

INTRODUCTION AU DEVELOPPEMENT « WEB »

HTML ...



Erwan TRANVOUEZ

erwan.tranvouez@univ-amu.fr

<http://erwan.tranvouez.free.fr>

PRINCIPE HTML...

- Objectif initial : être indépendant de tout couple matériel / logiciel ...
 - doit être utilisable quelque soit le matériel (à l'époque – 1993 – on pensait surtout à ordinateur mais avec OS différents : Unix, Dos/Windows, Mac...).
- ... logique car sinon il faudrait développer autant de page web que de version de client ...
- Donc solution la plus simple : fichier texte avec langage dédié
- Langage à base de balise :
 - `<MABALISE mapropriete="valeur">contenu de la balise</MABALISE>`
 - => un navigateur **lit** (puis l'interprète) le texte de haut en bas et de gauche à droite !



QUE VOIT LE NAVIGATEUR ...

- Il utilise le protocole HTTP pour dialoguer avec un serveur web pour en définitive lui demander une page web ...
- ... càd un fichier texte (=très longue chaîne de caractères) contenant du « code » HTML
- => un navigateur lit le texte de haut en bas, ligne par ligne et de gauche à droite !
 - Il sait que le langage comprends des balises (tout ce qui n'est pas compris entre balise sera affiché tel quel ...)
 - `<MABALISE mapropriete="valeur">contenu de la balise</MABALISE>`
 - `<` tiens une balise ?
 - `<MABALISE>` : donc la balise est MABALISE ... je connais ? (sinon je n'affiche pas)
 - `</MABALISE>` : ok donc fin de la balise ...
 - Il construit un arbre (explication tableau) et associe les valeurs et les propriétés aux nœuds de l'arbre ... (we'll get back to this)
 - Et applique ensuite la mise en forme adéquate ...



REGLES D'ECRITURES ...

- Les balises à l'intérieur d'autres balises sont des « enfants » de ces balises => définit une hiérarchie ...
- La première balise étant : **HTML**
- On ferme les balises dans l'ordre qu'on les a ouvert :
 - **<AA>** dkfslckf **<BBB>** sldkf **</BBB>** lkdk **</AA>**
 - **No overlapping**¹ (**<a>**dssdf****dsfsd********)
 
 - Lorsqu'une balise existe sans contenu, on peut la fermer en 1 fois : **
** (évite un **
</BR>**) BR = saut de ligne / break line
- Commentaires encadrés par **<!--** et terminé par **-->** (fonctionne sur plusieurs lignes)
 - apparaissent dans le code HTML (donc lisible)... alors éviter **<!--** mot de passe root est mot2passe mais chut **-->**

¹ = pas de chevauchement



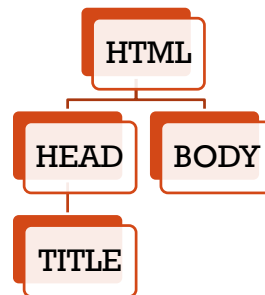
REGLES D'ECRITURES

- Une balise peut se voir associer des propriétés (attributs) séparées par un espace, déclarées a l'intérieur de la balise ouvrante (la 1^{ère})
- Ce n'est pas vous qui décidez ! Elles sont prédéfinies dans le langage HTML (sinon, il ne saurait pas quoi en faire)
- `<MABALISE prop1="10" prop2="rouge" prop3="il était un petit homme...">` valeur de la balise `</MABALISE>`



PREMIERES BALISES (dans l'ordre)

NOM BALISE	Explication
HTML	Déclare que le fichier contient ... du HTML
HEAD	Entête, contient des infos sur le contenu, les informations de styles (CSS, we'll see that later), les mots clefs de la page (pour les moteurs de recherche), du code Javascript (also later), et toute sorte de choses
TITLE	À l'intérieur du HEAD, titre de la page web qui sera affiché dans l'onglet du navigateur
BODY	Le corps du document HTML càd le contenu à afficher dans le navigateur



PREMIERES BALISES (dans le BODY)

NOM BALISE	Explication
H1 ... H6	Titre (Heading) de niveau 1..6 par défaut applique un style différent du texte (taille de texte décroissante)
P	Paragraphe ... 2 paragraphes seront séparés par un saut de ligne
BR	Saut de ligne (line BR eak). Balise sans contenu donc plutôt d'écrire <code>
</BR></code> il est possible d'écrire <code>
</code>
HR	Ligne horizontale (joli) en HTML 5 a un sens supplémentaire (sémantique) soit un changement de thématique (on parle d'autre chose après). Idem que BR peut s'écrire <code><HR/></code> (Horizontal Rule)
A	LA (the) balise internet (Anchor): utilisée pour faire référence à un lien internet (local ou externe), pour marquer l'endroit (pour être utilisé en suite par un autre A...) grâce à l'attribut HREF ...
UL, OL	Pour indiquer une liste de valeurs : ordonnée (Ordered List) ou non (UnOrdered List). Et on peut jouer sur la forme des puces ... via l'attribut type ('disc', 'square'...)
LI	Chaque valeur (List Item ?) indiquée par LI



PREMIÈRE PAGE HTML : CODE SOURCE

```

<HTML>
  <HEAD>
    <TITLE>Ceci est ma premièrè page</TITLE>
  </HEAD>
  <BODY>
    <H1>Ceci est un titre de haut niveau</H1>
    <P>Petit paragraphe qui
    est sur
    plusieurs lignes
    mais cela se verra-t'il dans le navigateur ?</P>
    <P> 2e paragraphe</P>
    <H2>2e titre</H2>
    <P>Voici un exemple de liste à puces</P>
    <UL type="Square">
      <LI>Baa baa Black sheep</LI>
      <LI>have you any wool ? </LI>
      <LI>yes sir, yes sir 3 bags full</LI>
    </UL>
  </BODY>
</HTML>

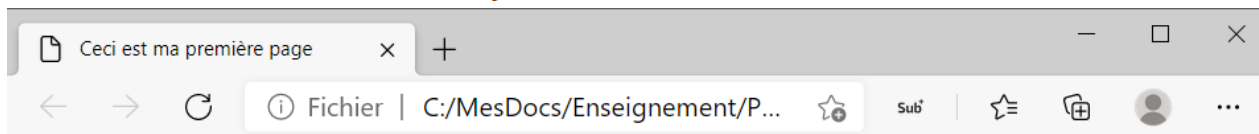
```

Les accents (é, è, à...) peuvent poser des problèmes d'affichage selon l'encodage par défaut du navigateur ... nous verrons + tard comment l'éviter...



RÉSULTAT

<Title> !!



Ceci est un titre de haut niveau

<H1>

Petit paragraphe qui est sur plusieurs ligne mais cela se verra-t'il dans le navigateur ?

2e paragraphe

2e titre

<H2>

<P> multiligne DANS le code source ... mais pas pris en compte par navigateur ... alors pourquoi faire ? Pour avoir code HTML lisible ... pour l'être humain

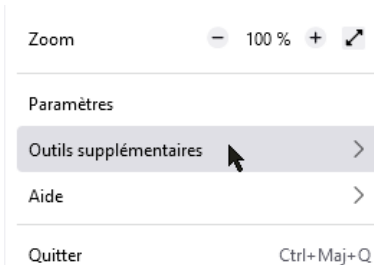
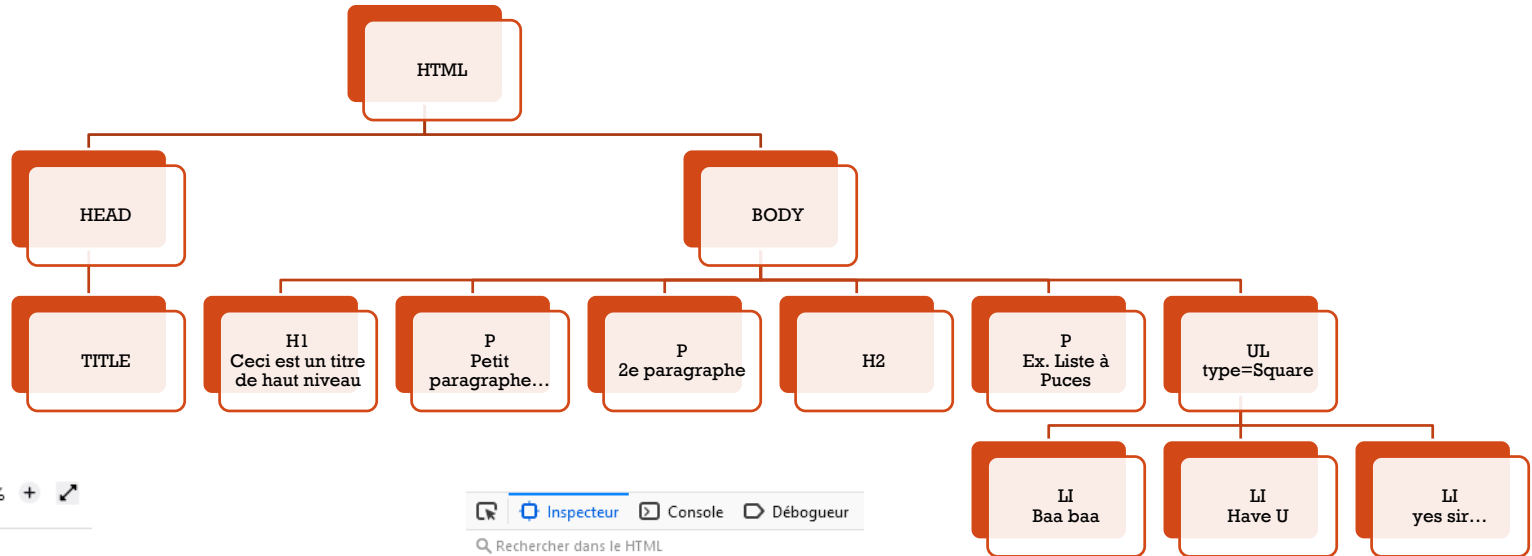
Voici un exemple de liste à puces

- Baa baa Black sheep
- have you any wool ?
- yes sir, yes sir 3 bags full

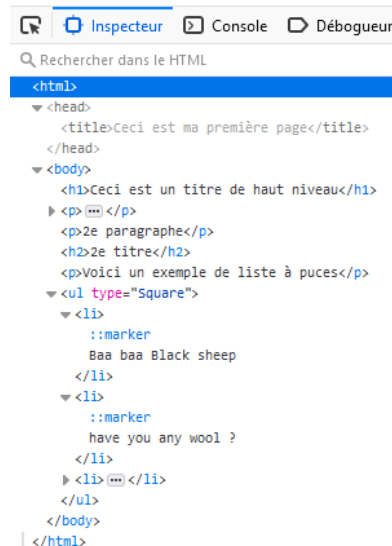
Type="Square"!!



PREMIÈRE PAGE HTML : DOM/HIÉRARCHIE



Aperçu dans
Navigateur web



ADRESSE RELATIVE/ABSOLUE

- `<A href="adresse">Texte qui sera affiché par navigateur et cliquable (lien)</a>`
- **adresse absolue :**  Texte qui sera affiché par navigateur et cliquable (lien)
 - IRL¹: j'habite 15 rue du chien qui pleure, 13001 LOTREVILLE
 - Web : <http://www.monsite.fr/peip/index.html> (ou https://)
- **adresse relative :**
 - Relative par rapport à quoi ? Relative à la page qu'on consulte presently (puisque un lien hypertexte se trouve forcément... dans une page web!):
 - Si la page [index.html](#) précédente contient un lien relatif le navigateur reconstruira l'adresse à partir de celle d'[index.html](#)
 - Ce qui donne :
 - IRL¹: 3^{ème} maison après celle de Gérard (on connaît l'adresse de Gérard)
 - Web : `<p>Cliquer <a href="info.html">ici</a>!</p>`  Cliquer [ici](#)!
 - Navigateur fera ²: <http://www.monsite.fr/peip/> + [info.html](#)
=> <http://www.monsite.fr/peip/info.html>

¹ IRL = In Real Life (dans la vie réelle #TOEIC)

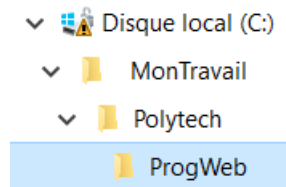
² lien visible en bas à gauche de la fenêtre du navigateur



ADRESSE RELATIVE/ABSOLUE...

CONTEXTE LOCAL ET DISTANT ?

- et histoire de mélanger tout : pourquoi ça marche chez moi et pas chez Gérard?
 - Vous avez zippé vos fichiers, envoyé par mail à Gérard et quand Gérard le dezippe chez lui, les liens ne fonctionnent pas !!!
 - Sans doute vos liens contiennent-ils des adresses absolues ... locales! C'est quelque chose comme :
file:///C:/MonTravail/Polytech/ProgWeb/MaPageWeb.html
- **Protocole**¹ **LettreLecteur** **chemin_repertoire/** fichier
- Or, quand Gérard clique sur le lien, le navigateur cherche le fichier MaPageWeb.html dans le répertoire ProgWeb qui n'existe pas chez Gérard ! (lui il a C:/MonTravail/Web...)
- => utiliser adresses relatives !

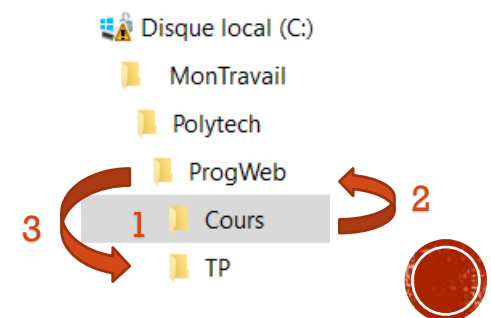


¹ suivant navigateur, protocole masqué :

ADRESSE RELATIVE/ABSOLUE...

CONTEXTE LOCAL ET DISTANT ?

- Finally,
 - adresse absolue sans contexte :
 - Utilisation de / comme 1^{er} caractère ex : /MonTravail... (avec file://¹ ou http://)
 - Signifie : whatever protocole you're using, start from root directory and follow the path...
 - Adresse relative avec navigation dans les répertoires...
 - Dans le fichier [mesCours.html](#) figure un lien vers les tp... [mesCours.html](#) se trouve dans le répertoire [Cours](#)... et les tp ... dans le répertoire [TP](#)
 - Adresse relative = *tu vois où tu m'as trouvé (répertoire Cours [1]), ben tu remonte d'un niveau [2] et tu prends le répertoire TP [3] ...*
 - Soit [../TP/MesTP.html](#) (.. = remonte d'un niveau)
 - Donc si on est dans Cours, [../../../../](#) fait référence au répertoire ... [MonTravail](#)



¹ à ne pas confondre avec ftp:// (ie file transfert protocol)

LIEN À L'INTERIEUR D'UNE PAGE WEB :

 & ID...

- On peut également avoir des références « internes » à la page ... lorsque la page est « longue », que l'on veut accéder directement à une partie (type table matière)
 - **Principe** : on crée des sortes de signets et on pointe vers les signets
 - Une balise peut être identifiée avec l'attribut ID
 - `<H1 id="H1">Titre de niveau 1</H1>`
 - Pour faire référence à H1, rendre possible d'aller vers cette zone, on peut utiliser une adresse « interne » à la page avec # (se dit dièse pas hashtag #toutnestpasuntweet)
 - `<A href="#H1">cliquer pour aller directement sur H1</A>`
- Exemple Wikidepia : [Second degré — Wikipédia \(wikipedia.org\)](https://fr.wikipedia.org/wiki/Second_degré)



LIEN À L'INTERIEUR D'UN FICHIER : & ID...

- Exemple Wikipedia : [Second degré — Wikipédia \(wikipedia.org\)](https://fr.wikipedia.org/wiki/Second_degr%C3%A9)

- Ladite page contient l'équivalent d'un renvoi vers une note de bas de page ¹ non prémédité.

- Le code HTML correspondant est :

... non prémédité^{1}.

Le 1 est encadré par un lien (<a>) pointant vers un « signet local #cite_note-1 : donc lorsqu'on cliquera sur ¹, le navigateur nous « déplacera » vers l'élément HTML identifié par #cite_note-1 ...

Càd

Notes et références [modifier | modifier le code]

1. ↑ http://www.educaloi.qc.ca/loi/contrevenants_et_accuses/330/ [archive]

^{xxx} mets xxx sous la forme d'un exposant ¹

La cible est en fait un item d'une liste à puce ordonnée identifié par cite_note-1

Dont le code source est :

```
<h2>Notes et références</h2><li id="cite_note-1"><a href="#cite_ref-1">↑</a><a href="lien vers l'article externe">lien externe</a></li>
```

Mais si vous êtes curieux, se vous avez noté qu'il y avait un href pointant sur #cite_ref-1 qui est en fait l'id de la balise sup !!! ¹ ↑

Donc en cliquant sur la flèche ↑ à côté du numéro d'item (1) on revient sur la zone ayant cité ce point (pratique!)



TABLEAU HTML ...

- « Dessiner » un tableau en HTML est un peu tordu ...
- Un tableau HTML est composé de lignes (rows)
- Une ligne est composée de cellules contenant des données
- Donc les lignes de tableau (table row = tr) et des cellules contenant des données (table data = td)
- Donc le tableau suivant donnerait en HTML :

```
<tr> ... </tr>
```

```
<tr> ... </tr>
```

```
<tr> ... </tr>
```


Autant de TR que de lignes ...



TABLEAU HTML ...

Autant de TD que de colonnes...

```
<tr><td>La...</td><td> ...</td><td> ...</td> </tr>
```

```
<tr><td> ...</td><td>Lala</td><td> ...</td> </tr>
```

```
<tr><td> ...</td><td> ...</td><td>La, la !</td> </tr>
```

On comprends qu'une mise en page rend le code source plus facile à lire (indentation) :

```
<tr>
```

```
    <td>La...</td>
```

```
    <td>...
```



La...		
	Lala	
		La, la !

La...		
	Lala	
		La, La!



TABLEAU HTML ... +JOLI

- Pour modifier l'apparence (en attendant CSS), on peut modifier les valeurs des attributs du tableau, des lignes ou des cellules ...
- `table`: [HTML5 Reference \(w3.org\)](http://www.w3.org/html5/reference)
 - `bgcolor` : background color, modifie... la couleur de fond du tableau (donc du fond de toutes les lignes et toutes les colonnes)
 - `cellspacing` : espace autour de la cellule (on y reviendra ds CSS)
 - `border` : épaisseur de la bordure (0 par défaut, that's why no border are shown when not specified... I had to use Firefox to show it)
 - `Width` : largeur....
- on peut utiliser certaines propriétés sur les lignes (`tr`) et les cellules (`td`)...
- Mais cela a l'inconvénient de mélanger contenu et mise en forme, et le validator (*spoiler*) lui-même conseille d'utiliser CSS plutôt que modifier ces attributs en HTML.



TABLEAU HTML ... +JOLI

- Pour finir, quelques éléments :
 - On peut rajouter une légende au tableau : `<caption>` qui doit se trouver après `<table>`
 - On peut rajouter une 1^{ère} ligne d'entête du tableau (et avoir un style différent) `<th>` qui fonctionne comme un `<tr>`
 - On peut fusionner des cellules ...
- Fusion des cellules :
 - Exemple 1 :
 - on veut que les cellules 2.2 & 2.3 soient fusionnées. En HTML on traduit ça par : on veut que la cellule 2.2 s'étende sur 2 « colonnes »... (qu'elle occupe l'espace de 2 cellules sur 2 colonnes)
 - => utilisation de l'attribut `colspan = "n"` avec `n = nb cellules` (`colspan = colonne` qui s'étends sur ...)
 - `<tr><td></td><td colspan=2>Lala</td></tr>`
 - Ne pas oublier d'enlever un td ! (doit en avoir 2 in all)

La...		1.3
2.1	Lala	
	3.2	La, la !



TABLEAU HTML ... +JOLI

- Fusion des cellules :

- Exemple 2 :

- on veut que les cellules 1.1, 1.2 & 1.3 soient fusionnées. En HTML on traduit ça par : on veut que la cellule 1.1 s'étende sur 3 « lignes »... (qu'elle occupe l'espace de 3 cellules sur 3 colonnes)
 - => utilisation de l'attribut `rowspan = "n"` avec `n = nb cellules` (`rowspan = ligne` qui s'étend sur ...)
 - `<tr><td rowspan=3>La...</td><td></td><td></td></tr>`
 - Ne pas oublier d'enlever un td par ligne pour les lignes concernées ! (doit en avoir 2 par ligne in all)

La...		
	Lala	
		La, la !

- Résumé => [UnPeu2Tout.html](#) sur Amétice!



COMMENT VÉRIFIER SON CODE HTML ?

- En ouvrant le fichier avec un navigateur web ! Et on vérifie si ce qu'on voit est bien ce qu'on voulait obtenir !
- Mais aussi, utiliser le [validateur HTML du W3C](http://validator.w3.org) (World Wide Web – 3W donc – Consortium...)
 - Validator.w3.org
- On peut :
 - Indiquer une adresse internet (Validate by URI)
 - Téléverser un fichier HTML (validate by File Upload)
 - Copier coller le code HTML (validate by Direct Input)
- Exemple soumission de `http://erwan.tranvouez.free.fr/cours/polytechPeiP.html` donne : 8 warning et 46 erreurs (allégorie du cordonnier ...)

Error Element `table` not allowed as child of element `ul` in this context. (Suppressing further errors from this subtree.)

From line 237, column 1; to line 237, column 7

```
> <br> <br> <br> <table> <br> <t
```

