

# HTML

## INTRODUCTION

L'HTML est un langage universel qui permet de créer des sites web. Il définit la structure et organise une page web. Par un système de balises, qui déterminent le contenu, il est ensuite traduit par le navigateur qui affiche le résultat à l'écran.

## L'éditeur de texte

Un éditeur de texte est un programme dédié à l'écriture du code. Il en existe de nombreux, ici nous utiliserons Sublime text qui est l'un des plus utilisés, car très simple et pratique.

## Les navigateurs

Le navigateur est l'intermédiaire entre le code et ce qui est affiché, c'est lui qui se charge de traduire le code pour obtenir le rendu souhaité à l'écran.

Il en existe plusieurs dont Mozilla Firefox, Google Chrome, Edge (anciennement internet explorer) etc. Tous peuvent donner une traduction différente du code, il est possible d'avoir des différences d'affichage en fonction du navigateur. Il est donc nécessaire de vérifier le bon affichage d'un site sur différent navigateur.

## En savoir plus

HTML signifie HyperText Markup Language.

Ce langage est inventé en 1991 par Tim Berners-Lee. Il a également créé le World Wide Web Consortium (W3C), qui définit les normes du web et qui en assure son évolution avec de nouvelles versions. Par exemple, au début il n'était pas possible d'afficher des images. L'HTML 5 est la dernière version qui date de 2014. Elle apporte des améliorations notamment sémantiques et de nombreuses fonctionnalités.

## CRÉER SON FICHER HTML

Pour créer son fichier HTML, depuis Sublime text, il faut simplement enregistrer le document avec l'extension **.html**

Maintenant que le fichier est créé nous allons ajouter la structure qui définit qu'il s'agit bien d'une page web :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>HTML initiation</title>
  </head>
  <body>
    <!-- Contenu de la page HTML -->
  </body>
</html>
```

- **doctype** : indique qu'il s'agit du langage HTML.
- **html** : balise qui contient l'ensemble du code de la page.
- **head** : balise qui contient des informations sur la page, ces informations ne seront pas affichées dans le navigateur.
- **meta** : cette balise avec son attribut **charset** définit l'encodage de la page, c'est-à-dire qu'il s'assure du bon affichage des caractères spéciaux (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

- `title` : indique le titre de la page web. (élément très important, car tout site doit avoir un titre décrivant ce qu'il contient. Il nomme également l'onglet de votre navigateur.
- `body` : la balise `body` représente le contenu du document que l'on veut voir afficher dans la page web de notre navigateur.

## LA SYNTHAXE

Les balises s'écrivent avec la syntaxe suivante :



## LES BALISES

Il existe deux types de balise :

- **Les balises dites « par paires »** : elles ont une balise d'ouverture et une balise de fermeture, entre les deux elles contiennent des éléments comme du texte, d'autres balises, etc.



- **Les balises dites « orphelines »** : contrairement aux balises « par paires » n'ont pas besoin de fermeture.

Dans la construction de la page on peut donc observer plusieurs balises « par paires » comme la balise `<html>` qui contient l'ensemble des autres balises dont la balise `<head>` et la balise `<body>`.

La balise `<head>` contient elle-même deux autres balises, la balise `<meta>` qui est une balise « orpheline » et la balise `<title>` qui contient du texte.

**À noter : Chaque balise à une utilité qui lui est propre.**

## STRUCTURE D'UNE PAGE

### La notion de parent et enfant

En HTML on a vu que l'on peut imbriquer des balises les unes dans les autres, on parle alors d'élément « **parent** » et élément « **enfant** ».

Par exemple, si on reprend la structure et la construction de la page :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>HTML initiation</title>
  </head>
  <body>
    <!-- Contenu de La page HTML -->
  </body>
</html>
```

La balise `<html>` est l'élément « parent » de `<head>` et de `<body>` qui sont alors considérés comme les éléments « enfant » de la balise `<html>`.

La balise `<head>` est aussi parente des balises `<meta>` et `<title>`.

### La sémantique

Dans le langage HTML chaque balise a son importance et une utilité qui lui est propre. Il existe donc un grand nombre de balise différente. **Il est important de respecter le bon usage des balises.**

### L'indentation

Lorsque l'on écrit du code, on pratique la méthode de l'indentation, c'est un procédé qui consiste à décaler l'alignement d'un paragraphe par rapport aux autres. Ce qui est très utile pour se repérer surtout entre les parents et les enfants de la structure.

### Les commentaires

Dans notre fichier, on peut également écrire des commentaires. Pour nous aider à nous repérer dans la page, ou pour faciliter le travail en groupe. Ils ne s'affichent pas à l'écran.

On les écrit comme ceci :

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

## ORGANISER SON TEXTE

### Les paragraphes

La balise `<p>` est une balise de paragraphe, elle doit donc contenir du texte. Pour forcer le passage à la ligne dans un même paragraphe on utilise la balise `<br/>`.

```
<p>mon texte
ne va pas à la ligne</p>

<p>Par contre, <br/> là, mon texte passe à la ligne</p>
```

## Les titres

Pour les titres il y a une particularité. En effet il existe 6 niveaux de titre d'importances différentes `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` et `<h6>`. Du plus important au moins important, le `<h6>` est donc le titre le moins important. Le `<h1>` quant à lui est considéré comme le titre de la page, cette balise ne peut-être utilisée qu'une seule fois dans une page. En général elle définit le sujet de la page.

Exemple de structure :

```
<h1>Ceci est mon portfolio</h1>
<h2>Mes créations</h2>
  <p>texte descriptif</p>
  <h3>Sous-titre</h3>
    <p>texte descriptif</p>
    <h4>Catégorie</h4>
    <h5>Sous-catégorie</h5>
```

## Faire une liste

Pour créer une liste on utilise les balises `<ul>` et `<li>`.

- `<ul>` : est la balise parente, elle définit l'ensemble de la liste
- `<li>` : la balise enfant, définit chaque élément de cette liste.

Exemple :

```
<ul>
  <li>Premier élément de ma liste.</li>
  <li>Deuxième élément de ma liste.</li>
  <li>Troisième élément de ma liste.</li>
</ul>
```

Si on veut une liste ordonnée, on remplace alors la balise `<ul>` par la balise `<ol>`.

Liste désordonnée avec la balise `<ul>`

- Premier élément de ma liste.
- Deuxième élément de ma liste.
- Troisième élément de ma liste.

Liste ordonnée avec la balise `<ol>`

1. Premier élément de ma liste.
2. Deuxième élément de ma liste.
3. Troisième élément de ma liste.

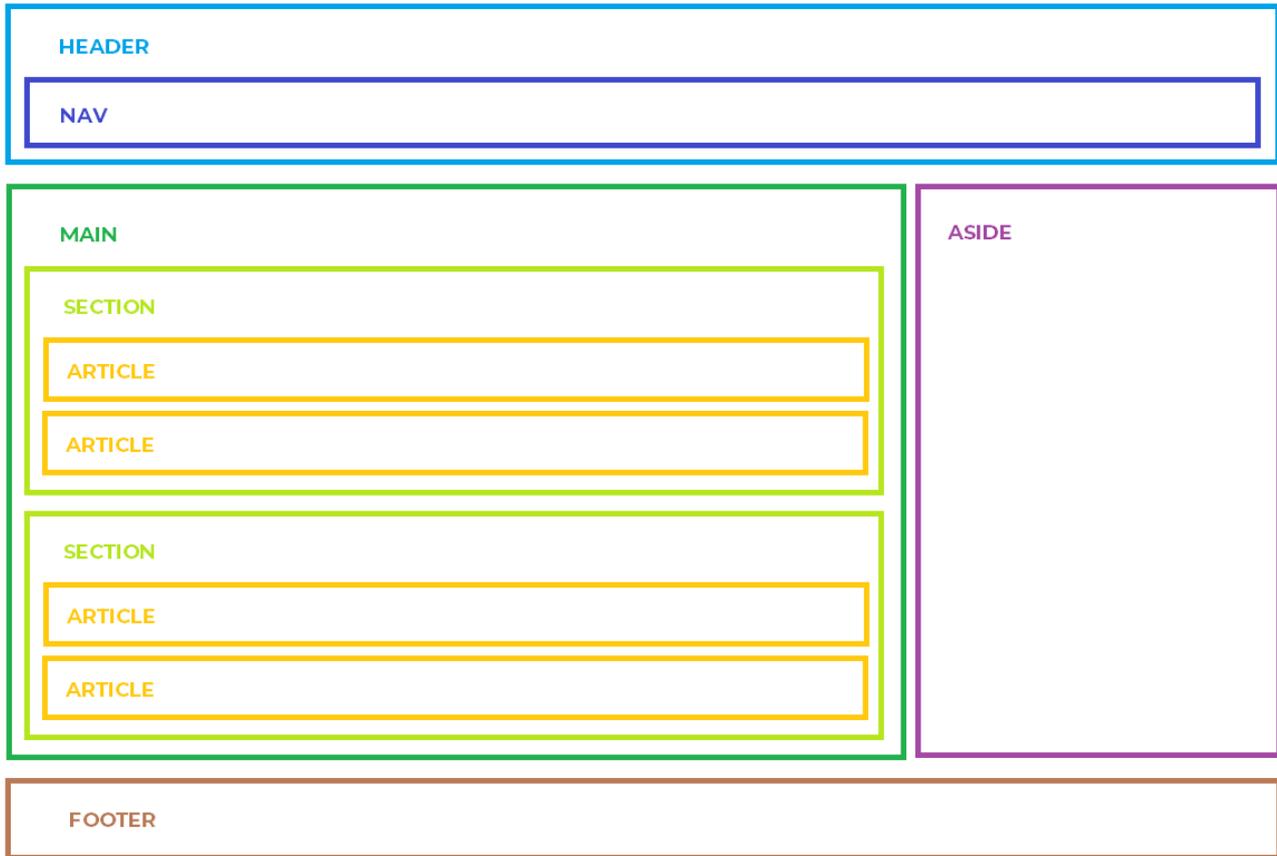
## SECTIONS D'UN SITE WEB

Un site web est divisé en plusieurs parties, chacune aillant sa propre balise :

- `<header>` : l'entête du site.
- `<nav>` : la navigation, ou plus communément appelé le menu.
- `<main>` : contenu principal de la page.
- `<section>` : découpage des différentes parties de l'article.

- `<article>` : élément de contenu complet et autonome.
- `<aside>` : élément d'un site qui a destination de rester sur le côté du site.
- `<footer>` : le pied de page.

Exemple de structure :

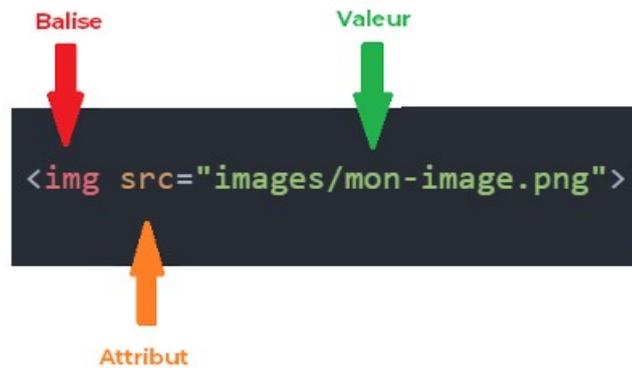


## LES ATTRIBUTS

Les balises peuvent avoir ce que l'on appelle des « **attributs** », ce sont des paramètres qui viennent ajouter des informations à celle-ci. Les attributs ont besoin d'une « **valeur** ». Certaines balises sont inutiles si on ne leur renseigne pas d'attribut, c'est le cas par exemple des liens ou des images.

## Les images

La balise `<img>` permet d'afficher une image, mais elle a besoin qu'on lui renseigne un attribut `src` (source) pour lui indiquer la bonne image à afficher. Pour cela on note alors la valeur qui est composée de son « **chemin** » (l'endroit où l'image se trouve), suivi du nom de l'image, avec son extension (en général **.jpg** ou **.png**). Comme dans l'exemple ci-dessous :

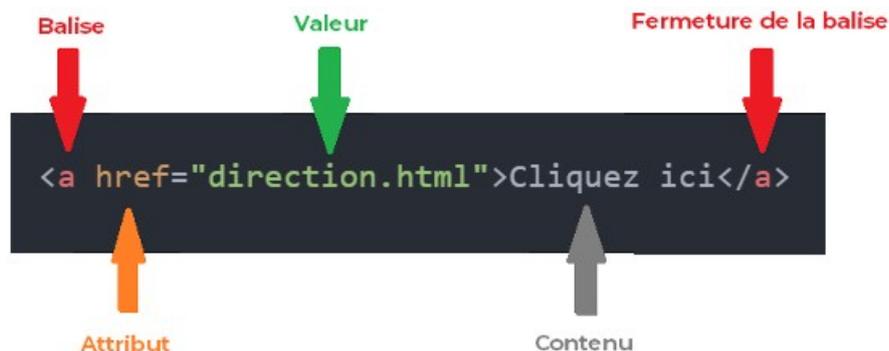


« images » indique que l'image est dans un fichier nommé « images ». « mon-image » est le nom de l'image à afficher. « .png » est le type d'image.

Il existe de nombreux attributs différents et de nombreuses balises qui ont besoin d'attributs pour fonctionner.

## Les liens

La balise `<a>` indique qu'il y a un lien cliquable. On peut l'utiliser sur du texte ou des images. Cependant, si on ne lui renseigne pas où le lien doit rediriger, alors rien ne se passe. C'est donc le rôle de l'attribut `href`, qui contiendra en valeur l'url d'une page HTML, d'un fichier, etc.



`direction.html` est la destination du lien, ici c'est une page HTML.

## Les types de liens

Il existe différents types de lien :

- **Les URL** : ce sont des adresses de page sur internet (ex : <https://www.wikipedia.org>).
- **Les pages** : ce sont des pages web faisant partie d'un même site web, il faut donc indiquer le chemin de la page (ex : `mon-dossier/ma-page.html`).
- **Les fichiers** : ce sont des documents inclus dans un site web, que l'on peut ouvrir ou télécharger. Il faut également indiquer un chemin (ex : `mon-dossier/mon-fichier.pdf`).
- **Les ancres** : ce sont des positions précises dans notre page web, par exemple pour rediriger vers une section en particulier. Elles sont appelées grâce à des identifiants (ex : `#mon-identifiant`).

## Précision sur les chemins

Lorsque un fichier est directement à côté d'un autre, son chemin est simple on note uniquement son nom et son extension. Par contre, si un fichier se trouve dans un dossier on doit d'abord noter le nom du dossier comme dans l'exemple suivant :

"mon-image.png" : l'image se trouve directement à côté de mon fichier (on dit alors qu'il est à la racine du fichier).

"images/mon-image.png" : l'image se trouve dans un dossier nommé « images » à côté de mon fichier.

"../mon-image.png" : l'image se trouve à l'extérieur du dossier dans lequel se trouve mon fichier.

On peut donc obtenir un chemin complexe comme : "../galleries/images/mon-image.png"

## PARTICULARITÉS À SAVOIR

### Les balises **block** et les balises **inline**

Les balises HTML ont un type de « disposition » défini par défaut, il est soit de type **block** comme les balises `<p>`, `<ul>`, `<header>`, etc. Soit de type **inline** comme les balises `<a>`, `<img>`, `<span>`, etc.

- **block** : L'élément prend la largeur entière de son élément parent qu'importe la longueur de son contenu.
- **inline** : L'élément à une largeur égale à la longueur de son contenu.

Balise de type **block**

Balise de type **inline**

### Les balises universelles

Il existe deux balises qui n'ont aucun sens en particulier :

- La balise `<span></span>` est de type **inline**
- La balise `<div></div>` est de type **block**

Étant donné qu'elles sont génériques, elles sont très pratiques. Mais attention à ne pas en abuser, en HTML **il est important de respecter la sémantique.**