

## (X)HTML

*Le but n'est pas ici d'avoir une maîtrise complète de (x)html mais bien d'être suffisamment à l'aise avec ce langage que pour pouvoir écrire les pages dont nous aurons besoin dans le développement des applications web. Nous voyons les bases du protocole, les balises les plus courantes et comment écrire des pages propres (nous verrons CSS par la suite). On voit aussi comment « valider » nos pages.*

## HTTP (HyperText Transfer Protocol)

---

Protocole de communication utilisé dans la communication entre un client Web (généralement un browser<sup>1</sup>) et un serveur Web dans une architecture Client/Serveur.

Le serveur Web est en écoute sur un port, généralement le 80.

Le client et le serveur échangent des données sous forme textuelle au travers de requêtes-réponses. Le protocole http est dit non-orienté session (on parle de protocole sans mémoire) dans la mesure où la connexion entre le client et le serveur est coupée après chaque paire requête-réponse. La notion de session devra, si elle est nécessaire, être implémentée au-dessus du protocole http.

### Requête:

Le client adresse ses requêtes au serveur suivant l'une ou l'autre 'Méthode' (Get, Post<sup>2</sup>, Put, ...) en précisant l'URL<sup>3</sup> de la ressource demandée, la version du protocole et un ensemble d'informations facultatives (browser utilisé, jeu de caractères, encoding, ...).

### Réponse:

Le serveur retourne sa réponse constituée de la version du protocole, du code de réponse<sup>4</sup> accompagné du texte explicatif du code ainsi que, généralement, du document demandé.

Exemple: Encodons dans notre browser favori: <http://www.heb.be>

La requête générée aura la forme:

```
GET / HTTP/1.1
Host: www.heb.be
User-Agent: Mozilla/5.0
...
```

- 
- 1 Navigateur, butineur, ... Client HTTP, dont les exemples les plus connus sont Internet Explorer, Mozilla Firefox, ...
  - 2 La méthode 'Post' envoie des données (paramètres de requêtes) dans le corps de la requête, la méthode 'Get' envoie ces paramètres dans l'url.
  - 3 URL: *Uniform resource Locator* chaîne de caractère qui localise une ressource Web (à ne pas confondre avec uri *Uniform resource Identifier* qui identifie une ressource Web ou avec urn *Uniform resource Name* qui identifie une ressource Web à l'intérieur d'un espace de nom).
  - 4 1xx: information, 2xx: succès, 3xx: redirection, 4xx: requête incorrecte, 5x: erreur interne au serveur

et la réponse du serveur présentera l'en-tête:

```
Content-Length: 6008
Content-Type: text/html
Content-Location: http://www.heb.be/Index.html
Last-Modified: Fri, 12 Oct 2007 08:42:38 GMT
Accept-Ranges: bytes
Etag: "985864d8abcc81:994"
Server: Microsoft-IIS/6.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
Date: Thu, 17 Jan 2008 13:49:39 GMT

200 OK
```

et sera suivie du document lui-même.

## Remarques :

- HTTP a vu différentes versions 0.9, 1.0, 1.1.
- HTTPS est une version sécurisée de http utilisant généralement le port 443.
- Certaines requêtes, utilisant la méthode GET, peuvent passer des paramètres de requête dans l'url. Pour le visualiser, faites une recherche quelconque sous Google par exemple.
- Remarquons que l'échange est décrit ici sans se préoccuper de ce que devra effectivement réaliser le serveur: peut être n'aura-t-il qu'à récupérer la page demandée et la transférer au client (page statique), peut être devra-t-il la composer en faisant appel à certains programmes et/ou d'autres serveurs (page dynamique).

## Langages de balisage

### Les langages de balisage

Les **langages de balisage** (sous-classe des langages de description) représentent une classe de langages spécialisés dans l'enrichissement d'information textuelle. Ils opèrent grâce aux **balises** (tags), unités sémantiques délimitant chacune un ensemble à l'intérieur d'un fichier texte, souvent en unicode. (Wikipedia)

Par exemple :

```
<str1> Mon texte de 'Str1' <str12> sous structure </str12> <str12> autre sous structure sur le
même pied que le précédent </str12></str1>
```

qu'il sera préférable de présenter sous la forme :

```
<str1> Mon texte de 'str1'
  <str12> sous structure </str12>
  <str12> autre sous structure sur le même pied que le précédent </str12>
</str1>
```

mais cela ne changera en rien la sémantique de l'information.

Les portées des balises ne peuvent jamais se chevaucher.

En pratique, un document écrit dans un langage de balisage représente un arbre où les noeuds sont définis par les paires de balise ouvrante/fermante et les valeurs des noeuds comme les informations textuelles reprises entre ces balises.

## SGML

(*Standard Generalized Markup Language* (langage normalisé de balisage généralisé)) : langage permettant de définir des langages de balisage. Chaque langage de balisage défini en SGML est appelé *application SGML*. Ce langage est normalisé par ISO. Nous n'en dirons pas plus mais il faut savoir que HTML est une application SGML.

## HTML

(HyperText Markup Language) application de SGML destinée à produire des documents (pages) Web.

## XHTML

(Extensible HyperText Markup Language) successeur de HTML, application de XML.

## XML

(Extensible Markup Language) langage de définition de langages de balisage et de structuration de données ayant notamment pour but de faciliter l'échange de données entre systèmes hétérogènes.

## [X]HTML

---

Nous allons aborder HTML, très proche de XHTML. HTML et XHTML sont les langages de publication du World Wide Web.

En pratique, HTML devrait être un langage utilisé uniquement pour exprimer la structuration d'information mais il permet, même si cela est largement déconseillé, de définir la présentation du document. Exprimer directement la présentation sans exprimer séparément la structuration a plusieurs effets pervers que nous aborderons ultérieurement. De plus, les éléments propres à la présentation peuvent être interprétés différemment d'un browser à l'autre.

Une exemple de base permettant de visualiser les caractéristiques suivantes<sup>1</sup> :

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<!-- évidemment je n'oublie pas de placer des commentaires -->
<title>HTML Test</title>
</head>

<body>
    Ceci est mon premier test HTML! <br> Élégant non ?

    Ca va marcher!
</body>
</html>
```

---

<sup>1</sup> Il vous suffit, lorsque vous visualiser un document html via un browser, de faire 'View source' pour visualiser le code html reçu par votre navigateur.

- Un ensemble de ‘blancs’ est considéré comme un seul caractère!
- Les caractères spéciaux seront interprétés différemment suivant l’encoding pris en compte (ici, ISO-8859-1 qui est l’option par défaut). Il est possible de spécifier en quel type d’encodage est utilisé pour le document.
- Les balises sont écrites en minuscule<sup>1</sup>.
- Nous retrouvons des **éléments** (contenu représenté entre la balise ouvrante et la balise fermante) mais aussi des **attributs** présentés dans la balise ouvrante de l’élément. Les attributs sont généralement cités sous la forme clé=valeur où la valeur est fournie entre guillemets (ou apostrophes). Un élément vide peut être exprimé en n’utilisant qu’un seul tag : <aa></aa> revient à <aa />.
- La plupart des tags doivent être appariés (la plupart des browsers développent cependant une tolérance aux erreurs) mais certains tags ne définissent jamais de contenu et ne demandent donc pas de balise fermante (cf. <br> ) mais il est actuellement conseillé d’utiliser la syntaxe liée aux éléments vides (<br />).

## Les caractères

Les caractères peuvent être encodés par saisie classique et/ou au travers d’une codification débutant par & et se terminant par ; (cf. [http://www.w3schools.com/tags/ref\\_entities.asp](http://www.w3schools.com/tags/ref_entities.asp))

Cette codification devra être utilisée dans les cas où le caractère a une signification particulière pour html.

Caractère	Code Iso	Représentation html
"	&#34;	&quot;
&	&#38;	&amp;
<	&#139;	&lt;
>	&#155;	&gt;

Mais encore

^	&#160 ;	&nbsp;
---	---------	--------

## Structure d’un document html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>...</HEAD>
  <BODY>Contenu de la page</BODY>
</HTML>
```

La première ligne constitue le prologue et présente la version de html utilisée.

<sup>1</sup> Malgré cette recommandation rarement suivie, HTML est insensible à la casse.

## Quelques balises

Il est nécessaire de bien percevoir la distinction – parfois floue – entre les éléments et attributs de structuration et ceux de présentation.

Parmi les éléments suivants lesquels sont de structuration et lesquels de présentation?

<code>&lt;b&gt;</code>	bold	<code>&lt;del&gt;</code>	texte barré
<code>&lt;big&gt;</code>	police plus grande	<code>&lt;sub&gt;</code>	indice
<code>&lt;blink&gt;</code>	clignotant	<code>&lt;sup&gt;</code>	exposant
<code>&lt;code&gt;</code>	code	<code>&lt;h1&gt;...&lt;h6&gt;</code>	niveau de titre
<code>&lt;em&gt;</code>	emphase	<code>&lt;p&gt;</code>	paragraphe
<code>&lt;i&gt;</code>	italique	<code>&lt;br&gt;</code>	retour à la ligne

## Quelques attributs

Tous les attributs ne peuvent pas être utilisés avec tous les tags.

<code>align</code>	avec comme valeurs possibles : LEFT, RIGHT, CENTER, JUSTIFY
<code>nowrap</code>	interdit le passage à la ligne
<code>id</code>	donne une identification à la balise (unique au travers du document)
<code>class</code>	assigne une classe à l'élément
...	

Nous ne pouvons passer en revue l'ensemble des balises et des attributs, donnons comme référence : <http://www.w3schools.com/html/default.asp>

## Les listes

Html propose trois type de listes; citons les deux plus fréquents :

- La liste non ordonnée :

```
<ul>
  <li>premier</li>
  <li>second</li>
</ul>
```

- La liste ordonnée :

```
<ol>
  <li>premier</li>
  <li>second</li>
</ol>
```

Même si ces tags génèrent une présentation, il s'agit de toute évidence d'outils de structuration.

## Les liens

```
<a href="url"> texte </a>
```

où url peut avoir la forme `http://...`, `mailto :.....`, `ftp://...`, `#lienInterne`

```
<a name="lienInterne" >...</a>
```

## Les tableaux

A nouveau, il s'agit d'un ensemble de tags mêlant la notion de structure et de présentation.

## Les meta-tags

Les méta-tags permettent de décrire le document et sera notamment utilisé par les moteurs de recherche pour leur référencement<sup>1</sup>. On y retrouve notamment : Author, copyright, subject, ...

## Autres tags

De nombreux autres tags existent : ceux liés aux frames<sup>2</sup>, aux formulaires, ... Nous vous laissons les découvrir vous-même ou nous les présenterons au fur et à mesure des besoins.

## Références

De nombreuses références et tutoriaux existent sur le web. Nous vous proposons de prendre comme référence commune : <http://www.w3schools.com/html>

## Validation

Chaque page html doit débuter par < !DOCTYPE ..... > qui permet de préciser la version de html ou de xhtml mise en œuvre dans votre document. La présence de ce ‘Doctype’ vous assurera de l’interprétation conforme à la norme de votre document.

Les différents ‘Doctypes’ :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

**Strict:** limite la syntaxe html à la version précisée et interdit l’utilisation de particularités obsolètes

**Transitional** : permet de ne pas se limiter à la syntaxe stricte et permet de rester compatible avec des navigateurs ‘anciens’

**Frameset** : lié à l’utilisation de frames.

Pour vous assurer que vous avez bien respecté les contraintes imposées par votre déclaration, il vous est fortement conseillé de **valider** votre document. Pour ce faire, vous pouvez vous rendre à la page : <http://validator.w3.org>

## Editeur

Nous nous contenterons, dans le cadre de ce cours et des exercices d’utiliser un banal éditeur de textes mais il existe de nombreux éditeurs facilitant la composition de document html (aide à la saisie, validations, prévisualisation, ...)

---

1 Référencement: faire prendre en compte de documents, au travers de mots-clés, par des moteurs de recherche.

2 Frame: cadre

## Exercices

---

- 1) Retrouvez les mots-clé de référencement de la page d'accueil du site de la heb.
- 2) Produisez une page de présentation du cours de ALG2, reprenant l'information de la page correspondante du site de MCD (vous pouvez vous dispenser de faire apparaître le menu). Ajoutez un lien vers la véritable page.
- 3) Produisez une page, qui pourrait faire partie du même site, présentant un formulaire demandant nom, prénom, adresse mail, section, année, sexe de l'étudiant de l'Esi désirant être averti de l'évolution du site. Veillez à ce que cette page soit esthétiquement cohérente avec la précédente et utilise les composants html les plus adéquats.
- 4) Visualisez ces pages au travers de divers navigateurs pour observer que les rendus ne sont pas nécessairement identiques.