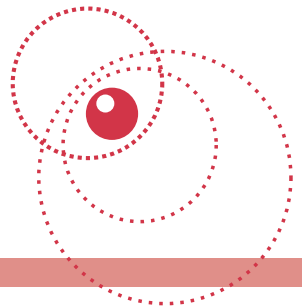

# Web Mining

## Web technologies: HTML

28 September 2010





Pierre Senellart

TELECOM ParisTech

# Le langage HTML









# Le langage HTML

## Présentation générale

### Le corps d'un document

## Formulaires HTML

## Les différentes variantes d'HTML

## Outils et Réferences



- normalisé par le W3C (World Wide Web Consortium) regroupant industriels (Microsoft, Google, Apple. . . ) et académiques (INRIA, MIT. . . )
- format ouvert : lecture possible dans des conditions correctes sans contrainte matérielle ou logicielle
- un fichier texte avec des balises
- description de la structure et du contenu d'un document, accent sur l'accessibilité
- on ne décrit pas la présentation (ce sera le rôle de CSS)
- on ne décrit pas de comportement dynamique (ce sera le rôle de JavaScript et des langages côté serveur)

- HTML est un langage qui alterne texte et balises ( `<blabla>` ou `</blabla>` )
  - Les balises permettent de structurer chaque partie du document et servent par exemple au navigateur pour réaliser la mise en page du document.
- Les fichiers HTML
  - sont structurés en deux parties principales : l'en-tête ( `<head> ... </head>` ) et le corps ( `<body> ... </body>` )
- En HTML, les blancs (espace, tabulations, retours à la ligne) sont en général équivalents et servent juste à délimiter mots, balises... Leur nombre n'a pas d'importance.

La syntaxe est (balises ouvrante et fermante)

```
<balise attributs>contenu</balise>
```

ou (balise sans contenu)

```
<balise attributs>
```

balise mot clé désignant un élément particulier

contenu peut contenir du texte ou d'autres balises

attributs représente les différents paramètres associés à l'élément, sous la forme d'une liste de `nom="valeur"` ou `nom='valeur'`, séparés par des espaces (les guillemets ne sont pas toujours indispensables, mais elles le deviennent dès que `valeur` contient des caractères exotiques)

- Les noms des éléments et des attributs sont souvent écrits en minuscule, mais `<head>` et `<HeAd>` sont équivalents.
- Les balises sont ouvertes et refermées dans l'ordre (`<b><i></i></b>` et non `<b><i></b></i>`).
- Des règles strictes déterminent quelles balises peuvent être mises à l'intérieur de quelles balises.
- Sous certaines conditions, une balise peut être implicitement refermée, mais ces conditions sont assez complexes à décrire.
- `<!--zut.-->` dénote un commentaire, qui ne sera ni affiché ni interprété par le client Web (utile pour documenter une partie d'une page Web).



## Exemples

- `<title>coucou</title>` pour attribuer le titre *coucou* au document
- `<em>cuicui</em>` pour mettre en *emphase* le texte *cuicui* (cela sera rendu, le plus souvent, par une mise en italique).
- `<strong>cuicui</strong>` pour indiquer que le texte **cuicui** est important (cela sera rendu, le plus souvent, par une mise en gras).

## Contre-exemple

```
<strong><em>bouh</strong></em>
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html lang="fr">
  <head>
    <!-- En-tête du document -->
  </head>
  <body>
    <!-- Corps du document -->
  </body>
</html>
```

- La déclaration `<!DOCTYPE ...>` précise la version d'HTML utilisée.
- La langue du document est précisée avec l'attribut `lang` de la balise principale `<html>`.



- L'en-tête du document est délimitée par les balises `<head> ... </head>`.
- L'en-tête contient des méta-informations concernant le document telles que son titre, son encodage, les fichiers annexes, etc. Les deux informations les plus importantes sont :
  - Le jeu de caractères de la page, à mettre tout au début de l'en-tête

```
<meta http-equiv="Content-Type"
      content="text/html; charset=utf-8">
```

  - Le titre de la page (la seule information obligatoire à préciser) ; celui-ci sera par exemple affiché dans la barre de titre du navigateur, il n'apparaît pas dans la page elle-même.

```
<title>Mon super site</title>
```

Unicode : répertoire de caractères, assignant à chaque caractère, de quelque langue que ce soit, un nombre entier.

## Exemples

| | | | | | |
|---|---|---|---|---|---|
| A | $\rightarrow$ | 65 | $\varepsilon$ | $\rightarrow$ | 949 |
| é | $\rightarrow$ | 233 | א | $\rightarrow$ | 1488 |

Jeu de caractères : moyen de représenter concrètement, par une suite de 0 ou de 1, un caractère Unicode.

## Exemples (é)

| | | |
|---|---|---|
| iso-8859-1 | `11101001` | Seulement pour certains caractères |
| utf-8 | `11000011 10101001` | |
| utf-16 | `11101001 00000000` | |

utf-8 présente l'avantage de pouvoir représenter tous les caractères d'Unicode, de manière compatible avec l'ancien encodage ASCII.







- Les balises `<body> ... </body>` délimitent le corps du document.
- Le corps est structuré en sections, paragraphes, listes, etc.
- Il existe 6 balises permettant de représenter les titres de sections, par importance décroissante :
  - `<h1>Titre de la page</h1>`
  - `<h2>Titre de section principale</h2>`
  - `<h3>Titre de sous-section</h3>`
  - `<h3>Titre de sous-sous-section</h3>`
  - ...

- Les balises `<p> ... </p>` permet de délimiter un paragraphe. Tous les paragraphes de texte doivent être balisés ainsi.
- La balise `<hr>` (horizontal rule) indique une séparation majeure dans le document (rendue par exemple graphiquement par une ligne horizontale).
- Directement à l'intérieur de `<body> ... </body>` n'apparaissent que des balises de bloc : `<p>`, `<h1>`, `<form>`, `<hr>`, `<ul>`, `<table>` ainsi que la balise `<div>` qui dénote un bloc sans sémantique particulière.

- XTHML possède plusieurs balises permettant de présenter le texte sous forme de listes.
- On en distingue trois types :
  - les listes non numérotées,
  1. les listes numérotées,

  les listes de définitions (ou lexiques)
- Ces listes peuvent être emboîtées les unes à l'intérieur des autres.



- Les listes classiques :
  - Les listes non numérotées délimitées par les balises `<ul> ... </ul>` (unordered list).
    - Les listes numérotées délimitées par les balises `<ol> ... </ol>` (ordered list).
    - Tous les éléments d'une liste numérotée ou non sont délimités par les balises `<li> ... </li>` (list item)
- Les lexiques sont délimités par les balises `<dl> ... </dl>` (definition list) et leurs entrées par les balises `<dt> ... </dt>` (term) et `<dd> ... </dd>` (definition).

## Exemples

```
<ol> <li>un</li> <li>deux</li> </ol>

<dl> <dt>lapin</dt> <dd>rongeur à oreilles</dd> </dl>
```

Les tableaux sont délimités par les balises `<table>... </table>`.

- Les balises `<tr> ... </tr>` (table row) délimitent les lignes.
- Les balises `<td> ... </td>` (table data) délimitent les cellules.
- Attention ! On déclare les lignes à l'intérieur du tableau, les cellules à l'intérieur des lignes.

## Exemple

```
<table>
  <tr> <td> l1, c1 </td> <td> l1, c2 </td> </tr>
  <tr> <td> l2, c1 </td> <td> l2, c2 </td> </tr>
</table>
```

Ajouter de la structure à un tableau en :

- donnant une légende au tableau avec les balises `<caption>... </caption>` juste après la balise ouvrante `<table>`.
- remplaçant les `<td> ... </td>` qui contiennent des en-têtes (de ligne, de colonne) par des `<th> ... </th>` (table header).

- Pour insérer une image dans un document HTML, on utilise la balise `<img>`.
  - L'attribut `src` permet de préciser où se trouve l'image.
  - L'attribut `alt` permet de remplacer l'image par un texte quand elle n'est pas disponible. Il est obligatoire de l'utiliser, pour que tout agent (malvoyants, navigateur texte, incidents techniques, robots) ne pouvant voir votre image puisse avoir un texte alternatif.

```
<img src="http://www.lri.fr/images/lri.png" alt="LRI">
<img src="../images/montblanc.png" alt="Mont Blanc">
```

- Les formats d'images utilisables pour le Web sont :
  - Le JPEG (.jpg), un format adapté aux photos.
  - Le GIF (.gif) et le PNG (.png), des formats adaptés aux autres types d'image ; le PNG est à préférer dans tous les cas (transparence, profondeur de couleurs...) sauf besoin d'images animées (à utiliser avec parcimonie !).

- Ce qui différencie une page Web (page HyperTexte) d'un banal document : ce sont les liens !
- Ils sont introduits par la balise `<a> ... </a>` (cette balise est une balise en ligne, il faut la placer à l'intérieur d'un bloc).
- En cliquant sur un lien, on peut se déplacer vers :
  - un autre serveur ou un fichier du même serveur
  - une autre partie du même document



Pour faire un lien, on utilise l'attribut `href` de la balise `<a>` dont le contenu formera le lien :

```
<a href="http://www.cnrs.fr/">
  <img src="images/cnrs.gif" alt="CNRS">
</a>

<a href="bio/indexbioinfo.html">Bioinformatique</a>
```

Les ancres servent à atteindre un endroit précis dans le document.

- On commence par définir les ancres, soit sur une balise existant déjà grâce à l'attribut `id`, soit avec un `<a id="...">` :

```
<h3 id="tutorials">Tutorials</h3>
<a id="tutorials">
```

- Ensuite, on fait le lien avec cette ancre.

```
<a href="#tutorials">tutorials</a>
<a href="http://www.w3.org/#tutorials">tutorials</a>
```

- On voit parfois l'ancienne syntaxe `<a name="...">`.

- De nombreuses autres balises en ligne : `<abbr>`, `<cite>`, `<var>`...
- Quelques autres balises blocs : `<blockquote>`, `<address`...
- Représentation des caractères spéciaux (ex., « é ») :
  - directement dans le codage du document (utf-8 de préférence, pour représenter tous les caractères possibles), à privilégier ;
  - par leur code en décimal (é) ou en hexadécimal (é) ;
  - par des entités caractères nommées (é)



## Formulaires HTML

- Généralités
- Les champs de saisie



## Formulaires HTML

### Généralités

### Les champs de saisie

## Les différentes variantes d'HTML

## Outils et Réferences



- Permettent d'interagir avec l'utilisateur en lui proposant d'entrer des informations
- En HTML : uniquement l'interface de formulaire
- L'essentiel du travail sera fait par le script qui traitera la soumission du formulaire



- Un formulaire HTML est placé à l'intérieur d'une balise `<form>`.
  - Celle-ci prend les attributs suivants :

| | |
|---|---|
| action | URL du script auquel sera soumis le formulaire. |
| method | Méthode HTTP, valant soit `"get"` soit `"post"`. |
| enctype | Encodage HTTP. Peut valoir `"application/x-www-form-urlencoded"` (valeur par défaut) ou `"multipart/form-data"`. |

## Example (Formulaire élémentaire)

```
<form action="action.php" method="get">
  <div><input type="submit"></div>
</form>
```

En HTML, il est interdit de mettre des champs de formulaire directement à l'intérieur d'un `<form>`. Il faut d'abord les regrouper :

- Dans des paragraphes `<p>` si les champs de formulaires sont à l'intérieur de paragraphes de textes (rare).
- Dans des ensembles de champ `<fieldset>` pour regrouper des champs de formulaire de sémantique proche. On pourra alors donner une légende à l'ensemble de champs avec la balise `<legend>`.
- Dans des divisions `<div>` sans contenu sémantique sinon.

## Exemple (Ensemble de champ)

```
<fieldset>
  <legend>Mensurations</legend>
  <input type="text" name="taille">
  <input type="text" name="poids">
</fieldset>
```

- La plupart des champs sont naturellement accompagnés d'une étiquette ( `<label>` ).
  - On peut la placer où on veut, en général juste à gauche ou à droite du champ.
  - Son attribut `for` référence l'attribut `id` du champ correspondant.
  - Dans les navigateurs graphiques, un clic sur l'étiquette d'un champ permet en général de sélectionner le champ.

## Example (Étiquette)

```
<label for="taille">Taille :</label>
<input type="text" name="taille" id="taille">
```



## Formulaires HTML

### Généralités

### Les champs de saisie

## Les différentes variantes d'HTML

## Outils et Réferences



- La balise `<input>` a une utilisation très vaste dans les formulaires. Elle représente un champ de saisie.
- L'attribut `type` détermine le type (texte, mot de passe, liste, etc.) du champ.
- L'attribut `name` (nom du paramètre de la requête HTTP) est obligatoire (sauf pour les types `"reset"` et `"submit"`) ; il permet de préciser au serveur à quelle saisie on fait référence.

## Exemple (Zone de texte pour écrire un commentaire)

```
<input type="text" name="Commentaire">
```

- `type ="text"` est utilisé pour la saisie d'un texte dont la taille est inférieure à une ligne.
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

## Exemple

```
<input type="text" name="prenom" value="Jordy" maxlength="50">
```

- `type="password"` est utilisé pour la saisie d'un texte dont les caractères sont remplacés par des astérisques : c'est généralement utilisé pour la saisie des mots de passe. Le mot de passe est quand même transmis en clair au serveur !
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

## Exemple

```
<input type="password" name="pwd" value="12345678">
```

- `type ="checkbox"` permet de choisir plusieurs éléments parmi une liste de possibilités.
- Cela se matérialise sous forme de cases à cocher.
- La valeur retournée est obligatoirement précisée à l'aide de l'attribut `value`.
- L'attribut `checked ="checked"` permet de cocher la case par défaut.
- Certains langages côté serveurs imposent que les noms de champs de formulaire à choix multiples se terminent par [].

## Exemple

```
<input type="checkbox" name="pub[]" value="site"
  checked="checked" id="pub-site">
<label for="pub-site">Recevoir des offres de notre site</label>

<input type="checkbox" name="pub[]" checked="checked"
  value="externe" id="pub-externe">
<label for="pub-externe">Recevoir des offres externes</label>
```

- `type ="radio"` permet de choisir un seul élément parmi une liste de possibilités.
- Cela se matérialise sous forme de boutons radio.
- La valeur retournée est obligatoirement précisée à l'aide de l'attribut `value`.
- L'attribut `checked ="checked"` permet de préciser la valeur par défaut.

## Exemple

```
Recevoir de la pub:
<input type="radio" name="pub" value="oui" id="pub-oui"
  checked="checked">
<label for="pub-oui">oui</label>

<input type="radio" name="pub" value="non" id="pub-non">
<label for="pub-non">non</label>
```

- `type ="file"` permet de joindre au formulaire un fichier.
- À cause de la taille de la requête due au téléchargement (upload) du fichier, il faut impérativement utiliser la méthode POST et l'encodage `multipart/form-data`.

## Exemple

```
<label for="fichier">Fichier:</label>
<input type="file" name="fichier" id="fichier">
```



`type="hidden"` permet de cacher des champs au client mais leur contenu est envoyé avec le formulaire.

- Ceci permet de préciser des informations, en utilisant l'attribut `value`, concernant l'interaction client/serveur.
- C'est à utiliser avec précaution car cela peut être à l'origine de problèmes de sécurité assez graves : ne pas oublier que le client peut éditer la page à la main pour changer la valeur de ces champs !

## Exemple

```
<input type="hidden" name="monnaie_utilisee" value="EUR">
<input type="hidden" name="customerCB" value="c2415-345-8563">
```



- `type ="reset"` permet de réinitialiser le formulaire en affectant aux différents champs leur valeur par défaut.
- L'attribut `value` permet de changer le texte du bouton correspondant.

## Exemple

```
<input type="reset" value="Tout effacer">
```



- `type ="submit"` permet de soumettre le formulaire.
- Le client envoie le contenu du formulaire à l'adresse précisée par l'attribut `action` de la balise `<form>`.
- L'attribut `value` permet de changer le texte du bouton correspondant.

## Exemple

```
<input type="submit" value="Envoyer">
```



Pour les saisies multiligne, on utilise la balise `<textarea>`.

- Le texte délimité par cette balise permet d'initialiser la valeur par défaut du champ.
- La balise fermante est obligatoire même si le champ est vide.
- Les attributs `rows` et `cols` (obligatoires) permettent de spécifier la taille en lignes et colonnes de la fenêtre de saisie.

## Exemple

```
<textarea name="bio" cols="40" rows="5">
  Fille de Josiane Balasko,
  Marilou Berry fait ses premiers pas au cinéma à 8 ans...
</textarea>
```

- La balise `<select>` permet d'ajouter une liste de sélection :
  - L'attribut facultatif `size` permet de préciser le nombre de choix apparaissant sur la page Web. Par défaut, ce nombre est initialisé à 1.
  - L'attribut `multiple ="multiple"` permet d'autoriser des réponses multiples. Dans ce cas, pour PHP, on donnera toujours un nom se terminant par `[]`.
- Les choix du menu sont indiqués à l'aide de la balise `<option>` :
  - L'attribut `selected ="selected"` permet de spécifier le(s) choix par défaut.
  - L'attribut `value` permet de spécifier la valeur associée au choix.

## Exemple

```
<select name="age">
  <option value="20">Moins de 20 ans</option>
  <option value="35" selected="selected">21 à 35 ans</option>
  <option value="50">36 à 50 ans</option>
  <option value="51">Plus de 51 ans</option>
</select>
```

Le langage HTML

Formulaires HTML

Les différentes variantes d'HTML

Outils et Réferences



- HTML 4.01 (1999) strict (as described earlier) and transitional

```
<!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">
```

- XHTML 1.0 (2000) strict and transitional

```
<!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">
```

- XHTML 1.1 and XHTML 2.0 : mostly a failure, unusable and unused in today's Web
- HTML 5 : the future standard, in development

- A lot of the HTML documents on the Web date back from before HTML 4.01
- In practice : many Web pages do not respect any standards at all (with or without doctype declarations) $\Longrightarrow$ browsers do not respect these standards $\Longrightarrow$ tag soup !
- When dealing with pages from the real Web, necessary to use all sorts of heuristics to interpret a Web page.



# XHTML : an XML format

- Tags without content `<img>`, are written `<img />` in XHTML.
- Some elements can be left unclosed in HTML (`<ol><li>one <li>two </ol>`), but closing is mandatory in XHTML.
- Attribute values can be written without quotes (`<img src=toto.png alt=toto>`) in HTML, quotes are required in XHTML.
- Element and attribute names are not case-sensitive in HTML (`<HTMl laNg=fr>`), but are in XHTML (everything must be in lowercase).
- Attributes `xmlns` and `xml:lang` on the `<html>` tag in XHTML.
- And some other small subtleties. . .



- XHTML 2.0 : initiative of the W3C, incompatible with HTML 4.01/XHTML 1.0, major changes
- HTML 5 : initiative of browser developers, compatible with HTML 4.01/XHTML 1.0, incremental but numerous changes
- XHTML 2.0 abandoned in July 2009
- HTML 5 features start appearing in recent browsers (Internet Explorer excepted, which has however indicated its interest)
- HTML 5 offers the choice between syntactic conventions inherited from both HTML 4.01 and XHTML
- New features : 2D drawing ( `<canvas>` ), multimedia ( `<audio>` , `<video>` ), better structuring elements ( `<section>` , `footer` ), etc.



Le langage HTML

Formulaires HTML

Les différentes variantes d'HTML

Outils et Réferences



- N'importe quel éditeur de texte (par exemple le bloc-notes de Windows, emacs, vim...). Privilégier les éditeurs avec coloration syntaxique. Bon choix : Scite (libre, léger, simple d'utilisation, multi-plateforme, coloration syntaxique pour HTML/CSS/PHP/JavaScript/...)
- N'importe quel navigateur, et plusieurs navigateurs avec un moteur de rendu différent
- Utiliser la fonction « Afficher la source » des navigateurs Web
- Firefox présente de nombreux avantages pour le développement/analyse de sites Web
- En particulier, extension Firebug de Firefox, excellent outil
- Pour vérifier qu'une page se conforme bien aux normes de HTML, utiliser le validateur du W3C : `http://validator.w3.org/`

- Spécification de HTML 4.01
  `http://www.w3.org/TR/REC-html40/`
- Spécification de XHTML 1.0
  `http://www.w3.org/TR/xhtml1/`
- Brouillon de XHTML 2
  `http://www.w3.org/TR/xhtml2/`
- Brouillon de HTML 5
  `http://www.w3.org/html/wg/html5/`
- *HTML et XHTML : La Référence*, O'Reilly