

# HTML

## Tp 1

### Une première approche des balises

Florent DEVIN

19 septembre 2008

## Préambule

Avant de commencer le TP, dans votre répertoire `Info`, créez un répertoire `Html`. Dans ce dernier créez un répertoire `Tp1`. Vous travaillerez dans celui-ci.

Il existe plusieurs façons de créer des pages HTML. La plus facile est d'utiliser un logiciel qui crée le code HTML pour vous. Nous n'utiliserons pas cette façon. Nous allons créer les pages HTML grâce à un éditeur de texte classique  $\Leftrightarrow$  *Xemacs*.

## 1 Présentation

Qu'est ce qu'HTML ? Avant de répondre à cette question, il faut d'abord répondre très brièvement<sup>1</sup> à la question qu'est ce qu'Internet<sup>2</sup> ? Même si la réponse semble évidente, elle mérite d'être. Internet désigne un ensemble de réseaux<sup>3</sup> et de machines connectées les uns aux autres. Cette réponse permet de mieux comprendre les problèmes qui se posent.

Tout d'abord, il existe une diversité énorme de matériel, en terme de processeur, carte graphique, disque, ... Cette diversité doit être capable de communiquer. Si la communication, et le bon fonctionnement au sein de la même machine est du ressort du système d'exploitation. La communication inter-machine fait intervenir plusieurs systèmes. Il a donc fallu créer un protocole de communication, puis s'accorder sur plusieurs codages pour les données. Parmi tous les formats disponibles pour communiquer, il existe une catégorie : les *Markup Language*. Le HTML<sup>4</sup> fait parti de cette catégorie.

---

<sup>1</sup>La brièveté des explications entraînent forcément quelques imprécisions

<sup>2</sup>On désigne aussi internet par le mot Web

<sup>3</sup>Un réseau est un ensemble de machine

<sup>4</sup>HyperText Markup Language

C'est en fait une simplification du langage SGML<sup>5</sup>. Vous découvrir d'autres langages de type ML dans la suite de vos études.

Avant d'entrer dans le vif du sujet, situons un peu le contexte d'Internet. Lors de sa création son usage était avant tout destiné à l'échange d'informations entre scientifiques. Ils se mirent "d'accord" sur un format, le HTML. Ce format était très austère. La convivialité d'internet a permis son essor. Une population de non scientifique a commencé à utiliser le web. Ces derniers, trouvant le support un peu trop "simpliste", ont essayé de mettre en forme leur texte. Des "designers" ont alors lancé plusieurs pistes. Le langage a alors évolué pour supporter ces nouveaux besoins. Jusqu'à l'arrivée de HTML 4.0.

Parallèlement un organisme a été créé pour réfléchir sur l'évolution possible du Web : W3C<sup>6</sup>. La gestion de cet organisme est assurée par le MIT, l'ERCIM et Keio. Il existe à l'heure actuelle 437 membres de ce consortium, parmi lesquels on trouve : Google, l'INRIA, AT&T, IBM, Mozilla, ... La tendance actuelle est de séparer le fond de la forme, un peu à la manière de L<sup>A</sup>T<sub>E</sub>X. Le fond sera géré par un langage à balise<sup>7</sup>, et la forme par un autre langage. C'est XHTML 1.0 qui doit être utilisé pour éditer le fond de votre document. Les CSS<sup>8</sup> permettront de mettre en forme le document final.

Actuellement, le W3C travaille, entre autre, sur le XHTML 2. Cette norme permettra d'intégrer une couche sémantique au web. Dans le cadre de ce cours, nous utiliserons la norme XHTML 1.0. Ce dernier doit remplacer progressivement le HTML 4.01. Il est plus simple, et plus strict que son prédécesseur. Un document XHTML décrit une page Web. Cette page Web doit par la suite être affichée par les différents navigateurs. La description d'une page Web se fait en utilisant les éléments du langage XHTML. Ces éléments ont pour unique vocation de décrire la *structure* du document, la mise en forme se faisant plus tard. L'interprétation des différents éléments est à la charge du navigateur qui souhaite visualiser une page. Il peut donc y avoir quelques différences de visualisation selon le navigateur.

Un élément encadre son contenu par des balises. Il existe deux formes de balises :

- `<balise>contenu</balise>` : cette forme est utilisée lorsque la balise doit agir sur un contenu précis, et affecte les caractéristiques de ce dernier. On écrira par exemple : `<title>Une première approche des balises</title>`, ce qui permet de spécifier un titre. Bien évidemment, tout élément ouvert doit être fermé. De plus vous devez veiller à la bonne imbrication des éléments  $\Leftrightarrow$  `<p><em>test</em></p>` et non pas `<p><em>test</p></em>`.
- `<balise/>` : cette forme est utilisée lorsque la balise n'agit pas sur un contenu. On écrira par exemple `<img src="" alt="" />`

Toutes les balises doivent être écrites en minuscules. Les paramètres

---

<sup>5</sup>Standard Generalized Markup Language

<sup>6</sup>WorldWideWeb Consortium

<sup>7</sup>C'est le cas pour les langages de type ML

<sup>8</sup>Cascading Style Sheet

éventuels de celles-ci doivent être encadré par des ". Enfin un document XHTML commence toujours par déclarer sa *DTD*<sup>9</sup>, suivi de la balise `html`, qui englobe la page. La définition de la DTD se fait via le texte suivant ou autre équivalent :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "xhtml1-strict.dtd">
```

**Question 1 :** À votre avis ce texte fait-il parti de la norme XHTML ? Justifiez.

L'élément `html` doit contenir au minimum les éléments `head` et `body`. L'élément `head` doit contenir au minimum l'élément `title`. Ce qui donne comme page minimum :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN
  "xhtml1-strict.dtd">
<html>
  <head>
    <title>Titre du document</title>
  </head>
  <body>
  </body>
</html>
```

Listing 1 – "Page minimum"

**Question 2 :** Enregistrer ce document dans un fichier que vous appellerez `ex1.html`. Puis tapez la commande : `firefox ex1.html`. Que constatez vous ?

Lorsque vous rédigez un document, vous spécifier le titre du document, puis des sections, sous-section, ..., des paragraphes. C'est dans ces derniers que vous mettez votre texte. Il en va de même en HTML. La section de plus haut niveau est l'élément `h1`, puis `h2`, ..., `h6`. L'élément `p` sert quand à lui pour indiquer que l'on est en train d'écrire un paragraphe.

**Question 3 :** Créez une page HTML, ayant pour titre : "*Seconde page Web*", et comme texte dans cette page : "*terminé*"

Le W3C met à notre disposition un programme qui permet de vérifier la validité de la page que l'on vient d'écrire. Vérifier que la page que nous venons d'écrire est valide n'est pas suffisant pour que la page soit correcte. C'est une étape. Un peu à la manière d'un algorithme, où vous respectez d'abord la syntaxe. Le respect de la syntaxe n'entraîne pas un algorithme valide. C'est exactement la même chose pour une page Web. La validation du W3C indique que vous avez correctement respecté la syntaxe. Cela n'indique en aucun cas que vous avez respecté la sémantique des balises.

Pour vous permettre d'utiliser le valideur du W3C, il y a plusieurs possibilités. Soit votre page est accessible depuis internet, soit elle ne l'est pas.

---

<sup>9</sup>Document Type Definition

Dans ce dernier cas<sup>10</sup>, on pourrait télécharger un module complémentaire de firefox. Nous n'utiliserons pas cette solution. Nous allons plutôt rendre vos pages web accessible depuis l'extérieur.

Chacun d'entre vous possède la possibilité de créer un site web qui sera hébergé à l'école. Pour ce faire dans votre répertoire d'accueil sur votre compte EISTI vous allez créer un répertoire `public.html`. C'est dans ce répertoire que vous allez stocker votre site. Il faut bien évidemment que ce répertoire soit accessible en exécution et en lecture.

**Question 4 :** Copiez la page que vous venez de créer dans votre `public.html`. Dans firefox entrez l'adresse suivante :

```
http://webetu.etude.pau.eisti.fr/~login/fichier.html
```

Si vous voyez afficher votre page, c'est que tout est bon. Chargez la page <http://validator.w3.org/> et validez votre page. Que vous affiche le valideur ? Résolvez les erreurs. Nous traiterons les warnings plus tard.

Une fois ceci fait, vous venez de créer votre première page HTML valide. Notez bien que pour l'instant nous ne nous intéressons pas à la mise en forme. Pour créer une page Web correcte, il faut utiliser les balises en respectant ce qui est imposé par la norme, mais aussi respecter la sémantique des éléments.

Par ailleurs dans le préambule<sup>11</sup> de votre page ajoutez l'élément `meta` suivant :

```
<meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-15" />
```

Ceci permet de définir dans quel format votre fichier est enregistré, et d'indiquer au navigateur d'utiliser le bon format. Bien évidemment si vous enregistrez vos fichiers au format UTF-8, il faudra remplacer ISO-8859-15 par utf-8.

## 2 Quelques balises

La table 1 vous présente les balises qui peuvent vous être utile pour réaliser le TP. Bien évidemment, il existe beaucoup d'autres balises, l'ensemble des balises possibles se trouve à l'adresse suivante :

```
http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/#s_xhtmlmodules
```

Même si d'autres balises peuvent exister, on ne devrait utiliser que les balises citées dans le document précité.

**Question 5 :** Reproduisez la page Html dont vous pouvez voir un aperçu sur la figure 1. Vous n'utiliserez que des balises XHTML strict.

---

<sup>10</sup>Il existe d'autres solutions pour valider votre page

<sup>11</sup>C'est à dire dans l'élément `head`

Éléments	Exemple	Signification
h1 ...h6	<code>&lt;h1&gt;coucou&lt;/h1&gt;</code>	Notion équivalente à section, sous-section, ...de L <sup>A</sup> T <sub>E</sub> X
a	<code>&lt;a href=""&gt;texte du lien&lt;/a&gt;</code> <code>&lt;a name=""&gt;texte du lien&lt;/a&gt;</code>	Déclare un lien hypertexte
ul	<code>&lt;ul&gt;éléments de liste&lt;/ul&gt;</code>	Liste énumérée
ol	<code>&lt;ol&gt;éléments de liste&lt;/ol&gt;</code>	Liste ordonnée
li	<code>&lt;li&gt;élément 1&lt;/li&gt;</code>	Un élément d'une liste
p	<code>&lt;p&gt;bla bla&lt;/p&gt;</code>	Déclare un paragraphe
img	<code>&lt;img src="toto.gif" alt="description"/&gt;</code>	Insertion d'une image
em	<code>&lt;em&gt;toto&lt;/em&gt;</code>	Mettre en évidence
sup	<code>1&lt;sup&gt;er&lt;/sup&gt;</code>	Mise en exposant
sub	<code>x&lt;sub&gt;1&lt;/sub&gt;</code>	Mise en indice
strong	<code>&lt;strong&gt;sémantique&lt;/strong&gt;</code>	Mise en évidence forte
abbr	<code>&lt;abbr title="baccalauréat"&gt;bac&lt;/abbr&gt;</code>	Définition d'une abréviation
acronym	<code>&lt;acronym title="École Internationale des Sciences du Traitement de l'Information"&gt;EISTI&lt;/acronym&gt;</code>	Définition d'un acronyme
dfn	<code>&lt;dfn title="Arbre de représentation..."&gt;ontologie&lt;/dfn&gt;</code>	Définition d'un terme
cite	<code>&lt;cite title="Super cours"&gt;cours&lt;/cite&gt;</code>	Citation d'un ouvrage
q	<code>&lt;q title="Florent Devin"&gt;Il y a des jours, vaudrait mieux que ce soit des nuits&lt;/q&gt;</code>	Citation d'un auteur
code	<code>&lt;code&gt;...&lt;/code&gt;</code>	Exemple de code
samp	<code>&lt;samp&gt;...&lt;/samp&gt;</code>	Exemple de sortie de programme
kdb	<code>&lt;kbd&gt;...&lt;/kbd&gt;</code>	Signale du texte que doit saisir l'utilisateur
var	<code>&lt;var&gt;...&lt;/var&gt;</code>	Signale une variable
hr	<code>&lt;hr/&gt;</code>	Trait de séparation horizontal
blockquote	<code>&lt;blockquote&gt;...&lt;/blockquote&gt;</code>	Bloc de citation
address	<code>&lt;address&gt;...&lt;/address&gt;</code>	Signale une adresse
div	<code>&lt;div&gt;...&lt;/div&gt;</code>	Regroupe des objets avec une mise en forme commune
span	<code>&lt;span&gt;...&lt;/span&gt;</code>	Mise en forme particulière d'un élément

TAB. 1 – Quelques balises

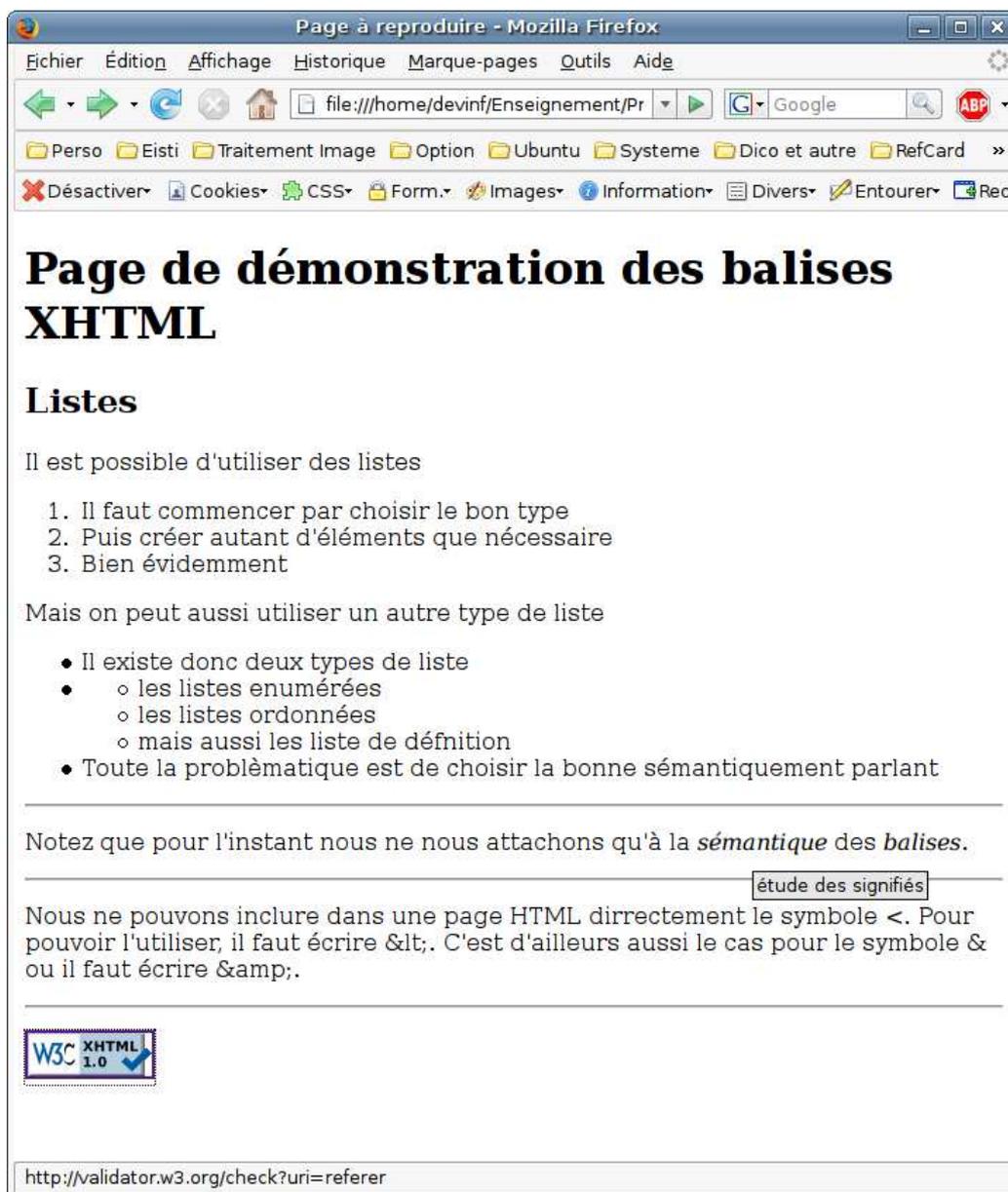


FIG. 1 – Page à reproduire