

HTML



L'HTML est **LE** langage de description permettant de représenter les pages Web.

Il permet de structurer et mettre en forme le contenu des pages et d'inclure des ressources (liens vers d'autres pages, images, vidéos ...). Il peut contenir d'autres langages pour rendre les pages dynamiques (JavaScript ou PHP). C'est un langage défini par une organisation internationale : le World Wide Web Consortium (W3C).

Le langage HTML est un langage **interprété**. Le navigateur Web est l'interpréter du code HTML. Tous les navigateurs (Chrome, Safari, Firefox, etc.) savent lire le HTML et le transformer en page web mise en forme avec des images, du texte, des tableaux, etc.



HTML a connu plusieurs versions au cours du temps. Nous nous concentrerons sur HTML 5 (apparu en 2008 et normalisé en oct. 2014). Cette version a révolutionné le Web en intégrant de fonctionnalités multimédia (vidéo, audio) et graphiques (canvas) ainsi que la notion de Responsive Web Design.

Mais c'est important de noter que l'objectif du HTML est de définir le fond : la **structure** et le **contenu** de la page et non pas la forme.

Comment coder en HTML ?

Il existe des logiciels spécialisés (BlueGriffon, Dreamweaver, etc.) qui permettent d'organiser une page Web comme un document à l'aide d'un traitement de texte. Mais il est souvent plus intéressant d'utiliser un éditeur de texte (comme Notepad++, Vim, etc.) car cela donne une meilleure maîtrise du code utilisé et donc de la page ainsi créée.

Les développeurs web choisissent souvent un éditeur de texte configurable avec des plugins qui colore les balises, les noms des attributs, la valeur des attributs et le contenu. C'est le cas des éditeurs Atom, Sublime Text (avec des plugins Linter-html-tidy), etc.

Balises

Le langage HTML est composé de **balises** pour décrire le contenu et la manière d'afficher ce contenu à l'écran. Une balise est un code ou mot-clé encadré par les caractères < et >. Comme par exemple <div>Bonjour</div> ou bien
. Ces balises peuvent s'imbriquer les unes dans les autres (mais toutes les combinaisons ne fonctionnent pas).



An autre exemple : pour «bold» (gras)

Ce texte est en gras ⇒ **Ce texte est en gras**

Les balises fonctionnent en général par paire. Par exemple la balise dite d'ouverture est associée à la balise de fermeture pour mettre les mots en gras.

Les balises ne peuvent s'entremêler :

`<a> ` : ok

`<a> ` : FAUX Notion d'arborescence hiérarchique.

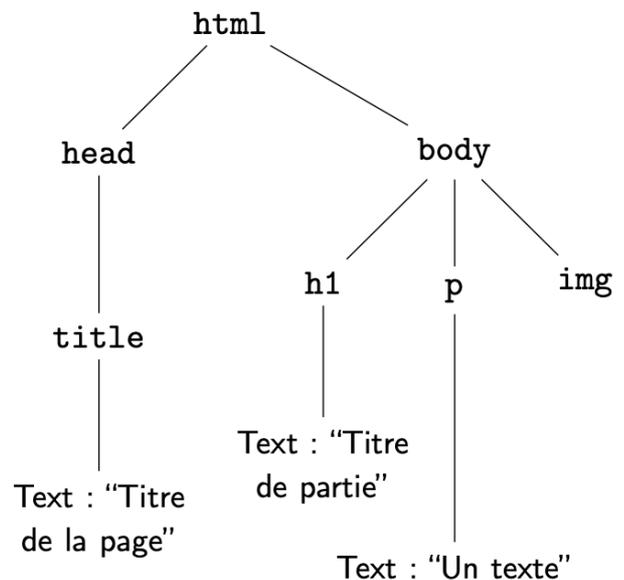
Toutes les balises ne sont pas destinées à contenir du texte. Certaines ne sont que de simples marqueurs (saut de ligne, insertion d'images, etc.). Il n'est, alors, pas nécessaire de délimiter un début et une fin. Ce sont des balises dit orphelines. Exemple : `` pour l'insertion d'une image `
` pour un saut de ligne¹.

Il existe environ 125 types de balises différentes (<https://jaetheme.com/balises-html5/>), certaines sont obsolètes, certaines sont obligatoires, d'autres facultatives, certaines ne peuvent exister qu'à l'intérieur d'une autre balise, etc. Vous ne connaîtrez jamais (... ou disons c'est improbable de connaître) les 125 balises et les certaines d'attributs possibles de HTML. Vous devrez apprendre les balises les plus importantes et simplement vous rappeler que certaines autres existent et être capable de les retrouver quand vous en aurez besoin.

Structure de base d'une page HTML5

Un fichier HTML doit toujours avoir l'extension `.html` afin d'être correctement reconnu en tant que fichier HTML par le navigateur. Voici une page.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- En-tete de la page -->
    <meta charset="utf-8" />
    <title>
      Titre de la page
    </title>
  </head>
  <body>
    <h1>
      Titre de partie
    </h1>
    <p>
      Un texte
    </p>
    <img />
  </body>
</html>
```



¹ Les navigateurs web sont le plus tolérants possibles. Ils accepteront quand même si vous mettez `</br>`, ils essaieront d'imaginer la place d'un `</div>` que vous auriez oublié, etc.

On a ici :

- Le doctype `<!DOCTYPE html>` : précise que c'est une page HTML5, c'est une exception dans sa forme de balise (commence avec un point d'exclamation).
- La balise `<html>` : nœud racine du code, début et fin.
- L'en-tête `<head>` : info générales sur la page comme titre, encodage.
- Le corps `<body>` : le corps de la page, là où on met tout.
- L'encodage `<meta charset="utf-8" />` : le codage utilisé pour le texte, l'UTF-8 permet d'afficher pratiquement presque tout donc choisissez-le.
- La balise `<title>` : le titre de la page tel qu'il apparaît dans l'onglet et sur google.

Le reste est une collection des balises de contenu :

- La balise `<p>` : un paragraphe, donc une façon d'organiser le contenu.
- La balise `<h1>` : est un titre, le plus important. En général, on s'en sert pour afficher le titre de la page au début de celle-ci. Après nous pouvons avoir plusieurs types de titres avec d'importance décroissante `<h2>` ... `<h5>`. Ne pas choisir le niveau de titre pour l'esthétique mais pour le sens : est-ce un titre important ou pas.
- La balise Commentaire : `<!-- Ceci est un commentaire -->`

Certaines balises ont un truc en plus. Par exemple au lieu de `<html>` nous pouvons utiliser `<html lang="fr">` : la balise de la ligne 2 qui est aussi fermée à la dernière ligne mais dans la balise ouvrant on trouve le `lang="fr"` qui est un **attribut** de la balise `html`.

Il peut y avoir de nombreux attributs dans une balise. Si nous partons d'une autre page qui liste toutes les balises (il y en a tellement !!!) : <https://fr.w3docs.com/apprendre-html/tableau-des-tags-html.html> ... on voit qu'il peut y avoir 18 attributs possibles pour la balise `<html>` + environ 75 attributs d'événement (mais encore certains attributs sont obsolètes).

Un code HTML mal formé pourra produire un rendu bizarre. Et un code HTML mal formé pourra fonctionner, puis en ajoutant un autre morceau de HTML bien formé dedans ou après, tout d'un coup le rendu ne fonctionne plus ... **Mais on a d'aide !**

Nos éditeurs aident souvent à signaler des problèmes (ex un `<div>` qui n'a pas son pair `</p>`).

Et il existe des sites web qui "vérifient" le HTML comme par exemple : https://validator.w3.org/#validate_by_input.

À la fin vous pouvez trouver des liens vers des ressources pour HTML.

À partir d'ici on a que des exemples des balises et du rendu.

Balises Corps & Texte, Listes, Images, Liens

C'est impossible de connaître tous les balises (<https://jaetheme.com/balises-html5/>), mais voici quelques unes qu'on utilise souvent pour référence. À noter qu'on voit pas ici la structure complète de la page (pas de `<!DOCTYPE html>` ni `<html>`) :

1. Corps & Texte

Environnement pour du contenu :

`<div></div>` : élément bloc (saut de ligne après)
`` : élément inline (s'insère dans texte)
`<p></p>` : contenu texte, élément bloc (paragraphe)

Saut de lignes :

Juste un saut de ligne dans le texte ne marche pas, faire un paragraphe `<p>`
Forcer le saut de ligne avec `
`

Emphase du texte :

`` : italique par défaut
`` : gras par défaut
`<mark></mark>` : surligne par défaut

Attention : la forme, ce n'est pas le but ! (le CSS gère la forme) mais de marquer le texte.
Ex : `` marque le texte comme «important» pour les moteurs de recherche...

```
<body>
  <!-- Corps de la page -->
  <p>
    <em> 1er paragraphe.</em>

    Le saut de ligne n'est <mark>pas visible</mark>.
  </p>
  <p> Ici au contraire...</p> <p>..on le voit bien </p>
  <div>
    Exemple de div
  </div>
  <div>
    Exemple de div 2 et <span> span a l'interieur. </span>
    On peut <br /> <strong>forcer</strong> les sauts de ligne aussi.
  </div>
</body>
```

1er paragraphe. Le saut de ligne n'est pas visible.

Ici au contraire...

..on le voit bien

Exemple de div

Exemple de div 2 et span a l'interieur. On peut
forcer les sauts de ligne aussi.

2. Listes

On a deux types de listes.

Non ordonnées :

`` délimite la liste (unordered list)

`` délimite un élément de la liste

Ordonnées :

`` délimite la liste (ordered list)

`` délimite un élément de la liste

```
<body>
  <h1>Les listes</h1>

  <h2>Liste de courses</h2>

  <ul>
    <li>Cadeaux de Noel</li>
    <ul>
      <li>Famille</li>
      <li>Profs?</li>
    </ul>
    <li>Gateaux</li>
    <li>Sodas</li>
  </ul>

  <h2>To-do list</h2>

  <ol>
    <li>Travailler sur son projet</li>
    <li>Remplir le compte-rendu</li>
    <li>Relire ses cours</li>
  </ol>

</body>
```

Les listes

Liste de courses

- Cadeaux de Noel
 - Famille
 - Profs?
- Gateaux
- Sodas

To-do list

1. Travailler sur son projet
2. Remplir le compte-rendu
3. Relire ses cours

3. Images (vidéo / audio)

Nous pouvons ajouter des images de différents façons :

1. `` : balise orpheline avec deux attributs obligatoires
`src` : source du fichier, soit sur le net soit en local
`alt` : texte alternatif pour décrire le contenu, utile pour les malvoyants et pour les moteurs de recherche

`title` : facultatif, donne une info-bulle

Mieux de la mettre dans un bloc (ex : `<div>`), convention W3C

2. On peut aussi mettre dans `<figure></figure>`

Cet environnement à un titre `<figcaption></figcaption>`

On a aussi audio et video qui fonctionnent de façon similaire, mais avec plus d'attributs possibles (`width`, `controls`, `autoplay`, `loop`, `preload`, etc.) Cf. [ici](#).

```
<body>
  <div>
    <img src = "images/framboisier.jpg"
        alt = "framboisier_delicieux"
        title = "miam">
  </div>

  <figure>
    <img src = "https://static.cuisineaz.com/240x192/i123605-
fondant-au-chocolat-noir-et-blanc.jpeg"
        alt = "chocolat">
    <figcaption>Du chocolat !</figcaption>
  </figure>
</body>
```



Du chocolat !

4. Liens

Comme on construit des pages web, les liens sont importantes !

Liens : balise `<a>`

Ce qui est dans la balise permet en cliquant d'aller sur le lien

Attribut `href` : indique où on va. Peut être lien absolu, relatif (même site), ou une ancre, voire ancre ailleurs. Si lien absolu remplacer `&` par `&`.

Pour une ancre : on utilise l'attribut `id` dans n'importe quelle balise. Ex : `id = "bas"`, on fait `href = "#bas"`.

Attribut `title` pour infobulle

Envoi mail : `href="mailto:votrenom@bidule.com"`

Téléchargement fichier : `href="fichier.extension"`

```
<body>
  <ul>
    <li> <a href = "#bas">Vers le bas!</a> </li>
    <li> <a href = "https://fr.wikipedia.org/wiki/La_Joconde">
      <img src = "images/monalisa.jpeg" alt = "Mona lisa">
      </a>
    </li>
    <li><a href = "images.html">Retour à la nourriture!</a></li>
    <li><a href = "images/chocolat.zip">Récupérer le chocolat!</
a></li>
    <li><a href = "mailto:votrenom@bidule.com">Envoyer un
message</a></li>
  </ul>

  <p id = "bas">En bas</p>

</body>
```

- [Vers le bas!](#)



- [Retour à la nourriture!](#)
- [Récupérer le chocolat!](#)
- [Envoyer un message?](#)

En bas

5. Tableau

Il y a des balises pour organiser le contenu sur un tableau. Avec les balises du type `<table><tr>` et `<th>` et `<td>`.

Tableaux :

`<table></table>` : contenu du tableau

`<tr></tr>` : début et fin d'une ligne de tableau

`<td></td>` : début et fin d'une cellule de tableau

`<th></th>` : cellule d'en-tête de tableau

`<caption></caption>` : Titre du tableau, avant l'en-tête, attribut `caption-side` qui peut être à `top` ou `bottom`

Attribut `colspan` ou `rowspan` : fusion de colonne ou ligne

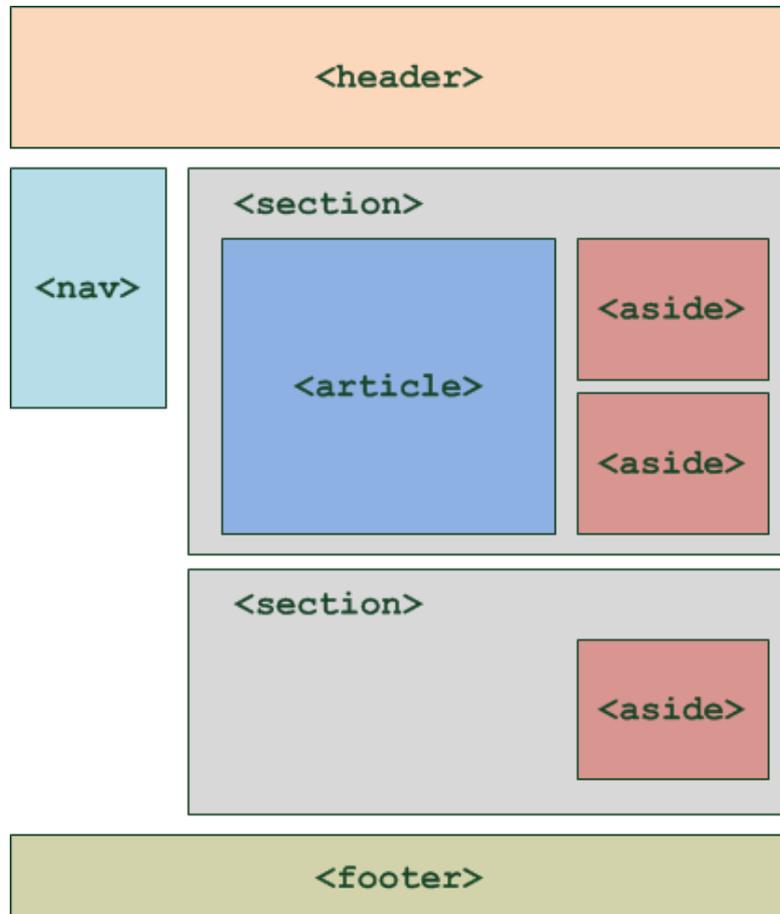
Autre possibilité de tableaux avec `<thead></thead>`, `<tfoot></tfoot>` et `<tbody></tbody>` (dans cet ordre !) qui contiennent les balises `<tr></tr>` et `<td></td>`.

```
<body>
  <!-- Corps de la page -->
  <table>
    <tr>
      <th>Type de film</th>
      <td>Horreur</td>
      <td>Documentaire</td>
      <td>Super-heros</td>
    </tr>
    <tr>
      <th>Pour enfants?</th>
      <td colspan = "2">Non, pas adapté</td>
      <td rowspan="2">Pour tous!</td>
    </tr>
    <tr>
      <th>Pour adolescents?</th>
      <td>Oui</td>
      <td>Vraiment?</td>
    </tr>
  </table>
</body>
```

Type de film	Horreur	Documentaire	Super-heros
Pour enfants?	Non, pas adapté		Pour tous!
Pour adolescents?	Oui	Vraiment?	

Autres

Et aussi autre balises structurantes (mais sans effet visuel sans l'utilisation de CSS, plus tard).



Liens utiles

1. Dans le site <https://fr.w3docs.com/> vous pouvez trouver des tutos en HTML et CSS en Français, y compris des petites exercices.
2. Toutes les balises / tags <https://fr.w3docs.com/apprendre-html/tableau-des-tags-html.html>
3. Cours de Pierre Giraud en Français sur l'HTML : <https://www.pierre-giraud.com/html-css-apprendre-coder-cours/element-balise-attribut/>
4. En Anglais, les tutos de w3d <https://www.w3schools.com/html/> qui gèrent des thématiques comme les videos, les cartes, les éléments interactives (buttons, listes), etc. Avec des exemples, des quizzes et des exercices.