

Les BASES du HTML

Sommaire

1	HTML = BALISAGE	2
1.1	STRUCTURE NORMALISEE D'UNE PAGE HTML	3
1.2	LES BALISES STRUCTURANTES UTILISEES PAR LE REFERENCEMENT	3
1.3	QUELQUES BALISES ORIENTEES MISE EN PAGE	4
2	LES LIENS HYPERTEXTES	4
3	ATTRIBUTS DES BALISES	5
4	BALISES MULTIMEDIA	5
5	LES TABLEAUX HTML	6
5.1	SYNTAXE MINIMUM	6
5.2	FAIRE APPARAITRE LES TRAITS	6
5.3	FUSIONNER DES CELLULES	6
6	LES FORMULAIRES HTML	8
6.1	LA BALISE PRINCIPALE : <FORM>	8
6.2	ATTRIBUT « NAME » DES CHAMPS DE SAISIE	8
6.3	LA BALISE <INPUT>	8
6.3.1	CHAMP DE TEXTE SIMPLE : TEXT	8
6.3.2	BOUTON ENVOYER : SUBMIT	8
6.3.3	BOUTON EFFACER : RESET	9
6.3.4	CHAMP DE MOT DE PASSE : PASSWORD	9
6.3.5	CHAMP CACHE : HIDDEN	9
6.3.6	BOUTON RADIO	9
6.3.7	BOUTON CHECKBOX	9
6.4	CHAMP DE SAISIE MULTI-LIGNE : TEXTAREA	10
6.5	COMBOBOX : LISTES DEROULANTES	10
6.5.1	BOUTON SIMPLE : BUTTON	11

Avertissement : ce cours n'est qu'un très bref aperçu du langage **HTML** (Hyper Text Markup Language) si vous voulez en savoir plus, il existe, sur Internet, d'innombrables sites, comme par exemple :

<http://www.w3schools.com>

<https://www.alsacreations.com/tutoriels/>

<https://www.pierre-giraud.com/html-css/cours-complet>

<https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/>

D'autre part, vous pouvez utiliser votre éditeur de texte favori (NotePad++ ou VI !!!!) pour taper les commandes HTML, mais il existe des éditeurs Wysiwyg¹ qui « masquent » le langage html (Adobe Dreamweaver, Microsoft Frontpage, Netscape Composer, ...) ou des éditeurs « intermédiaires » qui facilitent la saisie des balises (ex : Script Edit, Brackets, Atom, Sublim Text, ...).

1 HTML = BALISAGE

HTML (Hyper-Text Markup Language) : Langage de BALISAGE hypertexte.

La notion d'hypertexte est maintenant bien connue : c'est un texte sur lequel on peut cliquer, et qui entraîne une action.

Le BALISAGE consiste à **STRUCTURER** des informations dans un document.

Exemple : pour un logiciel de traitement de texte, c'est un moyen de repérer les titres, les sous-titres, les mots clés, les notes en bas de page, les sections, ... afin de réaliser par exemple une table des matières. Le langage de balisage utilisé dans ce cas est le XML (openDoc, docX, pdf ...)

Donc le but d'HTML n'est pas de décorer un texte, mais de le structurer. C'est le rôle de la BALISE (**Tag** en anglais).

La **décoration** sera obtenue par un autre moyen (Attributs des balises, ou langage CSS).

Un document bien structuré sera parfaitement référencé **dans les moteurs de recherches** car son thème et ses mots clés seront facilement détectés par les robots chargés les analyser.

Historique des versions :

HTML est apparu en 1991 et n'a cessé de s'enrichir et de se normaliser. Certaines habitudes doivent être changées pour s'adapter à la version du moment. Malheureusement, on trouve beaucoup de sites et beaucoup de tutoriels qui ne sont pas à jour. La version **HTML5** date de fin 2014 et sera la seule utilisée dans ce document.

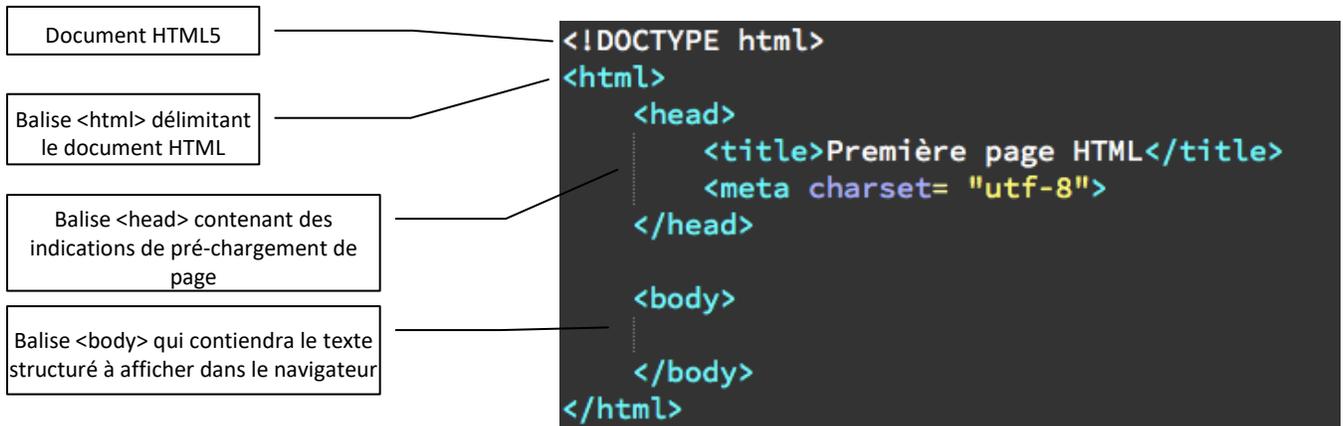
CSS est apparu en 1996 et est actuellement à la version **CSS3**. Entre 1991 et 1996, HTML a proposé des méthodes pour décorer les balises (couleur, centrage, gras, couleur de fond ...). Ces méthodes fonctionnent encore mais sont **obsolètes**. Pour enrichir le CSS, il existe aussi les ajouts des concepteurs de navigateurs, appelés **WebKit**. Certains ajouts du WebKit sont maintenant normalisés et sont intégrés à tous les navigateurs. Il est conseillé de vérifier la compatibilité avant de les utiliser (Wiki CSS : https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade#CSS3)

¹ **Wysiwyg** veut dire, Wysiwyg veut dire, *What You See is What You Get*, autrement dit : ce que vous voyez est ce que vous obtenez.

Les éditeurs **Wysiwyg** permettent de saisir les informations affichées sur votre page Web sans se soucier du codage HTML.

1.1 Structure normalisée d'une page HTML

Voici une page HTML de bas, créée par un logiciel « Editeur de texte » (NotePad++, Komodo Edit, GEany, ...) Le nom de la page doit comporter l'extension .html pour être reconnue par le navigateur internet.



- On écrit les balises en minuscule !
- Les balises HTML ont une marque de début '<' et une marque de fin '>'
- Une zone structurée est encadrée d'une balise d'ouverture (ex : <body>) et d'une balise de fermeture (ex : </body>)
- Certaines balises ne nécessitent pas de fermeture. Ex :
 <input />
- <title> </title> apparaît sur l'onglet du Navigateur. Ce n'est pas un contenu affiché.
- *Charset* permet de régler les problèmes des caractères accentués de la langue française.

1.2 Les balises structurantes utilisées par le référencement

La plupart de ces balises modifient également l'apparence du texte. Mais ce seulement pour aider le programmeur à les repérer. Par exemple, la balise <h1> s'utilise pour indiquer un titre. Par la même occasion, il écrit le texte en plus grand et en gras. Si vous voulez simplement un texte grand et gras, mais qui n'est pas un titre, n'utilisez pas une balise <h1> mais du CSS !!!

- Titres :

`<h1> Gros titre de la page </h1>`

Egalement <h2> <h3> ... jusqu'à <h6> : plus le chiffre est petit plus le titre sera gros).

- Paragraphes :

`<p> Texte du paragraphe </p>`

- Texte assez important

` Texte assez important – il sera en italique `

- Texte très important

` Texte très important – il sera en gras `

- Mots importants de la page

`<mark> mot important pour le moteur de recherche </mark>`

1.3 Quelques balises orientées mise en page

- Indentation (décalage vers la droite) :

```
<blockquote> Texte à indenter </blockquote>
```

- Conserver les espaces dans le texte :

```
<pre>
```

- Nouvelle ligne :

```
<br>
```

- Trait horizontal :

```
<hr>
```

- Les listes non numérotées : (pour les listes numérotées, utiliser)

```
<ul> ou <ol>
<li> choix1
<li> choix2
</ul> ou </ol>
```

- choix1
- choix2

1. choix1
2. choix2

- NB : CARACTERES SPECIAUX :

```
é : &eacute;
à : &agrave;
ô : &ocirc;
< : &lt;
> : &gt; ;
& : &amp;
```

2 LES LIENS HYPERTEXTES

C'est le « HT » de « HTML », donc très important à maîtriser.

Un lien peut pointer vers une page locale du site, un autre site (adresse URL) ou vers un repère placé dans la même page HTML.

- Lien vers un autre page de notre site :

```
<a href="page2.html"> Page suivante </a>
```

[Page suivante](#)

- Lien vers un autre site (URL)

```
<a href="http://www.qwant.fr"> Recherchez! </a>
```

[Recherchez!](http://www.qwant.fr)

- Lien vers un repère de la page : Le repère est un attribut « id » de la page

```
<a href="#voila"> Paragraphe suivant </a>
...
...
<p id="voila"> texte du paragraphe</p>
```

3 ATTRIBUTS DES BALISES

Dans ce qui précède, nous avons complété certaines balises avec des informations supplémentaires :

```
<a href="page2.html"> Page suivante </a>
```

```
<p id="voila"> texte du paragraphe</p>
```

« href » et « id » sont des exemples d'ATTRIBUTS ajoutés à la balise. Ils complètent cette balise.

Dans le cas de la balise <a>, « href » est le moyen d'indiquer vers où pointer le **lien**.

Dans le cas de la balise <p>, « id » est un attribut spécial qui identifie de façon **unique** cette balise <p>. Il ne doit pas y avoir d'autres attributs « id='voila' » dans le reste du document HTML.

Chaque balise peut avoir un ou plusieurs attributs. Par exemple, pour effectuer des réglages décoratifs, le CCS3 a prévu deux attributs : **style** et **class** (Voir le cours sur le CSS).

4 BALISES MULTIMEDIA

- Pour les images :

```

```

On a le cas ici d'une balise AUTO-FERMEE. Il n'y a pas de balise de fermeture .

L'attribut « alt » permet de marquer du texte au cas où l'image n'est pas trouvée à l'adresse indiquée.

On peut aussi indiquer une image extérieure au site, par son URL :

```

```

On aura sans doute besoin d'adapter la taille de l'image à notre page. Pour éviter de tomber dans le piège du vieux HTML, voici comment faire simplement avec du bon CSS3. Avec l'attribut « style » on fixe la largeur et la hauteur de l'image à 50 pixels :

```

```

- Balise audio : <audio>
- Balise vidéo : <video> Voir les tutoriels pour ces balises qui remplacent la balise <media>

5 LES TABLEAUX HTML

Le tableau permet de structurer des informations en lignes / colonnes.

On utilise la balise <table> pour englober TOUT le tableau

Chaque ligne du tableau est encadrée par la balise <tr>

Dans une ligne, chaque cellule du tableau est encadrée par la balise <td> pour une donnée, <th> pour un titre de colonne.

5.1 Syntaxe minimum

```
<table>
  <tr>
    <td> cellule 1</td>
    <td> cellule 2</td>
  </tr>
  <tr>
    <td> cellule 3</td>
    <td> cellule 4</td>
  </tr>
</table>
```

5.2 Faire apparaitre les traits

<table> est une balise de la première heure ... donc beaucoup de réglages ont été créés « à l'ancienne » : border, bgcolor, width, height, cellpadding, cellspacing, align, ...

On préférera utiliser du CSS pour modifier l'apparence du tableau :

Exemple 1 :

```
<style>
  table,th,td {
    border: 1px solid black;
  }
</style>
```

cellule 1	cellule 2
cellule 3	cellule 4

Exemple 2:

```
<style>
  table {
    border-collapse: collapse;
  }
  table, th, td {
    border: 1px solid black;
  }
</style>
```

cellule 1	cellule 2
cellule 3	cellule 4

5.3 Fusionner des cellules

Pour cette opération, nous resterons avec les anciens attributs HTML ... toute règle a ses exceptions !

6 LES FORMULAIRES HTML

Le formulaire de saisie est apparu avec ce qu'on a appelé le WEB 2.0, c'est-à-dire un Web où l'internaute ne se contente pas de visionner des pages statiques mais peut saisir des informations qui seront traitées par les serveurs et donc qui modifieront les contenus des sites. C'est un premier pas vers les pages web dynamiques.

Le premier usage a été le fameux « blog », ancêtre des réseaux sociaux. L'internaute pouvait écrire des messages sur une page que tout le monde pouvait consulter ; une révolution !

Avec le formulaire HTML sont apparus les langages côté serveur (PHP, ASP) pour traiter ces données saisies et générer de nouvelles pages html.

6.1 La balise principale : <form>

Le formulaire doit être contenu dans la balise <FORM> </FORM>

La balise <FORM> possède 2 attributs spécifiques à sa fonction :

```
<form action="action.php" method="post">
```

Action indique le nom du script qui doit traiter les données du formulaire. C'est le nom d'une page web spéciale.

Method indique le mode de transmission des informations. POST: dans l'en-tête http, GET : dans l'URL appelée.

Cas particulier :

Ici, les données sont envoyées au format texte par e-mail (votre logiciel de messagerie doit être bien sûr correctement paramétré) :

```
<form action="mailto:jdurant@site.com" method="post" enctype="text/plain">
```

Ici le formulaire de saisie

```
</form>
```

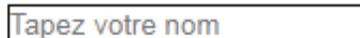
6.2 Attribut « name » des champs de saisie

Chaque champ de saisie doit avoir un attribut *name* pour que sa donnée saisie soit transmise au serveur. Le *name* choisi correspondra au nom de la variable qui sera utilisée par le serveur.

6.3 La balise <input>

Cette balise a de multiples usages en fonction de son attribut *type*.

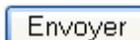
6.3.1 Champ de texte simple : text



```
<input name="nom" type="text" placeholder="Tapez votre nom"/>
```

Dans ce premier exemple et dans les suivants, on remarque l'attribut « name » : quand le formulaire sera envoyé, ce qui a été saisi dans <input> sera transmis au serveur dans une variable nommée *nom*.

6.3.2 Bouton Envoyer : submit



Une fois que votre visiteur a fini de remplir le formulaire, le bouton *submit* enverra le formulaire et toutes ses données au serveur indiqué dans *action*.

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

6.3.3 Bouton Effacer : reset



L'internaute peut effacer toutes les réponses du formulaire :

```
<input type="reset" value="Effacer"/>
```

6.3.4 Champ de mot de passe : password

Il vous est aussi donné d'utiliser ce champ afin de saisir des mots de passe de façon confidentielle.

```
<input name="mdp" type="password">
```

6.3.5 Champ caché : hidden

Il est quelque fois nécessaire d'inclure des données cachées permettant de savoir de quel formulaire viennent les données, de quel site web elles proviennent (en cas de formulaire partagé) ou d'insérer des données issues de calculs ou encore des données provenant de JavaScript.

```
<input type="hidden" name="secret1" value="le contenu secret"/>
```

6.3.6 Bouton radio

Le bouton radio sert à faire faire un choix à l'internaute parmi plusieurs propositions. Un seul choix est possible et le nombre de propositions n'est pas limité.

On utilise pour cela la balise `<INPUT TYPE="radio">` dont les attributs sont NAME : le nom du groupe de contrôles correspondant c'est-à-dire le nom du champ dans lequel ira la valeur finalement sélectionnée, VALUE : la valeur c'est-à-dire le texte qui sera inscrit dans le champ, CHECKED : facultatif car indique le bouton coché par défaut.

Vous êtes :


```
<input type="radio" name="qualite" value="MME" checked/>Une femme<br>
```

```
<input type="radio" name="qualite" value="M"/>Un homme<br>
```

```
<input type="radio" name="qualite" value="Bug"/>Un mutant<br>
```

Vous êtes :

- Une femme
- Un homme
- Un mutant

Quand l'attribut *value* est précisé, son contenu sera envoyé au serveur. Sinon, c'est le texte du choix qui sera envoyé.

6.3.7 Bouton checkbox

Le bouton *checkbox* est très semblable au bouton radio à la seule différence qu'il est possible d'en sélectionner plusieurs. L'attribut *name* est de type « tableau » récupéré dans le php par un *foreach*.

Vos sports préférés :


```
<input type="checkbox" name="sport[]" value="natation" checked/>Natation<br>
```

```
<input type="checkbox" name="sport[]" value="cyclisme"/>Cyclisme<br>
```

```
<input type="checkbox" name="sport[]" value="ski"/>Ski<br>
```

```
<input type="checkbox" name="sport[]" value="petanque" checked/>Petanque<br>
```

```
<input type="checkbox" name="sport[]" value="basket"/>Basket
```

Vos sports préférés :

- Natation
- Cyclisme
- Ski
- Petanque
- Basket

6.4 Champ de saisie multi-ligne : textarea

Il est souvent nécessaire à l'internaute d'écrire un commentaire assez long, la balise précédente ne convient pas. On utilise le conteneur <TEXTAREA> pour offrir au visiteur un espace multi-lignes qui lui permettra d'écrire un texte long.

Il a pour attributs NAME : le nom du champ, ROWS : le nombre de lignes, COLS : le nombre de colonnes, READONLY : le champ se trouve alors en mode lecture seule c'est-à-dire que le texte par défaut qui s'y trouve ne peut en être effacé.

Syntaxe :

```
<textarea name="nom_du_champ" rows="nombre_de_lignes" cols="nombre_de_colonnes"></textarea>
```

6.5 ComboBox : listes déroulantes

Les boutons radio et checkbox sont bien mais quand on a un choix à faire entre 36 valeurs différentes, mieux vaut utiliser les listes déroulantes avec le conteneur <SELECT> dont les attributs sont NAME : le nom de la liste, SIZE (facultatif) dont la valeur unité (par défaut) signifie qu'on a affaire à une liste déroulante sinon c'est une liste normale mais avec barre de défilement, MULTIPLE (facultatif) qui signifie qu'il est possible de sélectionner plusieurs valeurs. Si MULTIPLE n'est pas spécifié, il ne sera possible de faire qu'un seul et unique choix.

Le conteneur <SELECT> en lui-même ne suffit pas, il faut lui rajouter autant de balise <OPTION> qu'il y a de choix possibles. Les attributs de cette dernière sont VALUE : valeur du choix sélectionnable c'est-à-dire le texte qui sera inscrit dans le champ portant le nom de la liste et SELECTED (facultatif) qui signifie que le choix correspondant sera sélectionné par défaut.

```
Quels pays limitrophe avez-vous visité ? <br/>
<select name="jeux size=4 multiple">
  <option value="fr"/>France
  <option value="es"/>Espagne
  <option value="it" selected/>Italie
  <option value="be"/>Belgique
  <option value="en"/>Angleterre
  <option value="sc" selected/>Ecosse
  <option value="lx"/>Luxembourg
  <option value="sw"/>Suisse
  <option value="de"/>Allemagne
</select>
```

Quels pays limitrophe avez-vous visité ?

6.5.1 Bouton simple : button

Pour des besoins de contrôle du formulaire avec JavaScript, pour pouvez être amené à créer un bouton qui ne soit dédié ni à l'envoi du formulaire ni à son ré-initialisation. L'utilisation de la balise `<INPUT TYPE="button">` fera apparaître un bouton simple qui peut appeler une fonction JavaScript.

Syntaxe :

```
<input type="button" value="texte à afficher" onClick="fonction_JavaScript()"/>
```

Ex :

```
<input id="bou1" type="button" value="Allez!"/>
<script>
bou1.addEventListener("click", fonc_depart); // Méthode DOM2

function fonc_depart() {
    alert("C parti !");
}
</script>
```

6.6 Pièce jointe: file

Vos visiteurs auront sans doute envie de vous laisser un fichier, pour cela il vous faut utiliser la balise `<INPUT TYPE="file">` dont l'attribut `SIZE` a pour valeur la taille (en caractères) du champ correspondant. Ce champ sera suivi d'un bouton qui permet d'ouvrir une boîte de dialogue **Open** pour choisir le fichier.

Point capital nécessaire pour l'envoi d'un fichier en pièce jointe : il faut changer la valeur de l'attribut `ENCTYPE` du conteneur `FORM` en `multipart/form-data`. En effet, `ENCTYPE` précise au navigateur le type MIME du message envoyé lors de la validation du formulaire. Le type MIME `plain/text` vu plus haut signifie texte brut, alors que `multipart/form-data` signifie que le message peut contenir des données binaires. Si vous oubliez de spécifier le bon code MIME pour l'envoi d'un fichier, vous ne recevrez qu'une chaîne de caractères contenant le nom et le chemin de ce fichier dans l'ordinateur de l'internaute !

Il est également possible de limiter la taille du fichier à envoyer afin de limiter l'encombrement de votre boîte aux lettres. Pour cela, ajouter un champ caché nommé `MAX_FILE_SIZE` et de valeur la taille maximal acceptée en octets.

Syntaxe :

```
<form name="nom_du_formulaire" action="mailto:votre@email" method="post" enctype="multipart/form-data">
  <input type="hidden" name="max_file_size" value="taille"/>
  <input type="file" name="nom_du_champ" size="taille"/>
</form>
```

Exemple :

```
<form name="cyberform" action="mailto:cyberzoide@multimania.com"
method="post" enctype="multipart/form-data">
<input type="hidden" name="max_file_size" value="204800"/>
<input type="file" name="fichier" size=25/>
</form>
```

6.7 Boutons personnalisés

Si ces boutons ne vous plaisent pas, rassurez-vous car vous pouvez en changer l'apparence tout simplement en créant vous même l'image correspondante au format GIF (ou JPEG). Pour le bouton reset, vous n'avez même pas besoin de passer par une balise de bouton car il vous suffit d'encadrer votre image par un conteneur <A HREF> pointant vers votre formulaire.

Syntaxe :

```
<a href="_du_formulaire"></a>
```

Exemple :

```
<a href="formulaire.php3!"></a>
```

The image shows the word "ANNULER" in a stylized, pink, bubbly font with a purple outline, representing the visual output of the HTML code example.

Pour le bouton d'envoi, il faut utiliser la balise <INPUT TYPE="image"> dont les attributs sont SRC qui a pour valeur l'adresse de l'image à afficher, WIDTH qui spécifie la largeur de l'image, HEIGHT pour la hauteur et BORDER pour la bordure.

Syntaxe :

```
<input type="image" src="adresse_de_l'image" width="largeur" height="hauteur" border="bordure">
```

Exemple :

```
<input type="image" src="images/envoit.gif" width=106 height=23 border=0 alt="Envoyer">
```

