fois peut être activée (le "ou" exclusif).

```
<body>
<form action="">
Sexe :<br>
<input type="radio" name="sexe">masculin<br>
<input type="radio" name="sexe">féminin
</form>
</body>
```

Se référer à la documentation officielle <http://www.w3.org/TR/html4/interact/forms.html#radio>

### Les boutons de choix multiples (checkbox)

La mise en œuvre de ces boutons de choix multiples est proche des boutons à choix unique mais à la différence que, dans le cas présent, plusieurs choix simultanés peuvent être réalisés.

```
<body>
<form action="">
Votre glace vanille avec :<br>
<input type="checkbox" name="nom0"> chocolat<br>
<input type="checkbox" name="nom1"> chantilly<br>
<input type="checkbox" name="nom2"> biscuit<br>
</form>
</body>
```

Se référer à la documentation officielle <http://www.w3.org/TR/html4/interact/forms.html#checkbox>

### Le bouton d'envoi

Par le bouton d'envoi, appelé parfois aussi bouton de soumission, les données du formulaire seront transmises selon l'attribut action défini dans la balise <form>. Il est donc indispensable pour l'utilité de votre formulaire.

```
<body>
<form action="">
<input type="submit">
</form>
</body>
```

### Le bouton d'annulation

Il est opportun de permettre à l'utilisateur d'annuler son encodage et de réinitialiser (reset) un formulaire vierge.

```
<body>
<form action="">
<input type="reset">
</form>
</body>
```

### Le bouton de commande

Il est également possible de créer des boutons dont l'action spécifique sera définie par le webmestre, généralement en faisant appel à du JavaScript, Php...

## TP2 – Les formulaires

On définit par exemple, une balise **<button> de type="button"** (car il y a aussi le type submit et reset). L'action du bouton est déclenchée par un événement JavaScript. Cet événement est ici onClick, soit au clic de la souris qui entraînera l'ouverture de la page dont la localisation est le fichier "suite.htm".

```
<body>
<form action="">
<button type="button" onClick="location.href='suite.htm'">Suite
</button>
</form>
</body>
```

Se référer à la documentation officielle <http://www.w3.org/TR/html4/interact/forms.html#push-button>

### Les formulaires de transfert de fichier

Les formulaires de transfert de fichier (file) permettent à l'utilisateur de transmettre avec un formulaire, un fichier de son ordinateur local vers un ordinateur de type serveur.

```
<body>
<form action="..." method="post" enctype="multipart/form-data">
<input type="file">
</form>
</body>
```

Dans la déclaration de la balise <form>, il faut impérativement utiliser la méthode post et enctype="multipart/form-data" pour le transfert du fichier au bon format.

### Les formulaires de mot de passe

Ce type de formulaire est en fait une simple zone de texte mais dont l'encodage est remplacé, à l'affichage, par des puces ou des astérisques. Ce formulaire de mot de passe ne protège en aucun cas les données, car elles seront **transmises en clair**. Elles vous protègent uniquement contre les personnes qui pourraient vous lire durant l'encodage (et encore !).

```
<body>
<form action="">
Votre nom :
<input type="password">
</form>
</body>
```

### L'organisation des éléments d'un formulaire

Dans le cas de formulaires longs et complexes, il est parfois utile de regrouper graphiquement certains éléments pour organiser la page et en améliorer la lisibilité.

La balise **<fieldset> ... </fieldset>**, qui se place après la balise <form>, regroupe les éléments du champ que vous déterminez. Le champ sera alors visualisé à l'écran par une bordure.

La balise **<legend>Texte de la légende</legend>** fournit la légende qui vient s'insérer dans le cadre dessiné par <fieldset>. Cette balise admet l'attribut align="left" (défaut) et align="right".

```
<body>
<form action="">
<fieldset>
<legend>Coordonnées</legend>
Nom : <input type="text" size="25"><br>
Prénom : <input type="text" size="25"><br>
E-mail : <input type="text" size="25">
</fieldset>
</form>
```

```
<legend>Suggestions</legend>
<textarea cols="25" rows="3"></textarea>
</fieldset>
</form>
</body>
```

La balise **<label>** associe explicitement l'intitulé à un contrôle de formulaire particulier. Un peu comme si on collait une étiquette (label) en face d'un élément de formulaire.

La balise **<label>** et son contenu n'ont aucun effet visible à l'écran mais elle est particulièrement utile dans le domaine de l'accessibilité des sites Web aux personnes non-voyantes. Cette balise **<label>** est prise en charge par les aides techniques et facilite grandement l'utilisation des formulaires par les personnes atteintes d'un handicap visuel.

Se référer à la documentation officielle <http://www.w3.org/TR/html4/interact/forms.html#edef-LABEL>

### 3. Créer un livre d'or (guestbook)

Votre nom :

Votre prénom :

Votre adresse :

Code Postal :

Ville :

Adresse e-mail :

# Les liens et les images

---

## Objectifs

- Utiliser les fonctionnalités hypertextes,
- Gérer les images,
- Définir l'arrière-plan

## Exercice 1 – Balise de liens

```
<a href="destination">texte du lien</a>
```

La balise de liens permet d'encoder la destination de sorte que le lien pointe :

- vers une autre page du site.
  - vers un endroit de la page en cours.
  - vers un endroit d'une autre page.
  - vers une autre page située sur le Web.
  - vers une adresse de courrier électronique.
  - vers un fichier que le visiteur pourra télécharger.
1. Créer 2 fichiers (page1.html et page2.html) situés dans le même dossier.
  2. Effectuer dans la première page un lien vers la deuxième page.
  3. Créer un lien dans la seconde pour revenir vers la première page.

L'attribut **title="Texte"** de la balise `<a>` permet de proposer une info-bulle explicative au lien.

Exemple : `<a href="page2.htm" title="Lien vers page 2">Lien</a>`

La couleur des liens varie suivant si le lien a déjà été visité ou non. Sauf spécifications de la part du concepteur, les liens non-visités s'affichent en bleu, les liens visités en mauve et les liens actifs au moment du clic par l'utilisateur en rouge.

## Exercice 2 – Les ancres

Il est parfois intéressant de pouvoir mener l'utilisateur vers un endroit précis du document. On utilise alors ce qu'on appelle les ancres. La syntaxe de création d'une ancre est :

```
<a name="nom de l'ancre"></a> ou  
<a name="nom de l'ancre">élément de la page</a>
```

Après avoir créé une ancre, vous pouvez définir un lien vers celle-ci. Pour créer ce lien, il suffit de reprendre la balise de lien en précisant comme destination l'ancre souhaitée par : **#nom de l'ancre**.

```
<a href="#ancre">Partie1</a>
```

1. Ouvrir le fichier ancre.html
2. Placer une ancre au début du texte et une seconde à la fin.

## Exercice 3 – Les images

Les images ont grandement contribué à la popularité de Web, créant ainsi son aspect convivial et attractif qui en fait un univers de création et de communication à part entière.

Pour réduire au maximum le temps de transmission, le Web réclame un format d'image réduit et donc compressé. Avant d'inclure vos images, il est souvent indispensable de les convertir dans un format (compressé) compatible avec le Web, d'en optimiser la dimension et surtout la taille en octets. Cette modification ne peut se réaliser que par des programmes graphiques comme Adobe Photoshop, Paint Shop Pro, Paint.net ou Gimp.

La balise d'insertion d'image est ``.

Le *fichier\_image* est le nom et l'extension de l'image, si elle est située dans le même dossier que le fichier Html. Dans le cas contraire, il faut en outre spécifier le chemin d'accès au *fichier\_image*. En cas d'erreur dans le nom du fichier et/ou dans l'adressage, l'image ne sera pas affichée.

Cette balise image comporte des attributs :

### L'attribut alt

L'attribut **alt="texte de remplacement"** fournit une info-bulle explicative du meilleur effet lors du survol de l'image par la souris.

### Les attributs width et height

Un navigateur n'a beau être qu'une visionneuse de page Web, sa tâche n'est pas pour autant facile. En effet, à partir du code source du document, il doit effectuer sa mise en page pour afficher du texte, des images, des tableaux, etc. En spécifiant la largeur et la hauteur de l'image par les attributs **width** et **height**, il pourra réserver son emplacement et continuer à composer et à afficher le texte avant son chargement complet.

La définition de la taille se fait par les attributs *width="x"* et *height="y"* où x est la largeur en pixels de l'image et y la hauteur en pixels.

### Les attributs hspace et vspace

Il est possible d'ajouter une zone (vide) autour d'une image pour éviter qu'elle ne soit collée au texte ou à tout autre élément de la page.

L'attribut de la balise image *hspace="x"* où x est exprimé en pixels, ajoute horizontalement de l'espace à gauche et à droite de l'image. L'attribut de la balise *vspace="x"*, ajoute verticalement de l'espace en haut et en bas de l'image.

### L'alignement d'une image

La balise image comporte également des attributs spécifiques pour l'alignement de celle-ci par rapport au texte ou aux autres éléments de la page.

L'alignement se réalise par l'attribut *align="type"* où type prend la valeur :

- **top** aligne l'image sur le plus haut élément de la ligne.
- **texttop** aligne l'image sur le plus haut du texte de la ligne.
- **absmiddle** centre l'image sur le milieu du plus grand élément.
- **middle** centre sur la ligne de base.
- **absbottom** aligne le bas de l'image sur le bas du plus grand élément.
- **bottom** aligne le bas de l'image sur le plus bas du texte.
- **baseligne** aligne le bas de l'image sur la ligne de base.

```
1 <html>
2 <head><title>Gestion des images</title></head>
3 <body>
4   Mascotte Linux <br>
5   Mascotte Linux 
6 </body>
7 </html>
```



## Exercice 4 – L'arrière-plan

Pour donner une couleur d'arrière-plan (unie) à la page, l'attribut **bgcolor="#\$\$\$\$\$"** ou **bgcolor="nom"** est ajouté à la balise <body>.

Une image peut être utilisée pour tapisser le fond de la page en mosaïque. Pour cela, à la balise <body> est ajouté l'attribut **background="fichier\_image"** où *fichier\_image* correspond au nom de votre image et éventuellement à l'adressage pour y accéder.

Il est possible d'insérer une image d'arrière-plan à un tableau. On ajoute l'attribut **background="fichier\_image"** à la balise <table>.

```
1 <html>
2 <head><title>Gestion des images</title></head>
3 <body bgcolor="#99CCFF">
4     Mascotte Linux <br>
5     Mascotte Linux 
6 </body>
7 </html>
```

## Exercice 5 – Lien sur une image

Pour insérer un lien, il suffit d'encadrer la balise image avec les balises de lien. Pour supprimer le cadre de couleur que le navigateur ajoute par défaut pour signaler le lien hypertexte, mettre l'attribut **border="0"**.

```
1 <html>
2 <head><title>Gestion des images</title></head>
3 <body bgcolor="#99CCFF">
4     Mascotte Linux <br>
5     Mascotte Linux <br>
6     <a href="page2.html"></a>
7 </body>
8 </html>
```

# Les feuilles de style

## Objectifs

- Comprendre la structure d'une feuille de style,
- Créer des feuilles de style simples,
- Appréhender la notion de cascade.

## Syntaxe d'un élément de style

La définition générale d'un élément de style est la suivante :

`balise { propriété de style: valeur; propriété de style: valeur }`

Ce qui se lit : à la balise déterminée, appliquer la première propriété de style ainsi que la propriété de style suivante.

### Exemple

`h2 {font family: Arial; color: #999999}`

Ce qui se lit : à la balise de titre h2, appliquer la propriété de police de valeur Arial, et la couleur désignée.

- Les feuilles de style s'ajoutent principalement à des balises, par exemple **p**, **body**, mais aussi à des éléments **a:link** pour un lien (non-visité).
- Les propriétés de style sont entourées par des accolades "{".
- Le couple "propriété de style et valeur" est séparé par un double point (:).
- Chaque couple "**propriété de style et valeur**" est séparé par un point-virgule (;).
- Il n'y a pas de limite pour le nombre de couples "**propriétés de style et valeur**".
- L'espace entre propriétés de style et valeur n'est pas obligatoire mais aide fortement à la lisibilité du code source.
- Bien que les styles ne soient pas sensibles aux majuscules et minuscules (case insensitive), l'usage veut que les feuilles de style soient notées en minuscules.
- Pour la lisibilité toujours, les styles peuvent être écrits sur plusieurs lignes :  
`h2 {font family: Arial;  
color: #999999}`
- Plusieurs valeurs peuvent être attribuées à une même propriété. Dans ce cas, les différentes valeurs sont séparées par des virgules.  
`h3 {font-family: Arial, Helvetica, sans-serif}`
- Un même style peut être attribué à plusieurs balises (séparées par des virgules).  
`h1, h2, h3 {font-family: Arial; font-style: italic}`

## Choix des polices de caractères

Il existe 5 familles de polices de caractères dites « génériques » :

- Serif
- Sans-serif
- Monospace
- Cursive
- Fantasy

Il n'y a pas de grosses différences avec les familles. Il vaut mieux donc se contenter des familles **serif** et **sans-serif**, les sans-serif étant les plus lisibles à l'écran.

Les familles de polices se déclarent à l'aide de l'attribut **font-family**, à déclarer dans le **body** pour que toute la page en bénéficie (c'est la notion d'héritage). On finit toujours la liste de fontes déclarées par sa famille générique à laquelle elle appartient. Ainsi, si la police déclarée n'existe pas sur l'ordinateur de votre visiteur, le navigateur affichera la police appartenant à la même famille.

Dans le cas où la police n'est pas disponible sur l'ordinateur de votre visiteur, il y a possibilité de la télécharger.

`<link href='http://fonts.googleapis.com/css?family=Doppio+One' rel='stylesheet' type='text/css'>`

Pour un large choix de polices, voir le site <http://www.google.com/fonts>

## Exercice 1 – Feuille de style interne

Les feuilles de style dites **internes** s'appliquent à tout un document HTML mais à ce seul document. Il est ainsi possible d'incorporer les feuilles de style internes dans l'en-tête, entre les balises <head>.

Effet de style qui enlève le soulignement à tous les liens d'un document HTML. Cet effet s'obtient par le code :

```
1 <html>
2 <head>
3 <style>
4 <style type="text/css">
5 a {text-decoration:none}
6 </style>
7 </head>
8 <body>
9 <a href="test.htm">Ceci est bien un lien</a>
10 </body>
11 </html>
```

## Exercice 2 – Feuille de style externe

Les feuilles de style externes (externes car regroupées dans un fichier séparé), sont idéales pour donner un style commun à toute une série de pages. Plutôt que de devoir mettre dans chaque page les lignes de code du style, il suffit de les encoder une seule fois et elles pourront servir à toutes les pages qui y font référence.

Pour appeler ce document; cela se fait par la balise <link> qui se positionne dans l'en-tête du document HTML <head>.

```
<head>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
```

- La balise **<link>** avertit le navigateur qu'il faudra réaliser un lien.
- L'attribut **rel=stylesheet** précise qu'il y trouvera une feuille de style externe. Remarquez que c'est stylesheet au singulier donc sans s.
- L'attribut **type="text/css"** précise que l'information est du texte et du genre Cascading Style Sheets (css).
- L'attribut classique de lien **href="destination"** donne le chemin d'accès et le nom du fichier à lier. Ici le fichier style.css est supposé être dans le même dossier que les pages Web.

1. Créer la feuille de style nommée **style1.css** :

```
1 body {
2   color: #000000;
3   font-size: 12px;
4   font-family: verdana, sans-serif;
5   background-color: #e8f6e9;
6 }
7
8 p {
9   color: #333333;
10  font-size: 16px;
11  font-weight: bold;
12  text-align: justify;
13 }
```

2. Créer le document HTML **feuille1.html** appelant la feuille de style.

```
1 <html>
2 <head>
3   <title>CSS externe</title>
4   <link rel="stylesheet" type="text/css" href="style1.css">
5 </head>
6 <body>
7   <p>Ceci est un paragraphe</p>
8 </body>
9 </html>
```

## Exercice 3 –Style local

Les styles sont insérés n'importe où dans la page en affectant un attribut style à une balise. À utiliser avec modération cependant, pour modifier quelques mots ou quelques lignes, une portion de texte, un paragraphe ou une cellule de tableau, puisqu'on se détourne ainsi de la véritable finalité du CSS, soit la séparation du contenu et de la mise en forme.

```
1 <html>
2 <head>
3   <title>Titre de la page</title>
4 </head>
5 <body>
6   <p style="color:#ff0000;font-weight:bold">
7     Le texte du paragraphe sera rouge (#ff0000) et gras (bold).
8   </p>
9 </body>
10 </html>
```

## Exercice 4 –Les éléments class et id

Une feuille de style est appliquée à une balise donnée. Les éléments **class** et **id** permettent, eux, de donner des effets de style différents à une même balise.

La définition d'un style devient :

```
balise.nom_de_classe { propriété de style: valeur }
```

Comme la mention de la balise est facultative, on peut aussi indiquer :

```
.nom_de_classe { propriété de style: valeur }
```

Pour appeler l'effet de style dans le document, ajoutez le nom de la classe à la balise :

```
<balise class="nom_de_classe"> .... </balise>.
```

Exemple :

1. Mettre un type de paragraphe en gras et en rouge désigné par le nom « important » et un autre type de paragraphe en italique nommé « remarque ».

```
1 <html>
2 <head>
3   <title>CSS</title>
4   <style type="text/css">
5     .important {font-weight: bold; font-color: red;}
6     .remarque {font-style:italic;}
7   </style>
8 </head>
9 <body>
10  <p class="important">Feuilles de style en cascade</p>
11  <p class="remarque">Cascading Style Sheets</p>
12 </body>
13 </html>
```

2. Ajouter un type de paragraphe souligné et vert désigné par le nom « attention ».

Il existe aussi l'élément **id** dont la fonction est similaire à l'élément class à ceci près. Comme la convention nom/point/nom est utilisée aussi en JavaScript, il a fallu trouver une autre convention d'écriture pour utiliser les feuilles de style avec du JavaScript.

Ce sont les id, aussi appelés les identifiants. Les id fonctionnent exactement comme les class. La syntaxe est :

```
#nom_de_id { propriété de style: valeur }
```

```
<balise id="nom_de_id"> .... </balise>
```

Un seul appel à #nom\_de\_id par page pourra être effectué.

Si vous pensez utiliser des feuilles de style, mais sans utiliser des scripts, oubliez au plus vite id et utilisez exclusivement les éléments class.

## Exercice 5 –Les balises span de div

Ces balises introduisent dans la conception de pages Web ce que l'on pourrait comparer aux zones de texte pour ceux qui utilisent les outils bureautiques ou aux calques pour les infographistes.

Ces zones ou boîtes ainsi déterminées ne prennent de sens que par les effets de style qui leur sont appliquées. Ces zones peuvent s'inclure dans le texte de la page mais aussi en relief par rapport à la page grâce à la propriété **z-index** des feuilles de style.

- La balise **<div>** crée une boîte ou un élément de type bloc. Ces éléments servent à distinguer des parties entières de texte comme les paragraphes **<p>**, les titres **<h>**, les listes **<ol>** ou **<ul>**, les citations **<blockquote>**, etc. Ces éléments sont distingués du reste du contenu par un retour chariot.
- La balise **<span>** est un élément en-ligne qui sert à distinguer une portion de texte incluse dans un élément bloc. Par exemple, quelques mots d'un paragraphe.

```
<span style="propriétés: valeur"> ... </span>  
<div style="propriétés: valeur"> ... </div>  
<span class="nom_de_classe"> ... </span>  
<div class="nom_de_classe"> ... </div>
```

### Exemples

```
1 <html>  
2 <head>  
3 <title>CSS</title>  
4 </head>  
5 <body>  
6 <h2>Nouveau, nouveau...</h2>  
7 <div style="background-color:#99ccff">Un concept nouveau est apparu  
8 dans l'univers de la publication Web, c'est celui des feuilles de style CSS  
9 </div>  
10 </body>  
11 </html>
```

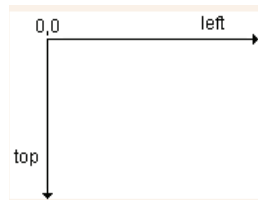
```
1 <html>  
2 <head>  
3 <title>CSS</title>  
4 </head>  
5 <body>  
6 <h2>Nouveau, nouveau...</h2>  
7 <p>Un concept nouveau est apparu dans l'univers de la publication Web, c'est  
8 celui des <span style="background-color:#99ccFF;">feuilles de style  
9 CSS</span>.  
10 </p>  
11 </body>  
12 </html>
```

Ce concept de boîte est dans certaines circonstances (avec ou sans doctype) interprété différemment par Explorer par Chrome et par Firefox. Sans entrer dans le détail, ce cas fournit un bel exemple de différences d'affichage d'un même code par des navigateurs différents.

## Exercice 6 –Le positionnement avec span et div

Il est possible de positionner, au pixel près, du texte ou une image avec les feuilles de style. Le positionnement peut se réaliser en position absolue ou en position relative.

La **position absolue** {position: absolute} se détermine par rapport au coin supérieur gauche de la fenêtre du navigateur. Les coordonnées du point initial sont top = 0 et left = 0. Les coordonnées d'un point s'expriment en pixels, de haut en bas pour top et de gauche à droite pour left.



La **position relative** {position: relative} se détermine par rapport à la position normale ou originale de l'élément dans la page, par exemple un élément du code Html.

Le positionnement absolu sera le plus souvent utilisé car il est plus facile et plus sûr. Mais il est important de ne pas oublier que le visiteur de votre site n'a peut-être pas la même résolution d'écran que vous. Ce qui tombe pile avec une résolution 800x600, risque de se retrouver n'importe où avec une résolution 1024x768.

Créer un effet d'ombrage en superposant, mais de façon légèrement décalée, un texte en rouge et son ombre (le même texte) en gris (créez deux zones en position absolue sous forme de <span> de 200 pixels de large et de 55 pixels de haut).

## Exercice 7 - Entraînement

1. Ecrire la syntaxe générale de la déclaration d'un style.
2. Ecrire le sélecteur et le style donnant une couleur rouge et un fond noir aux éléments <h1>et <h3>.
3. Ecrire le sélecteur afin que tous les éléments de la page soient écrits en vert.
4. Ecrire une classe qui définit un fond jaune et appliquez-la aux éléments <h1>et <p>.
5. Ecrire une classe spécifique à un élément <code>afin que son texte soit bleu.
6. Ecrire les classes correspondant aux styles « fondgris », « textevert » et « textejaune ». Appliquez la première et la troisième à un paragraphe, puis la première et la deuxième à une division <div>.
7. Ecrire le sélecteur afin que l'élément dont l'attribut id vaut « menu » ait un fond rouge.

## Exercice 8 - Synthèse

Réaliser la page HTML ci-dessous (le texte seulement) en utilisant la Doctype Strict et l'encodage UTF-8.

Créer une feuille de style externe pour obtenir le résultat ci-dessous.



## Annexes

### Les unités de mesure

Le HTML n'utilisait que le pixel ou le pourcentage. Les feuilles de style permettent d'utiliser d'autres unités de mesure.

px	en pixel
%	en pourcentage
in	en pouce (1 inch = 2,54 centimètres)
cm	en centimètre
mm	en millimètre
pt	en point (1 point = 1/72 inches)
pc	en pica (1 pica = 12 points)

### La codification des couleurs

Outre la codification par nom et en valeur hexadécimale, les feuilles de style introduisent d'autres façons d'encoder les couleurs. Soit :

- par un nom fonctionnel,
- par la valeur hexadécimale composée de 6 chiffres précédée d'un dièse # soit #000000. Donc comme en Html mais sans les guillemets !
- par une valeur hexadécimale à 3 chiffres. Chaque chiffre est alors implicitement dupliqué, ainsi #fd3 est équivalent à #ffdd33,
- par la notation fonctionnelle rgb qui prend 3 arguments en l'occurrence 3 nombres entiers compris entre 0 et 255 ou 3 pourcentages entre 0% et 100%, par exemple, color : rgb(255,0,0) ou color : rgb(50%,50%,50%).

# HTML 5 et CSS 3 (partie 1)

## Objectifs

- Connaître les différences au niveau des balises et des attributs,
- Créer un document HTML5,
- Définir une feuille de style.

Quelques règles simples qu'il faut respecter lors de l'écriture des balises HTML5 :

- En HTML5, les balises ne sont pas sensibles à la casse. On peut écrire indifféremment <BALISE>, <Balise> ou <balise>.
- La règle générale veut que toute balise ouverte <balise> doit être fermée </balise>. Les libertés prises dans l'écriture du HTML 3.2 à cause du fonctionnement plus ou moins permissif des navigateurs ne sont plus d'actualité. La rigueur apportée par le HTML 4.0 strict et le XHTML 1.0 doit continuer à guider votre écriture.
- Les balises doivent être correctement imbriquées. Lorsqu'on affecte plusieurs balises à un élément, l'ordre de fermeture de celles-ci est essentiel. La première balise de fermeture doit correspondre à la dernière balise d'ouverture non fermée.
- Les valeurs des attributs doivent toujours figurer entre des guillemets.

Les balises et attributs HTML 4.0 disparus :

- Disparition complète des cadres. Les balises <frame>, <frameset> et <noframes> ont définitivement disparu.
- Disparition de certaines balises de présentation comme <big>, <center>, <font>, <strike>, <tt> et <u>. Ici aussi, pas de grande surprise car le principe de la séparation de contenu et de la présentation est à présent bien établi. Ces balises de présentation doivent maintenant être prises en charge par des feuilles de style CSS.
- Disparition de tous les attributs relatifs à l'alignement, la largeur, les arrière-plans (de couleur ou avec image), les bordures et la numérotation des listes qui sont passés à la trappe. Tous ces attributs doivent être pris en charge par les feuilles de style qui deviennent ainsi indissociables du code HTML5.

Exemples :

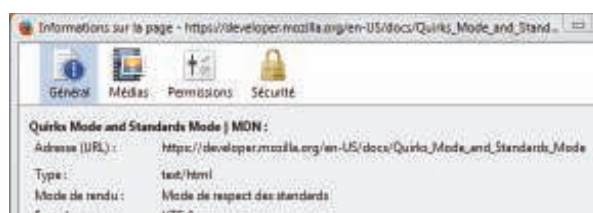
- align dans les balises <caption>, <iframe>, <img>, <input>, <object>, <legend>, <table>, <hr>, <div>, <hx>, <p>, <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> et <tr>.
- alink, link, text et vlink dans la balise <body>.
- background dans la balise <body>.
- bgcolor dans les balises <table>, <tr>, <td>, <th> et <body>.

## Le document HTML5 minimum

Tout document HTML doit commencer par un **doctype**. Le HTML5 propose un doctype unique et simplifié :  
(<!DOCTYPE html>)

Le doctype, aussi appelé DTD sert à indiquer au navigateur à quelles règles d'écriture obéit le code source de la page HTML ou XHTML. Il utilisera celles-ci pour afficher la page selon les standards du W3C. Tous les navigateurs appliquant la même règle, vous êtes ainsi assuré d'un rendu identique entre les différents navigateurs.

En l'absence d'un doctype, le navigateur ignore selon quelles règles il doit traiter la page. Il se rabat ainsi sur des procédures qui lui sont propres pour afficher votre page HTML. Ce qui peut entraîner des différences de restitution sensibles entre les différents navigateurs. Ce mode bancal est appelé **mode compatibilité** ou **quirks mode**. Avec Firefox, cliquer sur le **bouton droit** de la souris puis **informations sur la page**.





La balise `<html>` indique au navigateur qu'il s'agit d'un document Html. La balise `<html>` est l'élément le plus haut ou l'élément racine du document. Elle prend donc place juste après la déclaration de doctype.

```
<!DOCTYPE html>
<html lang="fr">
...
</html>
```

La balise `<html>` peut prendre comme attribut `lang="fr"` qui spécifie que le document est en langue française. Cette information est fort prisée par les moteurs de recherche comme Google ainsi que par les synthèses vocales utilisées par les personnes malvoyantes.

La balise `<head> ... </head>` qui se positionne juste après le doctype et la balise `<html>` contient ce qu'on appelle l'en-tête du document. Dans cet en-tête du document se situe toute une série d'informations relatives au document lui-même, indépendamment du contenu de celui-ci. Ces informations peuvent être de natures très diverses :

- Le titre du document : la balise `<title> ... </title>`.
- Des déclarations ou des appels de feuilles de style CSS.
- Des fonctions ou des liens vers des fichiers JavaScript.
- Des informations à l'intention des moteurs de recherche pour le référencement de la page.
- Des informations à l'intention des navigateurs.
- Des balises méta qui reprendront quant à elles l'encodage, la description de la page, les mots-clés associés à la page, le nom de l'auteur, des mentions de copyright, etc.

### L'encodage du document

La notion du jeu de caractères utilisé (**charset**) est une notion essentielle dans le développement de pages Web. Le jeu de caractères désigne la façon dont les caractères d'un alphabet donné sont convertis en octets dans un fichier informatique (et vice-versa). Certaines méthodes d'encodage sont spécifiques à un environnement informatique dans une langue ou un alphabet donné, d'autres sont multiplateformes et multilingues.

ASCII	Jeu de caractères basique mais relativement ancien qui a permis l'échange de textes en anglais à un niveau mondial. Il a été repris à l'origine de la toile mais comme l'anglais ne comporte pas d'accents, il fallait encoder les caractères accentués et spéciaux par des entités du genre <code>&amp;acute;</code> (pour le é) ou <code>&amp;euro;</code> (pour l'€).
ISO-8859-1	Extension de l'ASCII, l'ISO-8859-1, souvent appelée Latin-1 ou Europe occidentale, est le codage fréquemment utilisé dans notre région du monde. Il ajoute de nombreux caractères latins ainsi que quelques autres signes de ponctuation ou symboles.
ISO-8859-15	Version actualisée de l'ISO-8859-1 qui comporte entre autres le signe €.
UTF-8	C'est un format de codage de caractères (charset) conçu pour coder l'ensemble des caractères internationaux d'Unicode de plusieurs langues, contrairement aux formats plus vieux comme l'iso-8859-1 qui requiert une forme d'encodage afin d'accepter, par exemple, des lettres accentuées ou des caractères spéciaux. Son principal avantage pour le web est qu'il permet de représenter les milliers de caractères des différentes langues et permet donc de créer des sites Web qui puissent être multilingues (français, chinois ou arabe,...) sans avoir à créer de multiples versions du site dans l'encodage du pays concerné.

Depuis fin 2011, le WC3 recommande le charset UTF-8 pour l'édition de pages Web en grande partie pour son universalité. Les navigateurs possèdent bien une fonction de détection automatique du jeu de caractères.

Voici donc le document HTML5 minimum :

```
1 <!doctype html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8">
5   <title>Titre de la page</title>
6   <link rel="stylesheet" href="style.css">
7   <script src="script.js"></script>
8 </head>
9 <body>
10   ...
11   <!-- Le reste du contenu -->
12   ...
13 </body>
14 </html>
```

## La validation du code HTML5

Les avantages d'un code parfait et de la validation sont :

- Une vérification en profondeur du HTML5 pour les codeurs débutants (et confirmés).
- L'assurance que votre page sera toujours correctement affichée par la plupart des navigateurs. Une page invalide peut amener les navigateurs à interpréter celle-ci de manière très différente.
- La démonstration de vos capacités professionnelles en fournissant un code de qualité, conforme au standard du HTML5.
- Les synthèses vocales et autres aides techniques à l'attention des personnes moins valides se basent sur un code valide pour restituer les pages de la toile.

Si le document soumis au validateur n'est pas correct, celui-ci vous fournira une analyse des problèmes qui vous permettra de corriger votre code. Les premières validations peuvent être très frustrantes mais constituent un excellent apprentissage du code source. Le rapport d'erreur est cependant souvent très technique.

- Le **W3C** lui-même présente un validateur de code (<http://validator.w3.org/>).
- Le site **validator.nu** propose également au stade expérimental un validateur HTML5 (<http://html5.validator.nu/>).

### Exercice 1 – Validation d'une page HTML5

1. Ouvrir la page **erreur.html** et repérer les erreurs de codage.
2. Valider la page **erreur.html** sur le site du **W3C** puis sur le site **validator.nu**.
3. Les validateurs détectent-ils les erreurs ?
4. Corriger les erreurs.

### Exercice 2 – Le texte

Pour mettre du texte en gras (bold en anglais), il suffit d'inclure celui-ci entre les balises `<b> ... </b>`.

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>Html5</title>
5  <meta charset="UTF-8">
6  </head>
7  <body>
8  <b>Du texte en gras</b>
9  </body>
10 </html>
```

1. Mettre en gras avec une feuille de style CSS sans passer par la balise `<b>`.
2. Procéder de même pour du texte en italique, la taille et la couleur des caractères, l'alignement et le soulignement d'un mot.

### Exercice 3 - La structuration d'un document

Le langage HTML5 propose six grandeurs de caractères pour les titres (balise `h<x>`, `x` varie de 1 à 6). Le W3C insiste pour qu'en HTML5, les balises de titre `<h<x>` retrouvent toutes leurs qualités de structuration des pages Web. L'usage des balises de titres doit donc être réservé à leur fonction originelle. Il importe de ne pas les détourner de leur but premier pour, par exemple, mettre du texte en gras ou avec une taille de caractères déterminée. Les feuilles de style CSS sont là pour réaliser ces présentations.

1. Ecrire le code pour indenter les titres de niveau 1 à 3.
2. Valider le code.

## Exercice 4 – Les divisions de page

Il est possible de diviser en bloc et en ligne des éléments. Il existe deux grands groupes de balises : les balises de rendu **CSS bloc** (block) et les balises de rendu **CSS en ligne** (inline).

Les balises de rendu CSS bloc occupent par défaut toute la largeur de la fenêtre du navigateur. Elles se placent ainsi les unes au-dessus des autres. Généralement, le navigateur insère automatiquement un espace au-dessus et en dessous du contenu de la balise. C'est le cas de la balise de titre `<h1>` et de toutes les balises de ce chapitre. Elles sont toutes des balises de type bloc.

Une balise de titre ou de paragraphe occupe toute la largeur de la page et est séparée par un espace plus ou moins important.

Le code `<p>Paragraphe 1</p><p>Paragraphe 2</p>` va s'afficher sur deux lignes car la balise `<p>` est une balise de type bloc. Chaque paragraphe va occuper une ligne.

Les principaux éléments de division de type bloc sont :

- Les balises de titres `<h1> ... </h1>`.
- Les balises de paragraphe `<p> ... </p>`.
- Les balises de citation `<blockquote> ... </blockquote>`.
- La balise `<hr>` qui introduit une ligne horizontale.
- Les balises `<ol> ... </ol>` des listes ordonnées.
- Les balises `<ul> ... </ul>` des listes à puces.
- Les balises `<dl> ... </dl>` des listes de définitions.
- Les balises de tableaux `<table> ... </table>`.
- Les balises de déclaration de formulaires `<form> ... </form>`.
- La balise `<div> ... </div>` spécialement conçue pour introduire une division (div pour division) sans passer par une des balises citées ci-avant. Cette balise est particulièrement utile pour appliquer une déclaration de feuilles de style CSS.

Au contraire des éléments de type bloc, les éléments en ligne se placent toujours l'un à côté de l'autre afin de rester dans le flux du texte. Ainsi, le code texte en `<b>gras</b>` et en `<i>italique</i>` est restitué sur une seule ligne, sans interrompre le flux du texte : texte en gras et en italique.

Les principales balises en ligne sont :

- La balise `<b> ... </b>` de mise en gras du texte.
- La balise `<i> ... </i>` de mise en italique du texte.
- La balise `<br>` de passage à la ligne.
- La balise `<a>` pour les liens.
- La balise `<img>` pour les images.
- Les balises de champs de formulaires `<input>`, `<select> ... </select>` et `<textarea> ... </textarea>`.
- La balise `<span>` pour introduire une division en ligne dans le texte. Cette balise est particulièrement utile pour appliquer une déclaration de feuilles de style CSS.

Une balise de type bloc peut contenir une (ou plusieurs) balise(s) de type bloc. Ainsi, par exemple, une citation incluse dans un paragraphe. L'extension **Web Developer** de Firefox permet de visualiser les différents éléments bloc d'une page et par conséquent la structure de celle-ci (<https://addons.mozilla.org/fr/firefox/addon/web-developer/>).

## TP4 – HTML 5 et CSS 3 (partie 1)

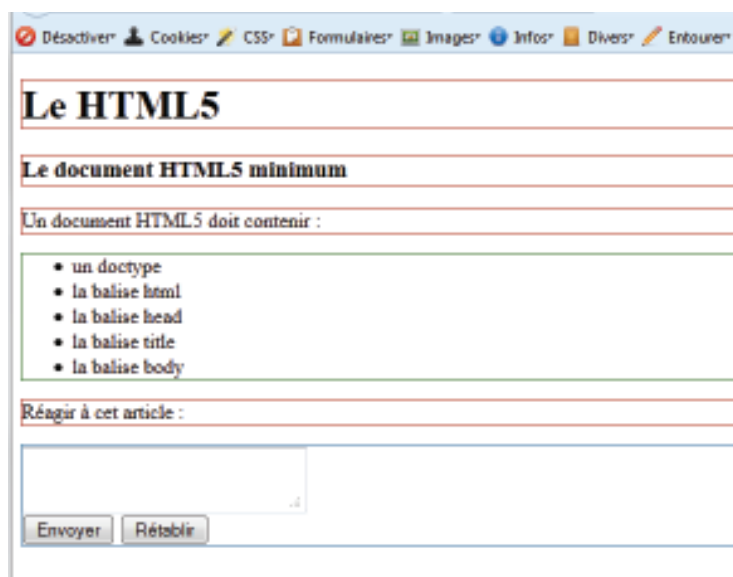
1. Reproduire la page HTML ci-dessous :

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>HTML5</title>
5  <meta charset="UTF-8">
6  </head>
7  <body>
8  <div> HTML5</div>
9  <div> Le document HTML5 minimum</div>
10 <p>Un document HTML5 doit contenir :</p>
11 <ul>
12 <li>un doctype</li>
13 <li>la balise html</li>
14 <li>la balise head</li>
15 <li>la balise title</li>
16 <li>la balise body</li>
17 </ul>
18 <p>Réagir à cet article :</p>
19 <form action="">
20 <textarea rows="2" cols="25"></textarea><br>
21 <input value="Envoyer" type="submit">
22 <input value="Rétablir" type="reset">
23 </form>
24 </body>
25 </html>

```

2. Installer l'addon **Web-Developer** de Firefox et entourer les différents éléments de type blocs de la page.



## Exercice 5 – Les liens

Les liens constituent l'essence du langage Html et des pages Web. La richesse de la fonction hypertexte permet de tisser cette toile gigantesque qu'est le Web.

La balise `<a> ... </a>` introduit un lien. Sa syntaxe de base est : `<a href="destination_du_lien">Texte de lien</a>`

La destination du lien peut être :

- Un endroit de la page en cours.
- Une autre page du site.
- Un endroit dans une autre page du site.
- Une page d'un autre site situé sur le Web.
- Une adresse électronique.
- Un fichier à télécharger.

La balise de lien possède différents attributs :

- href  
L'attribut **href** définit l'adresse (url) de la cible du lien.
- hreflang  
Précise la langue du document cible si celle-ci est différente de celle du document de départ.

## TP4 – HTML 5 et CSS 3 (partie 1)

- ping  
**Nouveau en HTML5.** L'attribut ping peut contenir une liste d'adresses url (séparées par un espace) qui recevront une notification lorsque l'utilisateur suit le lien.  
`<a href="html5.htm" ping="http://www.monsite.com/data/visites.php">Html5</a>`  
 Le navigateur va alors envoyer une requête POST à l'adresse url spécifiée dans l'attribut. Ce nouvel attribut sera particulièrement utile pour les statistiques d'un site. Dans le cas présent, il permettra de connaître le nombre de visiteurs de la partie consacrée au Html5.
- rel  
**Nouveau en HTML5.** Spécifie la relation entre le document de départ et le fichier cible du lien.  
 Les valeurs sont nombreuses : alternate, archives, author, bookmark, contact, external, first, help, icon, index, last, license, next, nofollow, noreferrer, pingback, prefetch, prev, search, stylesheet, sidebar, tag, up.  
 Cet attribut n'est que très partiellement implémenté dans les navigateurs récents.
- target  
 Spécifie au navigateur la façon d'afficher la cible du lien. Cela peut être dans une nouvelle instance du navigateur ou un nouvel onglet du navigateur (target="\_blank"), dans la même fenêtre que la page de départ du lien (target="\_self") et dans la même fenêtre mais en occupant la totalité de la fenêtre du navigateur.
- type  
 Indique au navigateur le type MIME<sup>1</sup> de la cible si celui-ci n'est pas un document HTML5, par exemple un fichier son ou une image.

Les attributs HTML 4.0 suivants ont disparu de la spécification HTML5 : charset, coords, name, rev et shape.

### Liens vers une ancre située dans la même page

Après avoir défini l'ancre, il faut effectuer le lien vers celle-ci. Pour créer ce lien, il suffit de reprendre dans l'attribut href le nom de l'ancre directement précédé du signe dièse (#) : `<a href="#nom de l'ancre">Texte du lien</a>`

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>HTML5</title>
5 <meta charset="UTF-8">
6 </head>
7 <body>
8 <a href="#chapitre2">Aller au chapitre 2</a>
9 <h2>Chapitre 1</h2>
10 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
11 non risus. Suspendisse lectus tortor, dignissim sit amet,
12 adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
13 diam...</p>
14 <h2>Chapitre 2</h2>
15 <a id="chapitre2"></a>
16 <p>Contenu du chapitre 2</p>
17 </body>
18 </html>

```

1. Réaliser un lien vers une ancre situé dans une seconde page.
2. Réaliser un lien pour télécharger le fichier pdf de ce TP.
3. Réaliser un lien pour ouvrir le site <https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5> dans un nouvel onglet.
4. Ajouter une infobulle sur le lien précédent.
5. A l'aide d'une feuille de style, enlever le soulignement des liens (text-decoration: none) et uniformiser la couleur des liens (propriété de style color).

Les liens peuvent maintenant porter sur n'importe quel élément de type bloc comme les divisions (<div>) et items de liste (<li>).

6. Réaliser le code ci-dessous et modifier-le pour que le lien s'ouvre dans un nouvel onglet. Conclure.

<sup>1</sup> Un Internet media type1, à l'origine appelé type MIME ou juste MIME ou encore Content-type2, est un identifiant de format de données sur internet en deux parties. Un type et un sous-type et d'un ou plusieurs autres champs au besoin. Par exemple, les sous-types du type text ont un champ optionnel charset indiquant le codage des caractères ; ou encore, les sous-types du type multipart définissent une option boundary entre les parties.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>html5</title>
5 <style>
6 div { width: 200px;
7       height: 100px;
8       border: 1px solid black;
9       padding-left: 10px;
10  a { text-decoration: none;
11     color: black;
12  }
13 </style>
14 <meta charset="UTF-8">
15 </head>
16 <body>
17 <p>Cliquez sur l'extrait pour le texte complet.</p>
18 <a href="#">Extrait</a>
19 <div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
20 non risus. Suspendisse lectus tortor, dignissim sit amet.
21 adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
22 diam, ...</div>
23 </a>
24 </body>
25 </html>

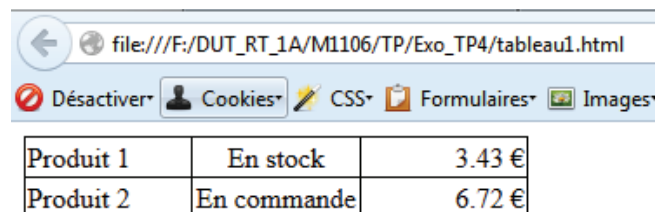
```

## Exercice 6 – Les tableaux

Un tableau (<table>) comporte des lignes <tr>. Ces lignes contiennent des cellules <td>. Les cellules peuvent contenir tous les éléments définis par le HTML, soit du texte, des images, des liens, des arrière-plans et même des tableaux.

En HTML5, les attributs width, height, align, valign, abbr, axis et scope du HTML 4.0 ont disparu.

1. Réaliser le tableau ci-dessous. La mise en forme du tableau est réalisée par une feuille de style interne.



Produit 1	En stock	3.43 €
Produit 2	En commande	6.72 €

2. Ajouter un arrière-plan de couleur pour les cellules des prix.

La **balise <caption>** permet d'associer un titre ou une légende au tableau. La balise <caption> doit être placée juste après la balise ouvrante <table> et ne peut apparaître qu'une fois dans le tableau. Par défaut, la plupart des navigateurs affichent le contenu de la balise <caption> de façon centrée au-dessus du tableau. Il est possible de mettre cette légende en dessous du tableau à l'aide de la propriété de style CSS caption {caption-side: bottom;};

3. Insérer la légende suivante « Tarifs des produits festifs » en-dessous du tableau.

Le HTML5 prévoit des balises pour structurer de façon logique le contenu d'un tableau. Ces balises sont :

- **<thead>** pour regrouper les informations concernant l'en-tête du tableau comme par exemple le titre et l'intitulé des colonnes.
- **<tbody>** pour le corps du tableau, soit l'ensemble des données de celui-ci.
- **<tfoot>** pour le contenu de bas de page comme par exemple des remarques ou une légende.

Ces divisions logiques n'affectent en rien la présentation par défaut du tableau mais elles peuvent être reprises par des propriétés de style pour agrémenter la présentation du tableau. La balise <tfoot> doit se situer avant la balise <tbody> de sorte que le navigateur puisse prévoir le pied de page avant la réception des lignes de données.

4. Construire le tableau ci-après en utilisant les balises précédentes.



La fusion	
colspan	Pour la fusion de colonnes
rowspan	Pour la fusion de lignes
Source : W3C Working Draft 24 June 2018	

## Exercice 7 – Les images et arrière-plans

Pour réduire au maximum le temps de téléchargement, le Web réclame des formats de fichier particuliers ; le format GIF, JPEG et PNG. Ces trois types de format comportent tous un algorithme de compression.

### ▪ Le format GIF

Le **format GIF** (pour **Graphics Interchange Format**) a connu son heure de gloire aux débuts du Web. Il était alors parfaitement adapté aux cartes graphiques de l'époque (souvent maximum 256 couleurs) et aux faibles capacités des lignes de téléphone analogiques (taux de compression élevé).

En raison de sa limitation à 256 couleurs, le format GIF n'est pas du tout approprié aux graphiques à haute résolution comme les photos, les images nuancées et les dégradés. Par contre, il excelle encore pour les petits logos, boutons, puces, barres, symboles, et autres cliparts fréquents pour certains graphiques du Web qui ne nécessitent pas une palette de couleurs très étendue. Il est cependant de plus en plus supplanté par le format PNG.

### ▪ Le format JPEG

Le **format JPEG** figure pour **Joint Photographic Expert Group**, donc non pas pour le nom du format graphique mais pour le nom de la corporation qui a développé ce format. C'est le type de format le plus répandu dans l'univers de la photo numérique.

Le JPEG est un excellent format pour les photos, spécialement celles où il y a des tonalités diverses et des dégradés de couleurs. Mais il n'est pas exempt de défauts avec ses pertes de données et ses déformations.

À cause des pertes de données et des déformations, il est impératif de respecter la règle de sauvegarder l'image originale et de travailler uniquement sur des copies de celle-ci.

### ▪ Le format PNG

Le **format PNG**, pour Portable Network Graphic, est présenté comme le format du futur. Il s'agit d'un format graphique conçu spécialement pour la mise en œuvre sur le Web. La recommandation du W3C concernant le PNG remonte à 1996. Le PNG doit théoriquement rassembler tous les avantages de GIF et du JPEG. Il commence à être largement retenu par les concepteurs d'applications Web.

Le format PNG est appelé à être de plus en plus utilisé à l'avenir car il représente un excellent compromis entre le format GIF et JPEG. Les professionnels l'adoptent largement en lieu et place du GIF et pour les petites photos.

### ▪ Le format WebP

Le **WebP** est un nouveau format d'image qui risque de prendre de l'importance dans les prochaines années. Ce format est développé par Google à partir d'un des formats de la nouvelle balise <video> du Html5 (le WebM). Selon les premiers essais, le WebP réduirait la taille des fichiers de 39 % par rapport aux formats JPEG, PNG et GIF, sans perte de qualité perceptible.

## Le poids des photos

Il semble illusoire de charger directement une photo de votre appareil photo numérique sur un site. Certaines "pèsent" jusqu'à 4 Mo, ce qui, même avec une connexion rapide, prendra un temps certain de chargement.

On peut conseiller :

- De recadrer les images pour ne recueillir que ce qui est utile.
- De redimensionner cette image recadrée.
- D'adopter un taux de compression d'environ 15 à 20 % pour les images JPEG. La perte de qualité n'est quasiment pas perceptible à l'œil normal.
- De diminuer le nombre de couleurs de l'image, si l'opération s'y prête.

Ces différentes manipulations devraient ramener la taille du fichier à une dimension beaucoup plus raisonnable et améliorer sensiblement le temps de téléchargement.

## L'insertion d'une couleur d'arrière-plan

Depuis la disparition de l'attribut bgcolor, il n'est plus possible d'ajouter un arrière-plan de couleur à un élément. Il faut impérativement passer par les propriétés de style CSS, **background-color** en l'occurrence.


## L'insertion d'une image d'arrière-plan

L'attribut background étant déclaré obsolète en HTML5, il n'y a d'autre solution que de passer par la propriété de style **background-image** pour insérer une image d'arrière-plan.



1. Réaliser le tableau ci-dessous.
2. Positionner le logo HTML5.
3. Bonus : insérer en arrière-plan le logo DUT Réseau-Télécom

### Module M1106

<b>UE 11</b>	<b>Développement et administration des services réseaux</b>	Volume horaire : <b>30 h</b>
Référence du module <b>M1106 (T3-12)</b>	Nom du module <b>Initiation au développement Web</b>	<b>Semestre 1</b>
<b>Objectif du module :</b> <ul style="list-style-type: none"> <li>• Homogénéiser les notions abordées pour le Bac</li> <li>• Découvrir le contenu et le contenant d'une page Web qui présente des informations sur Internet pour PC, Tablette et Smartphone</li> <li>• Découvrir les notions de base sur le fonctionnement du modèle client-serveur</li> <li>• Savoir écrire et modifier des pages Web dans un langage normalisé de description de contenus</li> <li>• Savoir modifier les attributs de présentation d'une page Web</li> <li>• Savoir adapter un contenu Web pour navigateur mobile</li> </ul>		
<b>Compétences visées :</b> <ul style="list-style-type: none"> <li>• Être capable de formater et de modifier des documents dans un langage normalisé de description de contenus</li> <li>• Être capable de mettre en œuvre des feuilles de styles</li> <li>• Être capable de mettre en œuvre le Responsive Design</li> </ul>		
<b>Pré requis :</b> <ul style="list-style-type: none"> <li>• Bac ou équivalent</li> </ul>		
<b>Contenus :</b> <ul style="list-style-type: none"> <li>• Langage normalisé de description de contenus Web <ul style="list-style-type: none"> <li>○ Un document minimal – structure et gestion de l'encodage</li> <li>○ Les éléments structurant un document</li> <li>○ Les éléments multimédias</li> <li>○ Les formulaires: types d'acquisition; attributs</li> <li>○ Les attributs événements</li> </ul> </li> <li>• Feuilles de styles</li> <li>• Responsive Design</li> </ul>		
<b>Modalités de mise en œuvre :</b> <ul style="list-style-type: none"> <li>• Pédagogie par projet</li> </ul>		
<b>Prolongements possibles :</b> <ul style="list-style-type: none"> <li>• Web dynamique (M2105 )</li> <li>• Application informatique dédiée aux R&amp;T (M4207C)</li> </ul>		
Source : PPN DUT R&T 2013		

### Complément sur les unités

Les unités **em** et **ex** dépendent de la police de caractères et peuvent être différentes pour chaque élément du document. L'unité em est simplement la taille de la police de caractères. Pour un élément dont la police est de taille 2in, 1em signifie 2in. Exprimer des tailles, comme les **margins** et les **padding**s, en em signifie qu'elles sont relatives à la taille de la police, et si l'utilisateur a une large police de caractères (sur un grand écran par exemple) ou au contraire une petite police (sur un smartphone par exemple), les tailles seront en proportion.

<http://www.w3.org/Style/Examples/007/units.fr.html>

### Références

<https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5>

<http://www.w3.org/TR/html51/>