

HTML : Hyper Text Markup Language

Plan

- 1 Introduction
- 2 Concept de balises
- 3 Structure d'une page HTML
- 4 Attributs d'une balise
- 5 Organisation du texte
- 6 Les titres et les listes
- 7 Les liens

Plan

- 8 Les images
- 9 Les tableaux
- 10 Les formulaires
- 11 Encore un peu de multimédia
- 12 Classification des balises
- 13 Restructuration d'une page web avec HTML5
- 14 Les caractères spéciaux
- 15 HTML5 et compatibilité des navigateurs

Hyper Text Markup Language

Définition et caractéristiques

- Ce n'est pas un langage de programmation
- C'est plutôt un langage de description
 - composé de plusieurs balises (tags)
 - interprété par le navigateur
- Il ne nécessite pas un éditeur de texte particulier
- Il n'est pas sensible à la casse

Hyper Text Markup Language

Quelques éditeurs de texte

- **Sublime text** : <https://www.sublimetext.com/3>
- **Atom** : <https://atom.io/>
- **Notepad++** : <https://notepad-plus-plus.org/download/v7.5.7.html>
- **Brackets** : <http://brackets.io/>
- ...

Quelques navigateurs

- **Google chrome** : <https://www.google.com/chrome/>
- **Mozilla firefox** : <https://www.mozilla.org/fr/firefox/new/>
- ...

Hyper Text Markup Language

Quelques éditeurs de texte

- **Sublime text** : <https://www.sublimetext.com/3>
- **Atom** : <https://atom.io/>
- **Notepad++** : <https://notepad-plus-plus.org/download/v7.5.7.html>
- **Brackets** : <http://brackets.io/>
- ...

Quelques navigateurs

- **Google chrome** : <https://www.google.com/chrome/>
- **Mozilla firefox** : <https://www.mozilla.org/fr/firefox/new/>
- ...

CodePen : une autre solution en ligne

Trois éditeurs en parallèle : un pour HTML, un pour CSS et un pour JavaScript
<https://codepen.io/>

Hyper Text Markup Language

HTML : évolution

- HTML1 : première version créée par Tim Berners-Lee en 1991.
- HTML2 : deuxième version, apparue en 1994. On commence à parler de W3C (World Wide Web Consortium, organisme de standardisation fondé par Tim Berners-Lee).
- HTML3 : apparue en 1996 avec plusieurs nouveautés comme les tableaux, les scripts, le positionnement du texte autour des images, etc.
- HTML4 : apparue en 1998 avec la possibilité :
 - d'utiliser de frames (découpage d'une page en plusieurs parties),
 - des améliorations sur les formulaires,
 - d'utiliser des feuilles de style (CSS).
- **HTML5** : finalisée en octobre 2014 et permet :
 - d'inclure facilement des vidéos,
 - d'ajouter plus de précisions sur les champs d'un formulaire

Les balises

Syntaxe :

```
<balise>  
    objet  
</balise>
```

ou bien (les balises orphelines ou auto-fermantes) :

```
[objet] <balise/>
```

Commentaire : balise particulière

```
<!-- ceci est un commentaire -->
```


Les balises

Propriétés

- Les balises doivent être ouvertes puis fermées récursivement, comme des parenthèses ([...]{(...)})
- Tout ce qui est entouré par deux balises <tag> et </tag> s'appelle un Élément de tag
- Un élément est ainsi considéré comme une boîte pouvant éventuellement contenir du texte et/ou images, liens...
- La fermeture et l'ouverture de deux boîtes ne peuvent pas se croiser : (...[...])...
- Lorsqu'un navigateur ne reconnaît pas une balise, ou lorsqu'il rencontre du texte en clair, (c'est à dire à l'extérieur de tout élément), le texte est affiché simplement.
- Les balises non reconnues ne sont toutefois pas affichées

Les balises

Les commentaires : propriétés

- Les commentaires en HTML sont ignorés par le navigateur
- Les commentaires ne sont pas affichés, mais ils restent visibles dans le code source de la page. Un commentaire ne doit donc jamais contenir une information confidentielle.
- On évite une erreur javascript en mettant en commentaire `//-->`

```
<script type="text/javascript">  
<!--  
function displayMsg() {  
    alert("Hello_World!")  
}  
//-->  
</script>
```

Organisation d'une page HTML

Composition

- Le contenu d'une page HTML est compris entre deux balises HTML
- Une page HTML est composée de deux parties
 - une entête 'HEAD'
 - un corps 'BODY'

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
  </body>
</html>
```

Et c'est quoi DOCTYPE ? !

DOCTYPE

- Ce n'est pas une balise
- C'est facultatif
- C'est une directive permettant de préciser qu'il s'agit d'un document HTML et indiquant sa version

```
<!-- HTML4.01 transitional -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01_
  Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

<!-- HTML5 -->
<!DOCTYPE html>
```

Que peut contenir <HEAD> ? !

```
<head>
  <meta charset="utf-8" />
  <meta name="description" content="Formation_
    POEC_PHP" />
  <title>Formation 2017</title>
</head>
```

<HEAD> : les méta informations

- <title> titre du document (affiché par le navigateur en haut de la page)
- <link/> pour référencer un fichier (CSS par exemple)
- <style> pour inclure du code CSS
- <meta> peut contenir :
 - des informations sur le codage
 - des informations pour les moteurs de recherche

Que peut contenir <BODY> ?!

<body>

informations qui vont s'afficher dans le navigateur

</body>

Dans le <Body>

- texte
- tableau
- image/vidéo/document
- menu
- lien
- formulaire
- liste
- ...

Une balise peut avoir des attributs

Dans certains cas, les seules balises ne suffisent pas. Il faut généralement y ajouter des attributs.

```
<tag attribute="value">...</tag>  
<tag attribute="property:value;">...</tag>  
<tag att1="v1" att2="v2" >...</tag>
```

Quelques attributs standards

- `class` : nom de classe (*à voir dans le cours CSS*)
- `id` : identifiant unique dans la page (*à voir dans le cours CSS*)
- `style` : source CSS style (CSS) de l'élément
- `value` : pour la valeur à afficher (*à voir dans les formulaires*)
- ...

L'attribut style

L'attribut style

```
<tag style="property:value;">...</tag>
```

Exemple

```
<p style="color:blue;">Ce paragraphe sera affiche en  
bleu.</p>
```

Résultat

Ce paragraphe sera affiche en bleu.

Une balise peut avoir des attributs

Quelles sont les autres propriétés de style ?

- `style="font-size:60%;"` : pour la taille du texte
- `style="text-align:center;"` : pour l'alignement du texte
- `style="background-color:red;"` : pour la couleur de l'arrière plan
- `style="font-family:arial;"` : pour le font du text
- ...

Pour choisir une couleur :

https://www.w3schools.com/colors/colors_picker.asp

Comment organiser notre texte ? !

Les paragraphes

- `<p>...</p>` : pour délimiter un paragraphe
- `
` : pour aller à la ligne
- `<nobr>...</nobr>` : pour forcer l'écriture sur une seule ligne
- `...` : pour mettre un texte en gras
- `<u>...</u>` : pour souligner un texte
- `<i>...</i>` : pour mettre un texte en italic
- `<hr/>` : pour afficher une ligne horizontale
- `<pre>...</pre>` : pour ajouter un texte pré-formaté

Il ne faut jamais oublier qu'on utilise HTML pour la structure et CSS pour le style

Comment organiser notre texte ? !

On peut plutôt utiliser :

- `...` : pour mettre un texte bien en valeur
- `...` : pour mettre un texte un peu en valeur
- `<mark>...</mark>` : pour marquer un texte

Quelques balises de formatage

Quelques exemples

- `<acronym>` : acronyme
- `<abbr>` : abréviation
- `<address>` : adresse formatée
- `<center>` : texte centré
- `<cite>` : citation
- `<code>` : élément de code informatique
- `` : texte supprimé dans un document
- `<ins>` : texte inséré dans un document
- `<sub>` : texte en indice
- `<sup>` : texte en exposant
- `<tt>` : texte télétype

Comment organiser notre texte ? !

Règles générales

- Les navigateurs ajoutent automatiquement un espace entre chaque paragraphe
- Bien que le texte 'flottant' (non inclus dans un élément de la page) soit affiché par les navigateurs, il vaut mieux pour des raisons de 'style' inclure la totalité du texte utile de la page dans des balises, `<p>` notamment.

Les titres et les listes

Les titres

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

Les titres et les listes

Les titres

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

Les listes

- `...` : une liste non-ordonnée (non-numérotée)
- `...` : une liste ordonnée
- `...` : un élément d'une liste

Les liens

Les liens

- `` : insérer une image
- `<a>...` : insérer un lien (interne ou externe)

Les liens

- pour créer un lien vers un autre document, relatif ou absolu, local ou distant, comme
` monsite ` ou
` Page 2 `
- pour pointer vers un signet ('name' ou 'id') dans un document
` vers le haut de page `
- Le plus important attribut de l'élément `<a>` est l'attribut `href`, qui indique la destination du lien.

Liens spéciaux

Lien pour télécharger un fichier :

```
<a href="monfichier.zip">Download</a>
```

Lien pour ouvrir une nouvelle fenêtre :

```
<a href="http://www.lsis.org/elmouelhia/" target="_blank">cliquer ici</a>
```

Les images

Les images

- La balise `` permet d'insérer des images dans une page internet affichée `` ``
- Une image n'est pas 'incluse' dans un document. Elle est référencée par son adresse sur internet, comme un lien
- Si le fichier (src) d'une image n'est pas accessible, un navigateur peut afficher soit une zone rectangulaire de bonnes dimensions, soit un texte alternatif

Les attributs

- src : URL de l'image
- alt : texte affiché si indisponible
- height : hauteur
- width : largeur

Les tableaux

Les tableaux

- Les tables constituent un mode privilégié de présentation d'information structurée.
- Une table (balise `<table>`) est divisée en lignes et colonnes
- Une bordure délimite ou non les cellules de la table (Balise `<td>`), organisée en lignes (balise `<tr>`)
- La table préserve en permanence l'aspect visuel de colonnes dont toutes les cellules ont la même largeur.
- En revanche, les lignes peuvent avoir des hauteurs différentes

Les tableaux

Les différentes balises d'un tableau

- `<table>` : tableau
- `<caption>` : légende du tableau (texte associé)
- `<th>` : cellule d'en-tête dans un tableau
- `<tr>` : ligne dans un tableau
- `<td>` : cellule dans un tableau

Les tableaux

```
<table border="1">  
  <caption>Moyenne par matiere</caption>  
  <tr>  
    <th>Matiere</th>  
    <th>Moyenne</th>  
  </tr>  
  <tr>  
    <td>PHP</td>  
    <td>10</td>  
  </tr>  
  <tr>  
    <td>Java</td>  
    <td>8</td>  
  </tr>  
</table>
```

Fusionner des cellules d'un tableau

```
<table border="1">  
  <tr>  
    <th>Nom</th>  
    <th colspan="2">Mail</th>  
  </tr>  
  <tr>  
    <td>Achref El Mouelhi</td>  
    <td>achref@elmouelhi.fr</td>  
    <td>elmouelhi@achref.fr</td>  
  </tr>  
</table>
```

Fusionner des cellules d'un tableau

```
<table border="1">  
  <tr>  
    <th>Nom</th>  
    <td>Achref El Mouelhi</td>  
  </tr>  
  <tr>  
    <th rowspan="2">Mail</th>  
    <td>achref@elmouelhi.fr</td>  
  </tr>  
  <tr>  
    <td>elmouelhi@achref.fr</td>  
  </tr>  
</table>
```

Autres attributs de la balise `<table>`

Les attributs

- `padding` : espace entre le contenu de la cellule et les frontières (la bordure)
- `border-spacing` : espace entre les cellules
- `width` : pour définir la largeur
- `height` : pour la hauteur
- `align` : pour l'alignement du texte dans une cellule
- `valign` : pour l'alignement vertical du texte dans une cellule
- ...

Les tableaux

Remarques

- Les tables ont été largement utilisées dans le passé pour organiser les pages des sites web.
- Cet usage est aujourd'hui obsolète et absolument découragé.
- On utilise pour cela les balises `<div>` et les styles CSS qui seront vus plus loin.

Les formulaires

Déclaration d'un formulaire :

```
<form method="post_ou_get" action="page_web_destination">  
</form>
```

Les attributs d'un formulaire

- `method` : concerne l'envoi de données et peut prendre deux valeurs.
 - `get` : non fréquemment utilisée car limitée à 255 caractères. En plus, les informations envoyées seront visibles dans la zone d'adresse.
 - `post` : plus utilisée que `get` car elle permet d'envoyer un grand nombre d'informations et les données saisies dans le formulaire ne transitent pas par la barre d'adresse.
- `action` : indique l'adresse de la page ou du programme qui va traiter les informations (généralement avec un langage autre que HTML).

Que peut-on mettre dans un formulaire ? !

Une zone de saisie monoligne :

```
<input type="text" name="nom_zone_texte" />
```

Une zone de saisie multiligne :

```
<textarea name="nom_zone_texte" id="identifiant"></  
textarea>
```

Un libellé associé à une zone de saisie :

```
<label for="identifiant_zone_saisie">text</label>
```

Exemple :

```
<label for="nom">Nom :</label> <input type="text" name="nom"  
id="nom" />
```

Généralement on associe la même valeur aux attributs `id` et `name`

Que peut-on mettre dans un formulaire ? !

Avec HTML 5, les 3 écritures suivantes sont équivalentes

```
<input type=text value=John>
```

```
<input type="text" value="John_Wick">
```

```
<input type='text' value='John Wick'>
```

Les autres zones de saisie

Avant HTML5

- `password` : pour les mots de passe

Avec HTML5

- `email` : pour les adresses emails
- `number` : pour les nombres (attributs `min`, `max` et `step`)
- `color` : pour les couleurs
- `date` : pour les calendriers
- ...

Les zones de choix

Trois types

- Les cases à cocher
- Les boutons radio
- Les listes déroulantes

Les cases à cocher :

```
<input type="checkbox" name="nom_du_choix" [checked] />
```

Les boutons radios :

```
<input type="radio" name="nom_du_choix" [checked] />
```

Les listes déroulantes :

```
<select name="liste">  
  <option value="valeur1" [selected]>valeur1</option>  
  ...  
  <option value="valeurN">valeurN</option>  
</select>
```

Les zones de choix

Les listes déroulantes (le regroupement) :

```
<select name="liste">
<optgroup label="label1">
<option value="valeur1">valeur1</option>
    ...
<option value="valeur1">valeurN</option>
</optgroup>
<optgroup label="label2">
<option value="valeur1">valeur1</option>
    ...
<option value="valeur1">valeurM</option>
</optgroup>
    ...
</select>
```

Les zones de choix

Les listes de données (datalist) = liste déroulante + zone de saisie

```
<input list="sports" name="sport">
```

```
<datalist id="sports">  
  <option value="football">  
  <option value="handball">  
  <option value="cross_fit">  
  <option value="tennis">  
  <option value="hockey">  
</datalist>
```


Regroupement des éléments d'un formulaire

Exemple :

```
<form method="post" action="traitement.php">
<fieldset>
<legend>Nom complet</legend> <!-- partie nom prenom -->
<label for="nom">nom :</label> <input type="text" id="nom
" /><br/>
<label for="prenom">prenom :</label><input type="text" id
="prenom" />
</fieldset>
<fieldset>
<legend>Sexe</legend> <!-- partie sexe -->
<input type="radio" name="homme" value="homme"/> <label
for="homme"> Homme </label><br/>
<input type="radio" name="femme" value="femme"/> <label
for="femme"> Femme </label>
</fieldset>
</form>
```

Les formulaires

Encore quelques attributs

- `required` : pour indiquer qu'un champ est obligatoire
- `autofocus` : pour placer le curseur dessus dès chargement de la page
- `autocomplete` : pour indiquer si on autorise auto-complétion
- `multiple` : pour indiquer qu'il est possible de choisir ou d'insérer plusieurs éléments
- `placeholder` : pour afficher un message indicatif dans un champ
- `readonly` : pour rendre le champ en lecture seule

Les boutons

Trois types de boutons

- `submit` : envoie le contenu d'un formulaire à la page indiqué dans l'action
- `reset` : remet à zéro (efface) le contenu d'un formulaire
- `button` : bouton générique qui permet de réaliser plusieurs tâches pour HTML (quitter une page,...) ou de déclencher un code JavaScript

Les fichiers audios

Insérer un élément audio :

```
<audio src="audio.mp3"></audio>
```

Attention

Les navigateurs ne supportent pas tous les formats audios existants

Les fichiers audios

Insérer un élément audio :

```
<audio src="audio.mp3"></audio>
```

Attention

Les navigateurs ne supportent pas tous les formats audios existants

Les attributs possibles

- `controls` : pour afficher les boutons lecture et pause
- `loop` : pour jouer le fichier audio en boucle
- `autoplay` : pour lire le contenu du fichier dès le chargement de la page

Les fichiers vidéos

Insérer un élément vidéo :

```
<video src="fichier.mp4"></video>
```

Attention

Les navigateurs ne supportent pas tous les formats vidéos existants

Les fichiers vidéos

Insérer un élément vidéo :

```
<video src="fichier.mp4"></video>
```

Attention

Les navigateurs ne supportent pas tous les formats vidéos existants

Pour éviter ce problème : définir plusieurs formats

```
<video controls>  
  <source src="fichier.mp4" />  
  <source src="fichier.webm" />  
  <source src="fichier.ogv" />  
</video>
```

Les fichiers vidéos

Les attributs possibles

- `poster` : image à afficher à la place de la vidéo
- `controls` : pour afficher les boutons lecture et pause
- `loop` : pour jouer le fichier audio en boucle
- `autoplay` : pour lire le contenu du fichier dès le chargement de la page

Deux types de balises

inline et block

- `block` : c'est une balise qui ajoute automatiquement un retour à la ligne avant et après.
- `inline` : c'est une balise qui se situe obligatoirement à l'intérieur d'une balise `block`. Elle ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne.

Exemple

- `block` : `<p>`, `<h1>`...
- `inline` : `<a>`...

Deux types de balises

Les balises universelles

Ce sont des balises sans aucune sémantique, utilisées souvent en CSS ou dans les formulaires.

- `...` : inline
- `<div>...</div>` : block

Deux types de balises

Quelques propriétés de balise `inline`

- Il ignore les marges `top` et `bottom` mais applique les marges `left` et `right`, ainsi que tout `padding`.
- Il ignore les propriétés `width` et `height`.

Vers un format uni ?

Les balises de restructuration

- `<header>...</header>` : l'entête de ma page
- `<footer>...</footer>` : le pied de ma page
- `<nav>...</nav>` : l'emplacement du menu
- `<section>...</section>` : le corps de la page peut être composé de plusieurs sections
- `<article>...</article>` : dans une section on peut définir un article
- ...

Vers un format uni ?

Les balises de restructuration

- `<header>...</header>` : l'entête de ma page
- `<footer>...</footer>` : le pied de ma page
- `<nav>...</nav>` : l'emplacement du menu
- `<section>...</section>` : le corps de la page peut être composé de plusieurs sections
- `<article>...</article>` : dans une section on peut définir un article
- ...

Ne pas confondre les deux balises `<head>` et `<header>`

Vers un format uni ?

Les balises de restructuration

- `<header>...</header>` : l'entête de ma page
- `<footer>...</footer>` : le pied de ma page
- `<nav>...</nav>` : l'emplacement du menu
- `<section>...</section>` : le corps de la page peut être composé de plusieurs sections
- `<article>...</article>` : dans une section on peut définir un article
- ...

Ne pas confondre les deux balises `<head>` et `<header>`

Ces balises sont plutôt sémantiques et n'ont pas de positions fixes

Les caractères spéciaux

Le codage de caractères spéciaux

- `é` : é
- `è` : è
- `ê` : ê
- `ç` : ç
- `&` : &
- `æ` : æ
- `×` : ×
- `"` : ”
- `<` : <
- `>` : >
- ...

Les caractères spéciaux

Le codage de caractères spéciaux

- `é` : é
- `è` : è
- `ê` : ê
- `ç` : ç
- `&` : &
- `æ` : æ
- `×` : ×
- `"` : ”
- `<` : <
- `>` : >
- ...

https://www.w3schools.com/charsets/ref_utf_latin1_supplement.asp

HTML5 et compatibilité des navigateurs

Pour tester la compatibilité d'un élément HTML5 avec les navigateurs

<https://caniuse.com/>