

Objectifs :

- ⇒ Savoir à quoi sert le langage HTML
- ⇒ Connaître la structure générale d'un fichier HTML
- ⇒ Ecrire des pages web simples en HTML

**LE LANGAGE HTML ENSEIGNÉ  
DÈS LA PRIMAIRE**



## I - HTML, qu'est-ce que c'est ?

Le langage HTML (HyperText Markup Language) est un langage de balisage qui sert à représenter les pages hypertexte<sup>1</sup> (essentiellement celle du Web (WWW : World Wide Web)).

Le HTML décrit la structure d'une page Web.

Il est généralement associé avec :

- des feuilles de style CSS (Cascading Style Sheets) pour la mise en page (la présentation, le design)
- des scripts en langage JavaScript pour obtenir un comportement dynamique
- des contenus actifs comme des applets java ou flash

Les langages HTML et CSS font partie des langages de description (contrairement à JavaScript qui est un langage de programmation).

Il existe de nombreuses versions de ce langage qui s'est étoffé au fur et à mesure des années.

Dans la suite de ce cours, nous utiliserons la version HTML5 qui est actuellement la plus utilisée.

Comme pour la majorité des langages, le code source HTML s'écrit dans un fichier texte. L'extension du fichier sera `.htm` ou `.html`, mais il s'agit d'un fichier texte normal et donc éditable avec le bloc note ou notepad++.

Un fichier html correspond à une page (mais qui peut être aussi longue que l'on veut).

Version	Année
HTML 1.0	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2012
XHTML5	2013

## II - Structure d'un fichier HTML

Un fichier html contient l'information (le texte et les références aux images), mais aussi la structuration (définitions des différentes parties et sous-parties) ainsi qu'éventuellement les indications de mise en forme du document.

Il est important de noter qu'une page html doit pouvoir s'afficher sur des systèmes très différents, notamment en termes de polices de caractère et de taille d'écran. Les directives de mise en forme disponibles dans le langage prennent donc en compte ces contraintes, et le concepteur d'une page html doit également les avoir en tête, sinon l'affichage de sa page sur un autre système que celui qui a servi à l'écrire pourrait être assez surprenant.

### 1) Les balises

Le langage html est un langage de balise, c'est-à-dire qu'en plus du texte à afficher, le document peut contenir des balises qui servent à donner des indications sur la structure et la mise en forme au navigateur en charge d'afficher la page.

<sup>1</sup> L'hypertexte est la faculté d'insérer dans le texte des liens vers d'autres parties du texte ou vers d'autres textes. C'est le principe des liens (par défaut ils sont écrits en bleu et soulignés). On peut ainsi « naviguer » dans un document qui renvoie vers d'autres documents.

**Les balises sont toujours intégrées entre des chevrons ouvrant et fermant.**

**Elles sont toujours accompagnées d'une balise fermante (qui reprend le même nom, mais commence par le caractère '/') et peuvent contenir (dans la balise ouvrante), des attributs pour en préciser les caractéristiques.**

```
<balise attribut1="valeur1" attribut2="valeur2"> ...contenu... </balise>
```

Ex :

```
<style>
... définition du style ...
</style>
```

ou

```
<p>Ceci est un paragraphe qui se finit au point final.</p>
```

Cas particulier :

Certaines balises n'ont pas de contenu, dans ce cas elles sont à la fois ouvrantes et fermantes.

On écrit juste `<balise />`.

`` est une balise de type « `img` », c'est-à-dire image. Dans ce cas, il n'y a qu'une seule balise qui commence par `<img` et finit par `/>` (il y a toujours le signe '/' pour signifier la fermeture).

`<br />` est une balise forçant un passage à la ligne (br signifie BReak pour « line break »)

Les balises peuvent être imbriquées, mais elles ne doivent en aucun cas se chevaucher (une balise fermante d'un autre type ne doit pas se trouver entre la balise entrante et fermante).

CORRECT	INCORRECT
<pre>&lt;body&gt;   &lt;h1&gt;     Titre n°1   &lt;/h1&gt;   &lt;p&gt;     Paragraphe &lt;b&gt;numéro1&lt;/b&gt;   &lt;/p&gt; &lt;/body&gt;</pre>	<pre>&lt;body&gt;   &lt;h1&gt;     Titre n°1   &lt;p&gt;     Premier paragraphe   &lt;/h1&gt;   &lt;/p&gt; &lt;/body&gt;</pre>

La partie en rouge est incorrecte car la balise `<h1>` est fermée avant que celle du paragraphe (`<p>`) ne le soit. On aurait pu intervertir les deux éléments (`</h1>` et `</p>`) pour que le paragraphe soit inclus dans les balises `<h1> ... </h1>`.

Il existe deux types principaux de balises en html :

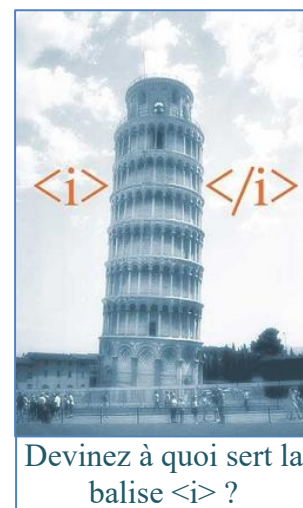
- Les balises « bloc »

Elles forment un bloc et à ce titre sont positionnables dans la page. **Elles sont toujours suivies d'un saut de ligne** (ou plus précisément elles s'affichent les unes en dessous des autres).

- Les balises « en ligne » (in-line)

Elles insèrent un contenu dans le flux de données (comme la balise `<img />` qui insère une image ou la balise `<a>` qui insère un lien hypertexte) ou modifient les propriétés du contenu (comme la balise `<b>` pour mettre en gras une portion de texte, ou `<sub>` pour mettre en indice).

**Elles ne sont pas suivies d'un retour à la ligne.**



Davantage de détails sur cet article (un peu trop) complet : <https://www.alsacreations.com/tuto/lire/530-La-structure-des-balises-bloc-et-en-ligne.html>.

**Q1** : A l'aide du bloc note ou de notepad++, écrire un texte simple (de trois ou quatre lignes sans aucune balise), puis sauvegarder le résultat dans un fichier avec l'extension .htm (ou .html). Ouvrez le fichier créé avec le navigateur. Que remarquez-vous ?

Ceci confirme que le fichier contient le texte, mais que la mise en forme (ici les passages à la ligne) est bien contenue dans les instructions des balises (que nous n'avons pas mises dans l'exemple précédent).

D'une manière générale, **les sauts de ligne sont ignorés dans un fichier html** et ne sont utiles qu'à la compréhension du lecteur humain.

## 2) Document minimal

Maintenant que nous avons vu les bases du langage, voyons la structure d'un document html :

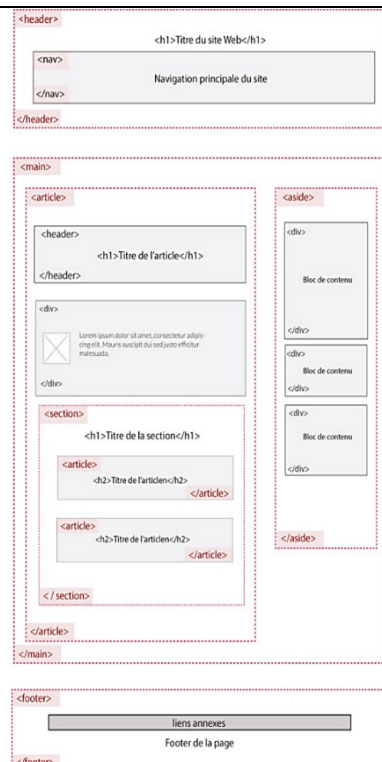
<code>&lt;!DOCTYPE html&gt;</code>	Tous les documents html doivent commencer par cette balise d'identification. Elle peut éventuellement indiquer la version de html utilisée pour l'écrire. La syntaxe de l'exemple ici correspond à du HTML 5.
<code>&lt;html lang="fr-FR"&gt;</code>	La balise html indique le début du document. L'attribut lang= donnant le langage de la page est optionnel.
<code>&lt;head&gt;</code>	Cette balise décrit l'en-tête du fichier qui va contenir les informations générales de la page (les métadonnées).
<code>&lt;meta charset="UTF-8"/&gt;</code> <code>&lt;meta name="robots"</code> <code>content="noindex"/&gt;</code>	Ici il s'agit de métadonnées sur le document, comme par exemple le type de codage de caractère, l'auteur, les instructions pour les robots d'indexation du web, ...
<code>&lt;title&gt;Premier document</code> <code>html&lt;/title&gt;</code>	C'est le titre de la page tel qu'il sera affiché en haut de la fenêtre par le navigateur.
<code>&lt;/head&gt;</code>	
<code>&lt;body&gt;</code>	
...	<i>Corps du document</i>
<code>&lt;/body&gt;</code>	
<code>&lt;/html&gt;</code>	On ferme toujours la balise html ouverte au début du document.

Le corps du document (body) est lui-même décomposé en différentes sections, comme indiqué sur le document ci-contre (voir le lien source pour le voir en plus grand avec des explications plus détaillées).

### Attention :

Les positions de ces sections ne sont pas du tout fixées (le `<aside>` par exemple peut être en haut ou à gauche). Ce sont les directives de placement de html (ou plutôt de CSS que nous verrons dans le prochain cours) qui vont leur donner leur position.

**Q2** : D'après vous, pourquoi les règles d'écriture en HTML sont-elles aussi strictes ?



### III - Ecriture d'une page web

Pour l'écriture de votre première page web, rendez-vous sur <http://jsbin.com>.

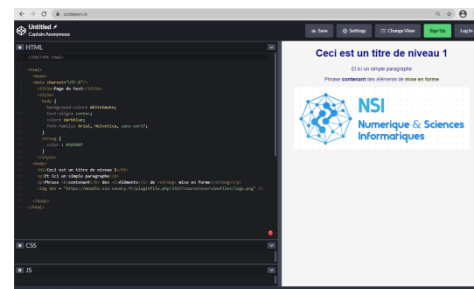
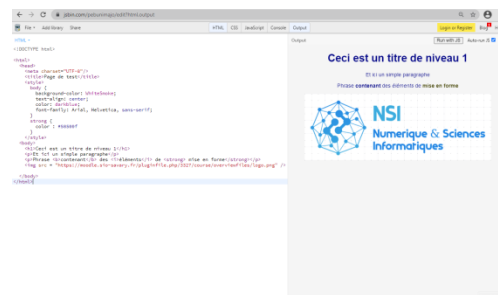
Vous pouvez écrire le code source html dans la fenêtre de gauche et le résultat apparaît en temps réel dans la fenêtre de droite.

Dans un premier temps, copiez-collez le contenu du fichier « Page\_demarrage.html » du répertoire de l'activité pour avoir une base de départ.

Une fois la page au point, on peut copier-coller le code source dans le bloc note (ou mieux dans notepad++) et l'enregistrer avec l'extension .html pour la visualiser ensuite avec un navigateur quelconque (par glisser/déposer du fichier ou par le menu ouvrir).

Vous pouvez aussi utiliser le site <https://codepen.io/pen/>.

On peut facilement changer la disposition suivant le format de sa page html.



Maintenant que nous sommes capables d'écrire du code HTML et de voir rapidement le résultat, nous allons pouvoir voir quelques éléments de la syntaxe HTML.

**Q3 :** D'après vous, les navigateurs interprètent-ils le code HTML de façon très stricte (aucune erreur acceptée), ou plus souple (essaye de s'adapter à un code incorrect) ? Après avoir répondu à la question, essayer d'écrire une page comportant des erreurs (balises fermantes absentes, erreurs dans les noms ou valeurs d'attributs, ...)

#### 1) Couleurs, distances et positions

##### a. Couleurs

On peut désigner la couleur de n'importe quel élément dans une page (fond, écriture, cadres...). Cela peut se faire de plusieurs façons différentes : par le nom (en anglais) de la couleur (ex : `color : green`) ou par la donnée de ses composantes RVB (2 méthodes, la plus utilisée étant en hexadécimal précédé par un signe « # ». Ex : `color : #A2C022`).

Voir <http://www.css-faciles.com/couleurs-css.php> pour davantage d'explications.

##### b. Distances

De même, les distances peuvent être exprimées avec différentes unités.

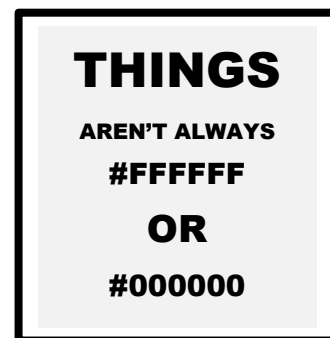
Voici une petite liste des plus fréquentes :

- Les unités relatives : `em` (relative à la taille de la police utilisée) et `%` (relative à la taille du contenant)
- Les unités absolues : `px` (pixel), `cm` (centimètre), `mm` (millimètre), `in` (inch : pouce = 2,54 cm), `pt` (point = 1/72in)...

Pour un rendu sur écran, **les unités les plus adaptés seront `em` et `%`** qui permettent de s'adapter à la taille du support (ne pas oublier que les fenêtres sont redimensionnables) **ainsi que `px`** (pour des positionnements précis).

Pour un rendu sur imprimante, les unités absolues `cm`, `mm` et `in` seront plus adaptées.

Voir <https://www.w3.org/Style/Examples/007/units.fr.html> pour davantage d'explications.



### c. Positions

Comme dans les traitements de texte, HTML propose 4 types de positions pour le texte ou les images : alignés à gauche, à droite, centré ou justifié (aligné des deux côtés).

En HTML, pour les versions inférieures à 5<sup>2</sup>, cela se fait simplement en ajoutant l'attribut `align = "type"` à la balise ouvrante.

Ex : `<p align = "left">Ceci est un paragraphe aligné à gauche</p>`

Pour aller plus loin sur la position des éléments dans la page, voir <http://fr.learnlayout.com/position.html> ou [https://www.w3schools.com/cssref/pr\\_class\\_position.asp](https://www.w3schools.com/cssref/pr_class_position.asp) (nous ne pourrions vraiment exploiter ces informations que lorsque nous aurons vu CSS dans le prochain chapitre).

## 2) Écriture d'un tableau

**Q4** : Créer une page web (vous pouvez repartir de la page d'exemple du site) dans lequel doit apparaître un tableau de deux lignes et 3 colonnes d'images prises sur le web sur un thème à choisir. Dans un premier temps faites un tableau affichant uniquement du texte dans les cellules, puis ajoutez les images.

Pour vous aider dans l'écriture d'un tableau, vous pouvez consulter le site <http://cyberzoide.developpez.com/html/table.php3> qui donne des explications assez claires.

Même si HTML n'est pas un langage de programmation, vous pouvez ajouter des commentaires dans la page html (qui ne seront pas interprétés par le navigateur). Ils permettent par exemple de signaler que telle portion correspond à tel affichage.

Les commentaires doivent commencer par « `<!--` » et finir par « `-->` »

## IV - Et maintenant un site ...

Nous en savons maintenant assez pour commencer à écrire des pages HTML. Mais pour écrire un site, il faut bien sûr créer des liens entre les différentes pages. Cela se fait simplement avec les balises `<a>` :

Syntaxe :

`<a href = "URL de la page">texte à afficher (qui sera bleu et souligné)</a>`

L'URL<sup>3</sup> peut être de type absolu (ex : « <http://monlycee.net/> ») ou relative (ex : « page2.html » ou « sous-repertoire/page3.html »)

**Attention** : Eviter d'utiliser des caractères spéciaux (hors lettres et chiffres des 128 premiers caractères ASCII) ou même des espaces (utiliser le sous-tiret '\_' à la place) dans les noms de ressources (fichiers, pages web).

Pour un site local, on utilisera donc des URL relatives en mettant tous les fichiers du site dans un même répertoire sur le disque.

**Q5** : Ecrire un site web présentant votre mini-projet. Ce site devra comporter au moins 3 pages HTML distinctes.

*Il est conseillé de faire des copier-coller du texte de son dossier et de se contenter de faire la mise en forme avec les balises html. On fera attention par exemple à bien respecter la hiérarchie des titres.*

<sup>2</sup> Pour la version 5 de HTML, cet alignement doit obligatoirement se faire avec des propriétés CSS que nous verrons dans un prochain cours. Voir <https://www.zonecss.fr/proprietes-css/text-align-css.html>.

<sup>3</sup> Uniform Resource Locator : une chaîne identifiant une ressource et éventuellement le moyen d'y accéder. Une 'adresse web' est une URL. Voir <http://blog.neocamino.com/qu-est-ce-qu-une-url/> (explication simple) ou <http://www.commentcamarche.com/contents/542-qu-est-ce-qu-une-url> (explication plus complète).



## V - L'inspecteur du navigateur

Pour aider au développement web, les navigateurs modernes intègrent différents outils.

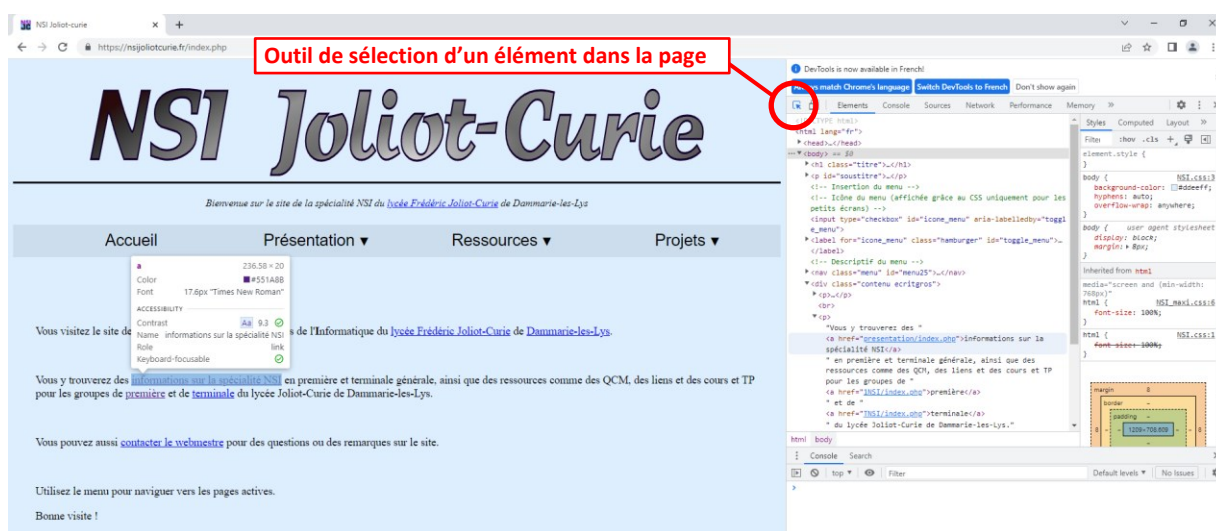
La première possibilité offerte, est celle d'afficher le code source de la page. Cela se fait généralement avec un clic-droit dans la page et « Code source de la page ».

**Q6 :** Visiter les sites [https://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language), <https://www.powtoon.com/>, <https://genial.ly/fr/> et afficher le code source de la page. Que remarquez-vous ?

Naviguer dans une page web réelle n'est pas toujours simple. Pour certaines, elles ne sont clairement pas faites pour être lues par des humains (parfois volontairement<sup>4</sup>) et sont souvent générées automatiquement par des algorithmes ou bien les espaces, sauts de lignes, indentations, etc... ont été supprimés pour rendre le fichier html plus petit (et donc plus rapide à charger).

Pour faciliter les choses, il existe également un inspecteur de page web qui est un peu l'équivalent du débogueur pour les langages de programmation (il fait d'ailleurs office de débogueur javascript comme nous le verrons dans un prochain chapitre).

On peut l'appeler simplement en utilisant la touche F12 ou un clic-droit et « Inspecter l'élément » (sous Firefox).



Ses usages sont nombreux. Pour l'instant nous pouvons nous contenter de cibler un élément de la page avec l'outil flèche (celui en haut à gauche de la fenêtre de l'inspecteur). L'inspecteur nous ouvre alors le code source à l'endroit où est décrit l'élément ciblé. On a alors la possibilité de le modifier directement, voire de le supprimer<sup>5</sup>.

**Q7 :** Inspecter la page <https://fr.wikihow.com/se-servir-de-l%27option-Examiner-l%27C3%A9l%27C3%A9ment-dans-Mozilla-Firefox> et déterminer les classes associées au bouton `Ctrl` affiché dans le 2 de la partie 1 (« Inspecter des éléments »). Essayer de modifier les propriétés du bouton (dans la fenêtre « style ») pour qu'il apparaisse avec un fond rouge.

### Références :

<http://www.pierre-giraud.com/html-css/cours-complet/cours-html-css-presentation.php>

<http://j-willette.developpez.com/tutoriels/html/les-bases-du-html/>

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

<https://jaetheme.com/balises-html5/>

Vidéo sur l'inspecteur de code : <https://www.youtube.com/watch?v=Yw0QBFEExOb8>

<sup>4</sup> On parle alors d'[obfuscation de code](#) : le but est de rendre le code inintelligible par un humain pour éviter que d'autres puissent l'utiliser.

<sup>5</sup> Evidemment, ces modifications ne seront valables que pour la page en mémoire et le simple fait de la recharger effacera les modifications.