

Module M1106

Langages HTML/XHTML



IUT de Béziers, dépt. R&T © 2005 - 2019

<http://www.borelly.net/>

christophe.borelly@umontpellier.fr

Généralités

- **HTML** : HyperText Markup Language
- Basé sur SGML (Standard Generalized Markup Language - ISO 8879)
- Langage à balise (TAG) qui permet de structurer un texte en en-tête, paragraphes, listes, liens hypertexte, formulaires, etc...
- **XHTML** (Extensible HyperText Markup Language) est le successeur de HTML.
- Recommandations W3C (World Wide Web Consortium) : <https://www.w3.org/TR/>

Historique

- HTML 5 (2014)
- XHTML 1.1 et 2.0 (2010)
- XHTML 1.0 (2002 - version XML de HTML)
- HTML 4.01 (1999)
- HTML 3.2 (1997)
- HTML 2.0 (1995 - RFC 1866 => RFC 2854)
- HTML+ (1993)

URI

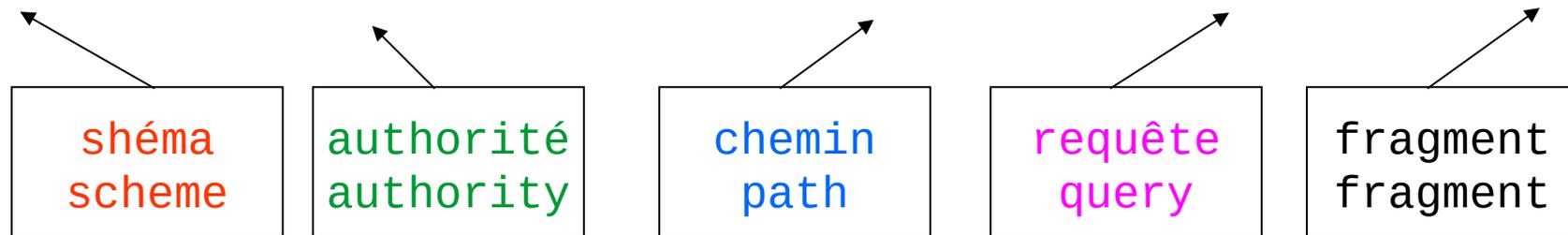
- Un URI (Universal Resource Identifier) permet d'identifier d'une façon unique une ressource sur le Web (RFC 3986).^{base}
- Un URI peut être sous classé en :
 - **URN** : Uniform Resource Name (RFC 1737)
 - **URL** : Uniform Resource Locator (RFC 1808, ...)

Caractères spéciaux

- Caractères de séparation : : / ? # [] @
- Autres délimiteurs : ! \$ & ' () * + , ; =
- Les données qui doivent contenir des caractères spéciaux doivent être encodés avec le caractère % suivi du code ASCII en hexadécimal :
 - Pour un espace (code ASCII 32), il faut utiliser %20
 - Pour % (code ASCII 37), il faut utiliser %25
- Les données d'une requête sont séparées par des & et les valeurs avec =.

Exemples d'URI

<http://example.com:8042/over/there?name=ferret#nose>



<https://www.net/a.php?user=to%20to&pass=ti%25ti%3D>

Valeur de user [to to]

Valeur de pass [ti%ti=]

URI relatifs

- Les URI relatifs ne contiennent que le chemin relatif par rapport à l'URI du document lui-même.
- Exemple : <http://www.acme.com/support/intro.html>
 - [suppliers.html](#) (ou bien [./suppliers.html](#)) correspond à <http://www.acme.com/support/suppliers.html>
 - [/index.html](#) correspond à <http://www.acme.com/index.html>
 - [../images/a.jpg](#) correspond à <http://www.acme.com/images/a.jpg>

Les balises

- Un TAG est entouré de < et >.
 - Exemple : <p>
 - En minuscule en XHTML.
- Certains TAG se terminent par </ et >.
 - Exemple : <p>Paragraphe</p>
- Un TAG peut avoir des attributs.
 - Exemple : <p id="p1">Paragraphe</p>

Structure d'un document HTML

- Trois parties (plus ou moins optionnelles) :
 - La version (!DOCTYPE)
 - L'en-tête qui contient les déclarations (HEAD)
 - Le corps avec le contenu du document (BODY)

Exemple simple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <h1>hello world!</h1>
  </body>
</html>
```

Interprétation des caractères

- Les caractères de passage à la ligne (LF 0x0A \n) sont ignorés.
- Remplacement de Carriage Return et Tab par des espaces (CR 0x0D \r, HT 0x09 \t).
- Remplacement des caractères encodés avec leur code ASCII ou UNICODE (**D** en décimal et **H** en hexadécimal) : `&#D;` ou `&#xH;` ou encore `&#XH;`;
- Pour les attributs, les espaces en début et en fin de données sont ignorés : " toto " est interprété comme "toto"

Caractères spéciaux

- La liste des caractères spéciaux est disponible sur :

<https://www.w3.org/TR/html5/syntax.html#named-character-references>

- ** **; représente un espace
- **<**; représente <
- **>**; représente >
- **&**; représente &
- **"**; représente "
- ...

Principales balises

- Commentaires : `<! - - ... - - >`
- Titres: `<h1>`, `<h2>`, `<h3>`, ...
- Paragraphes : `<p>`
- Saut de ligne : `
`
- Sections d'un document :
 - Texte : ``
 - Zone rectangulaire : `<div>`

Ajout de style CSS

- Cascading Style Sheet (voir cours 3)
- Balise **<link>** dans l'en-tête avec un URL

```
<link rel="stylesheet" href="theme.css"/>
```

- Balise **<style>** en « direct » dans le document

```
<style>  
body {background-color:red;}  
</style>
```

Balises meta

- A placer dans l'en-tête.
- Permet de décrire le document :

```
<head>  
  <meta charset="UTF-8" />  
  <meta name="description" content="Module 1106" />  
  <meta name="keywords" content="HTML, CSS" />  
  <meta name="author" content="John DOE" />  
</head>
```

- Possibilité de redirection automatique :

```
<meta http-equiv="refresh" content="30" />
```

Les liens

- `<a>` permet de créer des liens.
- Les principaux attributs sont :
 - `href` pour spécifier l'URI cible du lien.
 - `target` pour spécifier la fenêtre cible.
 - Valeurs prédéfinies : `_blank`, `_self`, `_parent` et `_top`.
 - `id` pour nommer un fragment de document.

```
<a id="debut"><h1>Sommaire</h1></a>
```

```
<a href="infos.php"  
  title="Afficher les informations">infos</a>
```

```
...
```

```
<a href="#debut">retour au début</a>
```

Les listes

- **** permet de faire des listes non ordonnées.
- **** permet de faire des listes ordonnées.
 - Pour **** et ****, l'attribut **type** permet de changer l'affichage (disc, circle, square, 1, a, A, i, I)
- **** désigne un élément de la liste (tag de fin en option).
- **<dl>** permet de faire des listes de définitions.
 - **<dt>** désigne un terme de la liste de définitions (tag de fin en option).
 - **<dd>** désigne la description du terme (tag de fin en option).

Exemple de listes

```
<ul>
  <li>Info 1
  <li>Info 2
</ul>
```

- Info 1
- Info 2

```
<ol>
  <li>Data 1
  <li>Data 2
</ol>
```

1. Data 1
2. Data 2

```
<dl>
<dt>Titre 1 :
  <dd>Exemple a
<dt>Titre 2 :
  <dd>Exemple a
  <dd>Exemple b
</dl>
```

Titre 1 :
 Exemple a

Titre 2 :
 Exemple a
 Exemple b

Les tableaux

- **<table>** permet de créer des tableaux.
- **<caption>** décrit la nature du tableau (unique et optionnel).
- **<thead>** décrit l'entête du tableau (unique et optionnel, tag de fin en option).
- **<tfoot>** décrit le « pied » du tableau (unique et optionnel, tag de fin en option).
- **<tbody>** décrit le ou les contenus du tableau (tag de début et de fin en option).
- **<tr>** décrit un ligne du tableau (tag de fin en option).
- **<th>** décrit une case d'entête du tableau (tag de fin en option).
- **<td>** décrit une case du tableau (tag de fin en option).

Exemple de tableau (1)

```
<table>
<caption>Cups of coffee consumed by each senator</caption>
<thead><tr><td colspan="5">...header information...</thead>
<tfoot><tr><td colspan="5">...footer information...</tfoot>
<tbody>
  <tr><th>Group<th>Name<th>Cups<th>Type of coffee<th>Sugar?
  <tr><td>a1<td>T. Sexton<td>10<td>espresso<td>no
</tbody>
<tbody>
  <tr><td rowspan="2">a2<td>J. Dinnen<td>5<td>decaf<td>yes
  <tr><td>A. Soria<td colspan="3">not available
</tbody>
</table>
```

Exemple de tableau (2)

Cups of coffee consumed by each senator

...header information...				
Group	Name	Cups	Type of coffee	Sugar?
a1	T. Sexton	10	espresso	no
a2	J. Dinnen	5	decaf	yes
	A. Soria	not available		
...footer information...				

Les formulaires

- **<form>** permet d'envoyer des données sélectionnées par l'utilisateur.
 - **action** définit l'URI cible (nécessaire).
 - **method** = get|post (get ajoute les données à la suite de l'URI cible).
 - **target** définit la frame cible.
 - **enctype** = application/x-www-form-urlencoded ou multipart/form-data (fichiers).

Les contrôles input

- `<input/>` permet d'ajouter dans un formulaire des éléments appelés contrôles (tag de fin interdit).
- Les principaux attributs sont :
 - `type` : Le type du contrôle (text, password, checkbox, radio, submit, reset, file, hidden, image, button).
 - `name` : Le nom du contrôle.
 - `value` : La valeur par défaut.
 - `size` : La taille graphique du contrôle.
 - `maxlength` : La taille maximum de la valeur.
 - `readonly` (text et password), `checked` (checkbox et radio).
 - `disabled` : Dévalide le contrôle.

Les contrôles select

- **<select>** permet de créer une liste déroulante.
- Les principaux attributs sont :
 - **name** : Le nom du contrôle.
 - **size** : La taille graphique du contrôle.
 - **multiple** : pour autoriser la sélection de plusieurs valeurs.
- Les valeurs sont définies avec **<option>** (tag de fin optionnel) :
 - **value** : La valeur réelle de l'option.
 - **selected**, **disabled** : sélectionné ou dévalidé.

Exemple de formulaire (1)

```
<form action="toto.php" method="get">
Nom :<input type="text" name="nom" value="toto"/><br/>
Password :<input type="password" name="pwd" value="titi"/><br/>

Homme :<input type="radio" name="c" value="homme" checked/>
Femme :<input type="radio" name="c" value="femme"/><br/>

10 :<input type="checkbox" name="choice" value="10" checked/>
20 :<input type="checkbox" name="choice" value="20"/>
30 :<input type="checkbox" name="choice" value="30"/><br/>

Pays :<select name="pays">
<option value="1">angleterre</option>
<option value="2" selected>france</option>
<option value="3">suisse</option>
</select>

<input type="hidden" name="cache" value="toto"/>
<input type="submit" value="ok"/>
</form>
```

Exemple de formulaire (2)

Nom :

Password :

Homme : Femme :

10 : 20 : 30 :

Pays :

Balises fieldset et legend

- Pour séparer graphiquement les éléments d'un formulaire, on peut utiliser **<fieldset>** et **<legend>**.

```
<fieldset>  
  <legend>Informations personnelles</legend>  
  Prénom: <input name="prenom" type="text"/><br/>  
  Nom: <input name="nom" type="text"/><br/>  
</fieldset>
```



Informations personnelles

Prénom:

Nom:

Les images

- Le tag `` permet d'inclure une image (tag de fin interdit).
- Les principaux attributs sont :
 - `src`, `alt`, `title`, `name`, `width`, `height`, `usemap` et `ismap`

```

```

Les map et area

- On peut définir des actions sur certaines zones d'une image ou d'un objet :
- **<map>** permet de donner un nom au groupe d'actions (attribut name).
- **<area>** définit chaque zone sensible ainsi que l'action correspondante.
 - **href** = URI ou nohref
 - **shape** = default|rect|circle|poly
 - **coords** = coordonnées relatives au point en haut à gauche de l'objet
 - Si shape est rect : left-x, top-y, right-x, bottom-y
 - Si shape est circle : center-x, center-y, radius
 - Si shape est poly : x1, y1, x2, y2, ..., xN, yN

Exemple de map

```

<map name="map1">
  <area href="guide.html" title="Access Guide"
    shape="rect" coords="0,0,118,28" />
  <area href="shortcut.html" title="Go"
    shape="circle" coords="184,200,60" />
  <area href="top10.html" title="Top Ten"
    shape="poly"
    coords="276,0,276,28,100,200,50,50,276,0" />
</map>
```

Ajout de scripts

- Permet de programmer en javascript coté client

```
<script src="jquery-min.js"/></script>
<script>
var i = 10;
...
</script>
```

- En XHTML, il faut ajouter `<![CDATA[...]]>`

```
<script type="text/javascript">
//<![CDATA[
var i = 10;
...
//]]>
</script>
```

Références

- <https://www.w3.org/>
- <https://tools.ietf.org/html/rfc3986>
- <https://validator.w3.org/>
- <https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5/>
- <http://www.w3schools.com/html/>