



# Tutoriel HTML

## Principaux éléments de structuration

D'après les cours de Sylvain Brandel et d'Olivier Glück (Lyon 1)  
Remanié par Lionel Médini

**M1 Informatique**

**MIF13 – Programmation web**

**Université Claude Bernard Lyon 1**

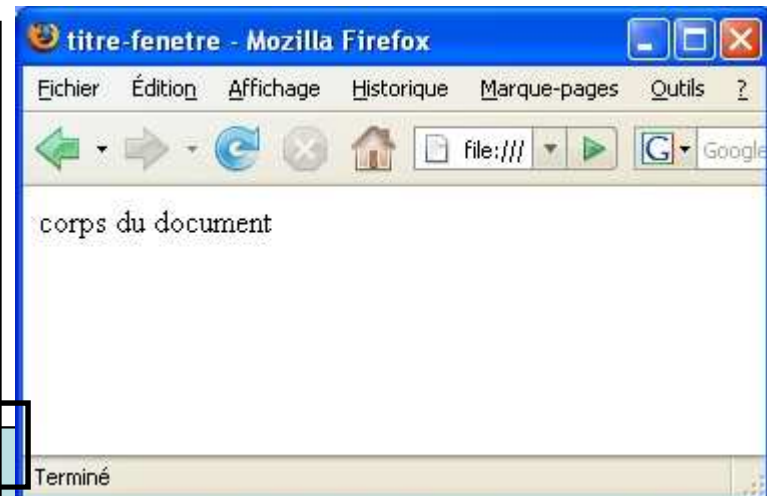
**UFR d'informatique**

# Structuration minimale d'un document

- `<html>` :
  - Définit le contenu du document HTML
- `<head>` :
  - En-tête du document : contient les méta-données et informations techniques
- `<title>` :
  - Titre général du document : affiché en haut de la fenêtre, utilisé dans les signets (*bookmarks*)
- `<body>` :
  - Corps du document : contenu affiché dans la fenêtre/ l'onglet du navigateur

```
<html>
  <head>
    <title>titre-fenetre</title>
  </head>
  <body>
    corps du document
  </body>
</html>
```

Attention à l'indentation...



# Syntaxe : éléments et commentaires

- Syntaxe des éléments : 2 possibilités
  - Une balise ouvrante, un contenu, une balise fermante `<p>bonjour</p>`
  - Une balise d'élément vide : `<hr />`
- Contenus des éléments : dépendent de leurs types

Type	Contenus	Exemples
En-ligne	Texte, éléments en-ligne	<code>&lt;span&gt;Ceci est un &lt;a href="toto"&gt;lien&lt;/a&gt;.&lt;/span&gt;</code>
Bloc	Texte, éléments blocs et en-ligne	<code>&lt;div&gt; &lt;h1&gt;titre&lt;/h1&gt; &lt;p&gt;du texte&lt;br/&gt;sur deux lignes&lt;/p&gt; &lt;/div&gt;</code>

➔ **Attention : nombreuses exceptions** (voir chaque type d'élément)

- Commentaires

`<!-- Ceci ne sera pas interprété -->`

# Syntaxe : attributs

- **Rappel** : un attribut est toujours défini
  - Dans une balise ouvrante
  - Dans une balise d'élément vide
- **Syntaxe** : nom = valeur entre guillemets
  - `<balise attribut1="10" attribut2="blue">`
- Attributs "de cœur" (disponibles pour la plupart des éléments)
  - `class="name"` : applique un style au contenu
  - `id="name"` : donne un nom unique à la balise
  - `lang="langage"` : spécifie la langue du contenu (ISO639)
  - des attributs liés aux événements :
    - `onclick="action"`,
    - `ondblclick`,
    - `onkeydown`,
    - `onkeypress`,
    - `onkeyup`,
    - `onmouseover`,
    - `onmouseout`,
    - ...

# Caractères spéciaux

- Références à des entités : `&nom_entité;`
  - Proviennent de la DTD HTML
    - ➔ correctement interprétées par tous les navigateurs
  - Indépendante du jeu de caractères utilisé
  - caractères accentués : `&<lettre><accent>;`
    - `accent` : `acute` (aigu), `grave` (grave), `circ` (^), `uml` (¨)
    - exemples : `&eacute;` (é), `&Icirc;` (Î), `&uuml;` (ü)
  - autres caractères :
    - `&ntilde;` (ñ), `&ccedil;` (ç), `&szlig;` (ß), `&copy;` (©)
    - `&aelig;` (æ), `&amp;` (&), `&nbsp;` (espace), `&quot;` (")
    - `&gt;` (>), `&lt;` (<)
- Référence au code Unicode du caractère : `&#code_Unicode;`
  - Exemple : caractère  $\Phi$  = `&#934;`

## Principaux éléments de structuration

- Permettent à la fois de structurer un document et de définir ses principaux styles (comme dans un traitement de texte)
  - Titres : éléments `<hn>...</hn>`
    - n varie de 1 à 6
    - Mise en forme spécifique (y compris dans les styles par défaut)
  - Paragraphes : élément `<p>...</p>`
- Éléments de type blocs
  - Peuvent être fils directs de body
  - Peuvent être imbriqués dans des div
  - Contenu : texte, éléments en-ligne (images, ancres de liens, sauts de ligne...)
  - Ne peuvent être imbriqués les uns dans les autres

# Éléments <h1>, <h2>... Et <p>

```
<html>
  <head>
    <title>titre-fenetre</title>
  </head>
  <body>
    <h1>Titre niveau 1</h1>
    <p>paragraphe 1</p>
    <h2>Titre niveau 2</h2>
    <p>paragraphe 2</p>
    <h3>Titre niveau 3</h3>
    <p>paragraphe 3</p>
  </body>
</html>
```



## Éléments `<hr />` et `<br />`

- `<hr />` : trait horizontal
  - Balise d'élément vide
  - Élément de type **bloc**
  - Exemples
    - `<hr size="2" />` (en pixels)
    - `<hr width="30%" />`
    - `<hr width="100" />` (en pixels)
    - `<hr align="center/left/right" />`
    - `<hr noshade />` (pas de relief)
- `<br />` : saut de ligne
  - Balise d'élément vide
  - Élément de type **en-ligne**
  - Pas d'attribut de style



# Listes

- Principe : 2 types de listes contenant des items
  - Contenu : éléments `<li>` (List Item) , de type bloc
- Listes non numérotées : `<ul>` (Unordered List)
  - attribut `type="square/circle/disc"` pour `<ul>` ou `<li>`
- Listes numérotées : `<ol>` (Ordered List)
  - attribut `type="1/A/a/I/i"` pour `<ol>`
  - attribut `start="valeur"` pour `<ol>` (valeur de départ)
  - attribut `value="valeur"` pour `<li>` (réinitialise le séquençement à la nouvelle valeur)
- Remarque : il existe un troisième type de liste pour les définitions
  - Moins utilisé → cf. spécifications ou tutoriels sur le Web

## Exemple de listes non ordonnées

```
<html>
  <head>
    <title>titre-fenetre</title>
  </head>
  <body>
    <ul>
      <li>élément 1</li>
      <li>élément 2</li>
    </ul>
    <ul>
      <li>élément 3.1</li>
      <li>élément 3.2</li>
    </ul>
  </ul>
</body>
</html>
```



## Styles physiques

- Indépendants du navigateur utilisé et de sa configuration
  - `<b>...</b>` : gras
  - `<i>...</i>` : italique
  - `<tt>...</tt>` : machine à écrire (police largeur fixée)
  - `<strike>...</strike>` : texte barré
  - `<u>...</u>` : souligné
  - `<sub>...</sub>` : indice
  - `<sup>...</sup>` : exposant
  - `<small>...</small>` : petite police
  - `<big>...</big>` : grande police

➔ Ne plus les utiliser (remplacer par des propriétés CSS)

## Styles logiques

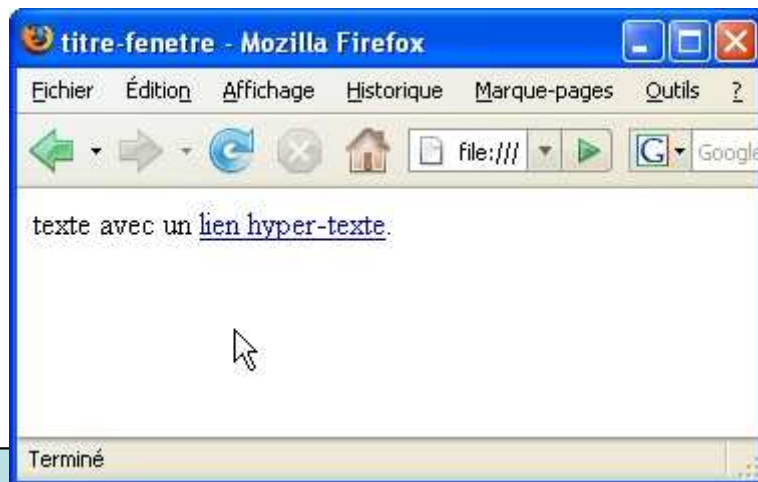
- Peuvent dépendre du navigateur utilisé et de sa configuration
  - `<strong>` : gras
  - `<em>` : mettre un texte en valeur (italique)
  - `<dfn>` : définition
  - `<cite>` : citation bibliographique
  - `<code>` : programme informatique
  - `<blockquote>` : tabulation + espaces paragraphes
  - ...

## Élément <a> Liens

- Définition d'un hyperlien
  - On définit l'**anc**re d'un lien dans un document
  - On spécifie la **cible** de ce lien (autre document, fichier, fragment...)
- Élément utilisé : **<a>** (anchor)
  - Attributs
    - **href** : URL de la cible
      - absolue : `http://www.w3.org/TR/xhtml1/`
      - relative : `Cours/index.html`
      - peut se terminer par une étiquette pour spécifier un fragment : `#label`
    - **title** : texte qui s'affiche dans une info-bulle
  - Contenu
    - Type de l'élément `<a>` : en-ligne
    - Peut contenir
      - Du texte, et d'autres éléments en-ligne (`<br/>`, `<img/>`...)
    - Exception : ne peut pas contenir un autre élément `<a>`

# Liens hypertexte

```
<html>
  <head>
    <title>titre-fenetre</title>
  </head>
  <body>
    <p>Texte avec un <a href="cible.html" title="cliquez
ici">lien hyper-texte</a>.</p>
  </body>
</html>
```



# Insertion d'images

- Insertion d'une image dans un document :
  - **src="nom"**
    - URL de la ressource permettant d'accéder à l'image (locale ou distante)
    - extensions : jpg, gif, xbm...
  - **alt="texte alternatif"**
    - Si le navigateur n'arrive pas à charger/afficher l'image et
    - Pour son indexation
  - Contenu
    - Balise d'élément vide → aucun contenu autorisé
  - Caractéristiques de style
    - Une image peut être redimensionnée en hauteur/largeur
    - Par défaut, les proportions sont conservées
    - Utiliser du **CSS** et non les attributs **height** et **width**

# Insertion d'images

```
<html>
  <head>
    <title>titre-fenetre</title>
  </head>
  <body>
    <p>Texte avant l'image</p>
    <p></p>
    <p>Texte après l'image</p>
  </body>
</html>
```





# Inclusion d'objets

- Élément `<object>`
  - Même principe pour n'importe quel type d'objet (applet, flash, autre page HTML...)
  - Contenu
    - Éléments param : `<param name="nom_param1" value="val_param1"/>`
  - Attributs
    - **type**="type\_MIME" : permet au navigateur de savoir comment traiter le contenu
    - **data**="URL\_données"
- Élément `<embed>` (déprécié en XHTML 1.0)
  - Permet d'inclure des fichiers multimédias : sons, vidéos
  - Attributs
    - **src**="URL\_fichier"
    - **autostart**="true|false" : lancement automatique du lecteur
    - **hidden**="true|false" : affichage ou non de la boîte de dialogue du lecteur multimédia associé au fichier
    - **loop**="n|infinite"

## Formatage du texte

- Comportement par défaut pour les séparateurs
  - N'importe quelle séquence de séparateurs (espaces, sauts de ligne) est affichée sous la forme d'une espace sécable
- Pour afficher plusieurs espaces
  - Utiliser le caractère « espace insécable » : `&nbsp;` ; (non conseillé)
  - Positionner son texte à l'aide de caractéristiques de style CSS
- Pour afficher un saut de ligne
  - Élément `<br />` (voir plus loin)
- Pour afficher du texte pré-formaté
  - Utiliser l'élément `<pre>` (preformatted)
  - Conserve le formatage du texte tel qu'il a été saisi dans le fichier source HTML (espacement, sauts de ligne...)
  - Utile pour les formules mathématiques, les textes en vers...

## Tableaux

- Complètement définis et normalisés dans HTML 4.0
- Permettent de présenter des données de façon répétitive et ordonnées
- Les cellules (cases) peuvent contenir des données multimédia (textes, images, liens...)
- Les tableaux ont été très utilisés pour gérer la mise en page  
**La bonne façon de faire est d'utiliser des caractéristiques de style CSS**

## Éléments `<table>` `<tr>` `<th>` `<td>`

- `<table>...</table>` : définit un tableau (élément de type bloc)
- Spécification d'une ligne (Table Row)
  - `<tr>...</tr>` : marquent le début et la fin d'une ligne du tableau
- Spécification des cellules
  - `<th>...</th>` : cellules d'en-tête (Table Header)
  - `<td>...</td>` : cellules de données (Table Data)
- Accélérer l'affichage des tableaux (HTML 4.0)
  - fixer le nombre de colonnes  
`<table cols="nb">`

## Élément `<caption>` Fusion de cellules

- Fusions de cellules

- `<th colspan="nombre">` : définit le nombre de colonnes sur lesquelles s'étend une cellule (=nombre de cellules à fusionner sur une ligne)

- `<td rowspan="nombre">` : définit le nombre de lignes sur lesquelles s'étend une cellule (= nombre de cellules à fusionner sur une colonne)

- ces fusions doivent être cohérentes par rapport au lignes et colonnes du tableau

- `<caption style="align: left | center | right;">` : permet de donner un titre au tableau