

Résumé concernant HTML

1. Introduction

1.1. Présentation générale et notion de balise

HTML est le langage utilisé pour définir la structure d'une page web. Il s'agit d'un langage fonctionnant avec des *balises*. Les balises fonctionnent un peu comme des parenthèses typées : une balise doit ouverte doit être fermée, et les balises peuvent être imbriquées, mais pas entrelacées.

On ouvre une balise *toto* avec la syntaxe `<toto>`, et on la ferme avec la syntaxe `</toto>`. HTML n'est pas sensible à la casse pour les noms des balises. Cependant, la convention est d'utiliser des minuscules.

Le concept de langage à balise est maintenant généralisé avec le langage XML.

Avec les années, le rôle d'HTML a changé : au début du Web, HTML donnait la structure et la mise en forme d'une page web. Maintenant, HTML ne doit normalement être utilisé que pour la structure, la mise en forme étant spécifiée par un autre langage, CSS.

Les pages web, en HTML, sont interprétées par les navigateurs web. Ceux-ci sont en général très tolérants aux erreurs de syntaxiques. Aussi, même si on donne ici des principes strictes, beaucoup de pages ne respectant pas ces principes s'afficheront malgré tout correctement. En l'absence de mise en forme spécifiée en CSS, les navigateurs appliqueront une mise en forme spécifique pour chaque balise.

Dans ce document nous ne présentons qu'un sous-ensemble restreint des balises du langage html. Mais cela vous donnera les bases nécessaires pour le cours de Javascript à venir.

1.2. Balise sans contenu

Lorsqu'une balise *toto* ne doit pas inclure un contenu, on utilise la notation `<toto/>`. Cette notation est souvent équivalente à `<toto></toto>`.

1.3. Contenu d'une balise

Le contenu d'une balise sera en général une liste de balises et de textes.

Exemple :

```
<div>
  <p>Un paragraphe.</p>
  <p> Un <span>autre</span> paragraphe</p>
</div>
```

Dans l'exemple ci-dessus :

- la première balise `</p></p>` Contient juste du texte.
- La deuxième balise `<p></p>` contient du texte, puis la balise ``, puis du texte
- La balise `<div></div>` contient du texte (le retour chariot et la tabulation), puis la première balise `<p></p>`, puis du texte encore un retour chariot puis la tabulation), puis la deuxième balise `<p></p>`, et enfin du texte (le dernier retour chariot).

1.4. Attributs

Une balise peut être paramétrée grâce à des attributs. Pour ce faire, dans la balise ouvrante, on rajoutera autant de parties `nomAttribut= "valeur"` que nécessaire.

Exemple :

```
<form method="get">... </form>
```

Certains attributs sont propres à certaines balises, d'autres sont partagés par plusieurs balises. Enfin, certains sont communs à toutes les balises. Notamment, chaque balise peut avoir les 2 attributs suivants :

- `id` : identifiant de la balise. Comme toute identifiant, il ne peut y avoir 2 balises avec le même identifiant dans une page donnée ;
- `class` : liste d'une ou plusieurs classes (à voir comme des « qualificatifs ») associés à la balise. Si on souhaite associer plusieurs classes à un balise, il suffit de séparer les noms des classes par des virgule dans la valeur de l'attribut `class`.

Exemple :

```
<p id="paraph1" class="intro, important">...</p>
```

2. Balises Essentielles

Une page html, dans la version 5 du langage, doit commencer par la ligne :

```
<!DOCTYPE html>
```

Le reste du fichier doit être inclus dans une balise générale, la balise `<html>...</html>`. Le contenu de cette balise est lui-même réparti à l'intérieur de 2 balises :

la balise `<head>... </head>` contient des méta-informations sur le documentation

la balise `<body>... </body>` contient le corps de la page.

Ainsi, une page html aura en général la structure suivante :

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

Un fichier html est donc un arbre de nœuds pouvant être soit des balises, soit du texte.

2.1. Commentaires

Il est possible d'inclure des commentaires dans une page HTML. Les commentaires sont introduits par la séquence de caractères `<!--` et sont fermés par la séquence `-->`.

2.2. Quelques balises de méta-information utiles

Toutes les balises suivantes doivent être mises dans la balise `<head></head>`.

2.2.1. `<title></title>`

Cette balise permet d'associer un titre à la page, titre qui sera en général utilisé par les navigateurs pour nommer l'onglet dans lequel la page sera chargée.

2.2.2. `<link/>`

Balise permettant de faire un lien vers un fichier. Une des utilisations principales de cette balise est celle consistant à préciser le ou les fichiers css définissant la mise en forme d'une page via leur URL¹ absolue ou relative.

Exemple :

```
<link rel="stylecheet" href="style.css"/>
```

Dans l'exemple précédent, *style.css* est le nom du fichier css à utiliser pour la page.

2.2.3. `<script></script>` ou `<script/>`

Cette balise sert à inclure du code javascript dans la page html. Voir la suite du cours pour son utilisation.

2.2.4. `<meta/>`

Cette balise sert pour toute meta-information pour lesquelles ils n'existe pas de balise spécifique, comme par exemple pour préciser l'encodage des caractères utilisés dans la page.

Exemple :

```
<meta charset="utf-8"/>
```

2.3. Quelques balises de contenu essentielles

2.3.1. Balises de titre

`<h1>... </h1>`, `<h2>... </h2>`, ... `<h6>... </h6>` sont des balises permettant de définir des titres dans la page. Plus le chiffre est faible, plus le niveau de titre est important.

Exemple :

```
<h1>Première partie</h1>
  <h2>Sous-partie a</h2>
  <h2>Sous-partie b</h2>
<h1>Deuxième partie</h1>
```

2.3.2. Balise `<p>...</p>`

Cette balise sert à indiquer un paragraphe. En général, html considère toute suite d'espaces, tabulations et retour-chariot comme un seul espace. Pour séparer le texte en plusieurs paragraphes, avec des retours à la ligne entre chaque paragraphe, il faudra mettre chaque paragraphe dans une balise `<p>... </p>`.

Remarque sur les caractères html spéciaux :

¹ On suppose la notion d'URL connue ; sinon, voir le rappel succinct en annexe

Il est possible d'inclure dans une page web des caractères spéciaux. Ces caractères sont indiqués par un code ayant en général la forme :

```
&nomDuCaractere;
```

Mais on peut aussi utiliser la forme suivante :

```
&#codeDuCaractere;
```

Par exemple, `π` ou `π` permettent tous les 2 d'afficher le caractère π . Ceci était la façon classique d'inclure des caractères accentués avant que l'on puisse préciser l'encodage d'une page (`é` correspond au caractère é, par exemple).

Un de ces caractères reste particulièrement utile : ` ` (*non-breakable space*, l'espace insécable).

2.3.3. Balises de séparation `
` `<hr/>`

Ces deux balises permettent de séparer clairement 2 parties dans du texte. `
` est en général traduit par un retour à la ligne (préférer l'utilisation des paragraphes) et `<hr/>` par un retour à la ligne avec insertion d'une ligne horizontale de séparation.

2.3.4. Balises en ligne ``, `` ``

Ces balises sont appelés « en ligne » car leur utilisation ne génère pas de retour à la ligne (explication simplifiée).

La balise ` . . . ` (*emphasize*) sert à indiquer, dans un bloc de texte, qu'un morceau doit être mis en évidence. Cela sera traduit par défaut par de l'italique (ou des caractères droits à l'intérieur d'un texte déjà en italique).

La balise `...` sert à indiquer, dans un bloc de texte, un morceau particulièrement important. Cela sera traduit par défaut par des caractères gras.

La balise ` . . . ` est une balise en ligne qui... ne fait rien de base ! Par contre, via un fichier de style css, elle permettra d'attribuer un style particulier à son contenu.

2.3.5. Balises de liste `...`, `...` et `...`

Les balises `` et `` permettent de définir des listes à points (`ul` : *unordered list*) ou numérotées (`ol` : *ordered list*). Chaque item est spécifié par une balise ` . . . `.

Exemple :

```
<ul>
  <li> item 1;</li>
</ul>
<ul>
  <li> sous-item 1,</li>
  <li> sous-item 2</li>
</ul>
<li> item 2</li>
</ul>
```

2.3.6. Inclusion d'une image : balise

Pour inclure une image dans une page, on utilise la balise . Cette balise doit être utilisée avec au minimum les deux attributs suivants : `src`, qui doit avoir pour valeur l'URL de l'image, et `alt`, qui doit avoir pour valeur un texte à afficher si l'image ne peut être affichée ou pour les personnes mal voyantes.

Il est aussi possible d'utiliser les attributs `width` et/ou `height` pour modifier la taille de l'image (si un seul attribut est spécifié, l'autre dimension est calculée pour conserver le rapport largeur/hauteur de l'image initiale). La valeur de ces attributs peut être spécifiée en pixels ou en pourcentage de la taille initiale.

Exemple :

```

```

2.3.7. Lien vers une page web : <a>

La balise <a>... permet de définir un lien vers une autre page, ou hyperlien. Le corps de la balise contient le nom sous lequel apparaît le lien. L'URL de la page pointée par le lien, quant à elle, est indiquée grâce à l'attribut `href`.

Exemple :

```
<a href="https://fr.wikipedia.org/wiki/Highland_(race_bovine)">Les vaches  
écossaises</a>
```

2.3.8. Balise de bloc générique : <div>... </div>

Tout comme la balise ... est une balise en ligne générique, la balise <div>...</div> est une balise de bloc générique. De base, elle met son contenu dans un « bloc » qui sera inséré dans la page entre le bloc précédent et le bloc suivant, sachant que le comportement par défaut du navigateur est de constituer une page en mettant les blocs les uns sous les autres.

2.3.9. Les balises <code>...</code> et <pre>...</pre>

Ces 2 balises permettent d'afficher du texte en police à chasse fixe (type courrier). La balise <code>...</code> est une balise en ligne, tandis que la balise <pre>...</pre> est une balise de type bloc. De plus, le texte à l'intérieur de la balise <pre></pre> sera affiché en respectant les espaces du fichier source.

3. Faire ses premiers essais en html

Vous pouvez créer des pages web en utilisant un éditeur de texte (pas un traitement de texte!) et en sauvegardant votre travail dans un fichier suffixé par « .html ».

Pour visualiser votre page, vous devez la charger dans votre navigateur. Pour ce faire, il y a plusieurs moyens :

- grâce à l'item de menu « Ouvrir un fichier.. » du navigateur ;
- avec un double-clic sur le fichier dans votre navigateur de fichier ;
- en tapant « `firefox monFichier.html` » dans le terminal. **Car bien sûr, vous utilisez firefox !**

Annexe A : notion d'URL

URL absolue

Une URL (Uniform Resource Locator) est une chaîne de caractères permettant d'identifier une ressource internet par son adresse. Nous ne parlons ici que des URL correspondant des ressources sur le web. La structure d'une telle URL est la suivante :

```
protocole://serveur:port/chemin_vers_la_ressource_sur_le_serveur?  
liste_paramètres
```

Le protocole peut valoir :

- `http` : protocole web non crypté, voué à disparaître
- `https` : protocole web crypté

Le `serveur` désigne la machine sur laquelle est hébergé la ressource.

Le `port` est un nombre qui, s'il n'est pas précisé, est fixé à 80 pour le protocole `http` et 443 pour le protocole `https`. Ce nombre permet d'identifier le processus qui, sur la machine serveur, sera chargé de fournir la ressource. Quand on ne précise pas le port, on ne met pas non plus les « : » qui précèdent.

Le chemin vers la ressource est souvent une succession de chaînes de caractères séparées par des « / », à la manière des chemins d'accès à un fichier : `a/b/c.html` signifie que `c.html` est un fichier du sous-répertoire `b` du répertoire `a`. Il ne peut s'agir que d'une structure logique.

La partie `?liste_paramètres` n'est pas détaillée ici.

Exemple :

```
https://mermet.users.greyc.fr/Enseignement/EnseignementInformatiqueLycee/Havre/
```

Notion d'URL relative

Plutôt que de préciser complètement une URL dans une page web, il est possible d'utiliser une URL relative. Une URL relative est caractérisée par le fait qu'elle ne commence pas par la précision du protocole. À partir d'une URL relative, on obtient l'URL absolue en concaténant l'URL relative à la suite de l'URL absolue de la page courante. Ainsi, dans la page d'URL :

```
https://mermet.users.greyc.fr/Enseignement/EnseignementInformatiqueLycee/Havre/
```

une URL relative valant `Didactique/index.html` désigne la page d'URL absolue :

```
https://mermet.users.greyc.fr/Enseignement/EnseignementInformatiqueLycee/  
Havre/Didactique/index.html
```

Dans une URL relative, l'utilisation de `..` permet de remonter d'un répertoire.

Ainsi, dans la page d'URL absolue :

```
https://mermet.users.greyc.fr/Enseignement/EnseignementInformatiqueLycee/  
Havre/
```

L'utilisation de l'URL relative `../../usageLangages.html` désigne la page d'URL absolue :

```
https://mermet.users.greyc.fr/Enseignement/usageLangages.html
```