

---

# Développement

## HTML 5

**Jason Le Scour**

---

### Sommaire

Développement	1
HTML 5	1
<b>Sommaire</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>Les balises</b>	<b>3</b>
<b>Structure d'une page</b>	<b>4</b>
<b>Lien hypertexte</b>	<b>6</b>
<b>Les tableaux</b>	<b>7</b>
<b>Les images</b>	<b>8</b>
<b>Les formulaires</b>	<b>9</b>

---

## Introduction

L'Hypertexte Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom.

- HTML : Hyper text Markup page.
- Mise en place d'une page web.
- Certains caractères correspondent à des liens hypertexte.
- Il permet d'inclure des ressources multimédias.

---

## Les balises

Les pages web sont uniquement codées avec des caractères ASCII. Le code ASCII permet de faire des relations d'ordres. Les différentes parties sont délimitées par des balises, qui donnent les indications de la mise en forme.

### Les attributs courants

Les attributs courants donnent les informations sur l'élément HTML.

Syntaxe : Attribut = « Valeur »

- L'attribut **id** sert à identifier un élément de manière unique en lui donnant un nom.
- L'attribut **CLASS** contient le nom d'une CLASS CSS, qui contient des définitions de styles.
- L'attribut **TITLE** contient un texte qui apparaît dans une bulle quand l'utilisateur positionne le curseur quelques instants sur un élément.
- L'attribut **style** permet de définir un style localement pour un élément donné. **Il est encore toléré en HTML 5, cependant il est déconseillé.**

### Règles d'écriture des balises

Les normes HTML 5 imposent les règles suivantes :

- Les balises s'écrivent toujours en minuscule.
- Chaque balise doit être refermée, même si elles n'entourent pas un contenu.  
Exemple : Balise tracée d'une ligne horizontale `<hr />`
- Les attributs des balises sont à écrire en guillemets simples ou doubles.
- S'il y a imbrication des balises, leur ordre de fermeture est inverse à celui d'ouverture.  
Exemple :

```
<footer id="footer">
  <div class="container">
    <ul class="icons">
      <li>
        <div class="copyright">
          &copy; 2017. TEST
        </div>
      </li>
    </ul>
  </div>
</footer>
```

---

## Structure d'une page

Une page HTML 5 s'écrit de la façon suivante :

- Sur la première ligne, la balise `<DOCTYPE ...>` indique la version du HTML utilisé.
  - Le reste de la page HTML est encadrée par des balises qui signifie le début et la fin du HTML : `<html> ..... </html>`.
  - Entre ces deux balises se trouve deux parties :
    - l'entête de la page entre `<head> ..... </head>`
    - Le contenu de la page entre `<body> ..... </body>`
- Pour voir un exemple vous pouvez afficher le code source d'un site web, par exemple le miens lorsque vous êtes sur un article.

### Espace, saut de ligne et commentaires invisibles.

Dans le fichier d'édition on peut sauter des lignes et aérer le code HTML, cela à pour but de faciliter la lecture du code et à s'y retrouver.

- Quel que soit le nombre d'espaces se succédant, le navigateur n'affiche qu'un seul espace. Pour forcer l'affichage de plusieurs espaces consécutifs sur une ligne, il faut utiliser un caractère spécial : `&nbsp;` (**non breakable space**).
- Quant au saut de ligne, ils sont imposés. Pour créer un saut de ligne qui sera affiché, il faut utiliser la balise `<br />` (break).
- Pour placer des commentaires dans le code, il faut les délimités par des balises : `<!-- Démonstration d'un commentaire HTML -->`

### Remarque :

Pour coder les accents, il faut utilisé le bon charset, mais on peut également les écrire de la manière suivante :

Lettre	Code pour accentuer
a / e / ...	acute;
	grone;
	cric;

LA première ligne de l'entête est nécessairement la balise `<meta> ..... </meta>` qui indique le codage utilisé.

Exemple : `<meta charset="utf-8" />` (utf-8 bits multi-langages).

- `<h1>` correspond au titre.
- `<h2>` corespond au sous titre, etc ...
- `<ul>` correspond à une liste de puces. Chaque ligne est représentée par :

- 
- `<li>` ce qui correspond à un retour à la ligne et d'une nouvelle puce.
  - `<ol>` correspond à une liste de puces numérotée. Chaque ligne est représentée par :
    - `<li>` ce qui correspond à un retour à la ligne et d'une nouvelle puce.
  - `<p>` correspond à un paragraphe.

## Mise en forme commune

Pour regrouper en un seul bloc un ensemble de paragraphe, les titres, etc ... il suffit de les encadrer avec des balises : `<div> ..... </div>` (div comme division).

Le bloc ainsi constitué peut être encadré, mis en forme ou positionné dans la page à l'aide des styles CSS.

Les balises `<span> ..... </span>` permettent de regrouper plusieurs mot d'un paragraphe et de leur donner une mise en forme commune, par exemple la taille, la couleur ...

---

# Lien hypertexte

## Lien hypertexte (externe)

La balise `<a> ..... </a>` permet d'écrire un lien hypertexte. Le curseur prend la forme d'une main au passage de la souris et un clic permet d'afficher la page référencée. L'attribut obligatoire `href` (hypertexte référence) indique l'adresse de la page à afficher. Cette balise ne produit pas de retour à la ligne.

**Exemple :**

```
<a href= "http://www.google.fr" > <span class=« label">GOOGLE</span></a>  
<a href= "index.html" >Page d'accueil</a>
```

**Remarque :**

L'attribut `title` permet d'afficher un texte automatiquement dans une bulle lors d'un passage de la souris sur le lien.

## Lien hypertexte (interne)

Pour mettre en place un lien vers un endroit précis de la page courante (la page active / visitée), il suffit d'ajouter un identifiant en attribut de la balise de destination. Par exemple : `id = "fin de page"`. Cette dernière permet d'atteindre directement la fin de la page grâce au lien.

**Rappel :**

Un attribut c'est ce qui définit la balise.

---

## Les tableaux

- Un tableau est délimité par les balises `<table> ..... </table>`.
- L'attribut **border** indique l'épaisseur de la bordure en pixels.
- L'attribut `cellspacing` précise l'espacement entre les marges intérieures des cellules.
- Le tableau se construit ligne par ligne avec les balises `<tr> ..... </tr>`.
- Les cellules du tableau sont définies à l'intérieur de chaque ligne, grâce aux balises `<td> ..... </td>`.
- Les balises `<th> ..... </th>` indiquent le titre des colonnes.
- Les balises `<caption> ..... </caption>` permettent de définir un titre pour le tableau.

### Exemple :

```
<table>
  <thead>
    <tr>
      <th>Les abréviations / Le jargon</th>
      <th>La définition</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>FAI</td>
      <td>
        FAI est le déminutif de "fournisseur d'accès à
        Interbet".
      </td>
    </tr>
  </tbody>
</table>
```

---

## Les images

Insérer une image revient à placer un lien vers le fichier qui la contient. Pour cela nous utilisons la balise `<img ..... />`.

- L'attribut **src** est **obligatoire**, il indique le nom du fichier.
- L'attribut **alt** est **obligatoire**, c'est le texte de remplacement.
- Il est obligatoire mais peut être vide : **alt=""**.
- L'attribut **title** permet d'afficher un texte dans une bulle au passage de la souris.
- L'attribut **width** donne la largeur.
- L'attribut **height** donne la hauteur.
- Ne définir qu'un seul de ces attribut, l'autre sera calculé par rapport aux proportions. Cela permet de ne pas déformer l'image et de respecter les proportions de celle-ci.



---

## Les formulaires

Insérer un formulaire avec les balise `<form> ..... </form>`.

L'attribut `action` est obligatoire, il indique l'URL qui doit recevoir les données du formulaire.

L'attribut `method` est obligatoire, il indique comment seront envoyés les données.

- **GET** : les données sont copiées en mode texte et visible dans l'URL.

Exemple : <http://www.google.fr?id=45677&nom=TotomdpP@ssword>.

- **POST** : les données sont codées et ne sont pas visible dans l'URL.

Exemple avec le même information que la method **GET** : <http://www.google.fr>

- L'attribut `id` permet de donner un nom unique au formulaire (ne plus utiliser l'attribut `name`).
- Les balises `<fieldset> ..... </fieldset>` créent un conteneur qui regroupe les champs ayant un contenu logique.
- On peut utiliser les balises `<legend> ..... </legend>` si on veut afficher une légende à ce conteneur.
- Les balises `<label> ..... </label>` créent une zone de texte devant un champ d'un formulaire.
- L'attribut `for` est obligatoire, il indique le nom du champ.
- L'attribut `id` est obligatoire, il indique le nom unique du champ côté client en javascript.
- L'attribut `name` est obligatoire, il permet d'indiquer le nom du champ lorsque le formulaire est envoyé.

### Remarque :

Les attributs `id` et `name` sont généralement identiques sauf pour les cases à cocher et les boutons radios. En effet, la case `name` doit être identique pour tous les boutons radios et les case à cocher d'une même section. Cependant l'`id` diffère puisqu'il indique le nom du champ.

Si les attributs `id` d'une même section sont différents pour des boutons radios, alors il sera possible de cocher tous les champs. Donc le type radio qui sert à cocher qu'une seule case ne servira plus à rien. On peut prendre l'exemple de l'état civil, nous ne pouvons pas être féminin et masculin à la fois, on est soit l'un soit l'autre, c'est pour cela qu'il est intéressant d'utiliser les boutons radios.

## Les formulaires input

Les balises `<input type="...">` créent un champ saisie d'un formulaire.

- L'attribut `type` est **obligatoire**, indique le type de champ.
  - `="text"` : zone de texte à une seule ligne.
  - `="checkbox"` : case à cocher.
  - `="radio"` : case d'option.

- 
- `="password"` : zone de texte pour un mot de passe " \*\* "
  - `="number"` : zone de texte pour des caractères numériques.
  - `="hidden"` : zone de texte caché.
  - `="button"` : bouton d'action.
  - `="submit"` : bouton qui envoie le formulaire.
  - `="reset"` : bouton qui efface le formulaire.
- 
- L'attribut **value** indique le contenu initial du champ.
  - L'attribut **size** indique la taille du champ.
  - L'attribut **maxlength** indique le nombre maximal de caractère que l'on peut saisir dans le champ.
  - L'attribut **checked** `="checked"` permet de valider une option.
  - L'attribut **disabled** `="disabled"` permet de griser une option.
  - L'attribut **placeholder** permet de mettre en gris et en arrière plan un texte dans un champ de saisi pour montrer l'exemple.
  - L'attribut **step** permet de définir combien de chiffre après la virgule nous pouvons écrire dans un champ **number**.

## Les formulaire (textarea)

Les balises `<textarea> ... </textarea>` créent une zone de texte large dans un formulaire.

- L'attribut **cols** indique le nombre de colonnes.
- L'attribut **rows** indique le nombre de lignes.
- L'attribut **disabled** `="disabled"` permet de griser un champ.

## Les formulaires (select)

Les balises `<select> ... </select>` créent une liste déroulante dans un formulaire.

- L'attribut **size** indique le nombre de ligne qui seront affichées.
- L'attribut **multiple** `="multiple"` permet de pouvoir sélectionner plusieurs lignes.
- Les balises `<optgroup> ... </optgroup>` créent un groupe dans une liste déroulante avec l'attribut **label**.

Les balises `<option> ... </option>` définissent les lignes de la liste déroulante avec l'attribut **value** donne la valeur qui sera envoyé. L'attribut **select** `="selected"` permet de sélectionner une ligne.