

## ❁ Chapitre 6 ❁

# Le langage HTML et CSS

## I. Qu'est ce que le HTML ?

Quand on consulte un site web, notre ordinateur se connecte à un serveur distant et celui-ci lui envoie une page web, que le navigateur va pouvoir afficher. En général la page est constituée de plusieurs fichiers :

- un fichier HTML (il existe d'autres formats qui seront étudiés plus tard : php, jsp, asp, ...)
- un fichier CSS
- un fichier JS
- des images, du sons, ...

Le plus important est celui dont l'extension est « .html » ou « .htm ». C'est lui qui contient toute la structure de la page et qui va indiquer au navigateur comment afficher la page. La première page d'un site doit s'appeler index.html!

La structure d'une page web est décrite à l'aide d'un langage de **balises** appelé **HTML** (HyperText Markup Language). On peut facilement afficher ces balises quand on navigue sur un site internet (on peut utiliser le raccourci « ctrl+u » sous firefox par exemple).

## II. Les balises

Une balise est un mot-clef permettant d'indiquer au navigateur ce qu'il doit faire avec le contenu. Les balises contrôlent tout. Elles permettent de gérer la structure, les images, les liens, la police d'affichage utilisée, les titres, ...

On écrit <balise> pour ouvrir une balise et </balise> pour la fermer. De manière générale, toute balise ouverte doit être fermée.

Voici la structure minimale (squelette) de toute page HTML :

```
1 <!DOCTYPE html> <!-- langage utilisé, html pour html5 -->
2 <html lang="fr"> <!-- début de la page web -->
3   <head> <!-- En tête de la page, la balise n'est pas affichée, mais elle
      permet de donner des informations telles que le titre de la page ou la
      police de caractères utilisée -->
4     <title> Le html en 1ère NSI </title>
5     <meta charset="utf-8"/>
6   </head>
7   <body> <!-- Corps de la page, c'est ici que l'on écrit tout ce qui doit être
      affiché -->
8     <p>
9       Bienvenue en cours de NSI.
10    </p>
11  </body>
12 </html>
```

Les balises <!DOCTYPE>, <html></html>, <head></head> et <body></body> sont obligatoires dans tout document HTML.

On peut remarquer que certaines balises particulières ne se ferment pas (comme la balise <meta/>). Dans ce cas il y a un « / » avant le « > » ?

Le nom d'une balise s'appelle un élément (exemple ici body, p ou meta). Certains éléments peuvent avoir des attributs pour définir des paramètres. Dans ce cas les attributs sont toujours donnés entre guillemets " " ou apostrophes ' '. De plus des balises peuvent être imbriquées les une dans les autres mais pas se chevaucher!

### Exemple 1:

```
1 <p>
2   Bienvenue en <em> cours </em> de NSI.<br/>
3   Ici les balises sont correctement imbriquées
4 </p>
```

```

1 <p>
2   Bienvenue en <em> cours de NSI.<br/>
3   Problème, la balise em est fermé après la balise p !!
4 </p></em>

```

### III. Block / Inline

Voici les éléments les plus courants mais il en existe d'autres. Ils peuvent être regroupés en deux catégories :

- les éléments de bloc (**block** en anglais) qui s'affichent comme un bloc (saut de ligne avant et après automatiquement),
- les éléments en ligne (**inline** en anglais) qui s'affichent dans la ligne courante. Un élément inline doit obligatoirement être à l'intérieur d'un bloc.

#### Balises de type block :

```

1 ...
2 <body>
3   <p>
4     Un paragraphe
5   </p>
6   <h1>
7     Un titre de premier niveau
8   </h1>
9   <h2>
10    Un titre de niveau 2
11  </h2>
12  <ul> <!-- une liste à puce (Unordered List)-->
13    <li> Premier élément de la liste à puce </li>
14    <li> Deuxième élément de la liste à puce </li>
15  </ul>
16  <ol> <!-- une liste numérotée (Ordered List)-->
17    <li> Premier élément de la liste numérotée </li>
18    <li> Deuxième élément de la liste numérotée </li>
19  </ol>
20  <div>
21    Un bloc quelconque, utile pour la mise en forme
22  </div>
23 </body>
24 </html>

```

Pour visualiser la page, cliquer [ici](#)

#### Balises type inline :

Balise	Fonction	Attributs utiles
 	Retour à la ligne	
<img/>	Permet d'insérer une image	<ul style="list-style-type: none"> <li>• src = 'emplacement de l'image'</li> <li>• alt = 'texte à afficher si problème'</li> </ul>
<a>	Ancre, permet de créer un lien	<ul style="list-style-type: none"> <li>• name = 'nom de l'ancre'</li> <li>• href = 'lien'</li> </ul>

## Exemple 2:

```

1  ...
2  <body>
3    <h1>
4      Un exemple avec des éléments inline
5    </h1>
6    <p>
7      Voici un paragraphe. Pas d'élément &lt;br/&gt; donc
8      pas de retour à la ligne.<br/>
9      Après la balise une nouvelle ligne commence.<br/>
10     Une ligne avec une image : <img src='https://
11       encrypted-tbn0.gstatic.com/images?q=tbn%3
12       AANd9GcR0TQuLKb3LDxIrGiuC6wowhIxugJ30Y46bHA&
13       usqp=CAU' alt='Mon image' /><br/>
14     Et maintenant un lien vers un autre site : <a href='
15       https://www.axelnax.fr/'> Le site du prof ! </a>
16   </p>
17 </body>
18 </html>

```

Pour visualiser la page, cliquer [ici](#)

## IV. Et le CSS alors?

Un fichier '.css' (cascading style sheet) accompagne souvent une page html. Il sert à définir tout ce qui concerne la forme du document (là où la partie HTML s'occupe plutôt du contenu). Il gère donc la taille de la police, la couleur, le positionnement des blocs, des images, ...

Exemple de page internet **sans** CSS et **avec** CSS.

Un fichier contenant du code CSS permet de donner des informations sur la forme que doit avoir une page internet (couleur de la police, position et taille des images, fond de page, menus, ...). Ce fichier peut avoir le nom que l'on souhaite (on utilisera néanmoins préféremment l'extension .css). Dans notre exemple ce fichier s'appellera « style.css ».

Modifications à apporter dans le fichier HTML :

- il faut donner le nom (et le chemin si nécessaire) du(des) fichier(s) CSS à utiliser. Cela se fait à l'aide de la balise suivante dans l'en-tête : `<link href="style.css" rel="stylesheet" type="text/css">`
- on peut maintenant utiliser 2 nouveaux attributs dans les balises HTML,
  - `id="nomUnique"` : permet de donner un nom unique à un élément pour lui appliquer un style particulier,
  - `class="nom"` : permet de donner le même nom à un ensemble d'éléments auxquels on veut appliquer le même style.

Dans le fichier CSS on peut maintenant créer un ensemble de règles à appliquer. Une règle est composée d'un sélecteur (l'objet auquel la règle doit s'appliquer) et de déclarations de propriétés. Les déclarations se trouvent entre accolades et les propriétés sont séparées par des points-virgules.

Il y a beaucoup de sélecteurs différents, les plus courants sont :

- le nom d'un type d'éléments (ex : `html`, `p`, `body`, `h1`, `img`, ...): la règle s'applique à tous les éléments de la page de ce type,
- le nom d'un id précédé d'un `#` : la règle s'applique au seul élément ayant cet id,
- le nom d'une classe précédé d'un point : la règle s'applique à tous les éléments ayant cette classe, qu'importe le type.

On peut mélanger plusieurs sélecteurs pour une même règle.

Il existe une très grande variété de propriétés, la liste dépendant du sélecteur choisi!

### Exemple 3:

```

1 <!DOCTYPE html> <!-- langage utilisé, html pour html5 -->
2 <html lang="fr"> <!-- début de la page web -->
3   <head> <!-- En tête de la page, la balise n'est pas affichée, mais elle permet
4     de donner des informations telles que le titre de la page ou la police de
5     caractères utilisée -->
6     <title> Le CSS en 1ère NSI</title>
7     <meta charset="utf-8"/>
8     <link rel="stylesheet" href="style.css"> <!-- import le CSS de la page
9     cible -->
10  </head>
11  <body>
12  <h1>Un joli titre centrée, en rouge et en italique.</h1>
13  <p>
14    Un beau paragraphe générique.
15  </p>
16  <p class="important">
17    Un paragraphe de la classe "important" que je vais faire apparaitre en
18    gras.
19  </p>
20  <p>
21    Un paragraphe normal dans lequel <a href="https://www.axelnax.fr/" class="
22    important"> le lien est important </a>.
23  </p>
24  <p id="fin">
25    Un paragraphe unique avec le texte en bleu.
26  </p>
27  
  </body>
</html>

```

```

1 body{
2   font-family: helvetica, arial, sans-serif;
3   margin: 2em;
4   background-color: beige;
5 }
6 h1{ /* tous les titres h1 */
7   font-style: italic;
8   text-align: center;
9   color: red;
10 }
11 p{ /* tous les paragraphes */
12   margin-left: 280px;
13 }
14 #fin{ /* l'unique élément d'id fin */
15   color: darkblue;
16 }
17 .important{ /* tous les éléments de la classe important */
18   font-weight: bold;
19 }
20 img#logo{ /* élément img avec l'id logo */
21   display: block;
22   border-radius: 20px;
23   width: 150px;
24   margin-left: auto;
25   left: 20px;
26   top: 20px;
27   position: fixed;
28 }

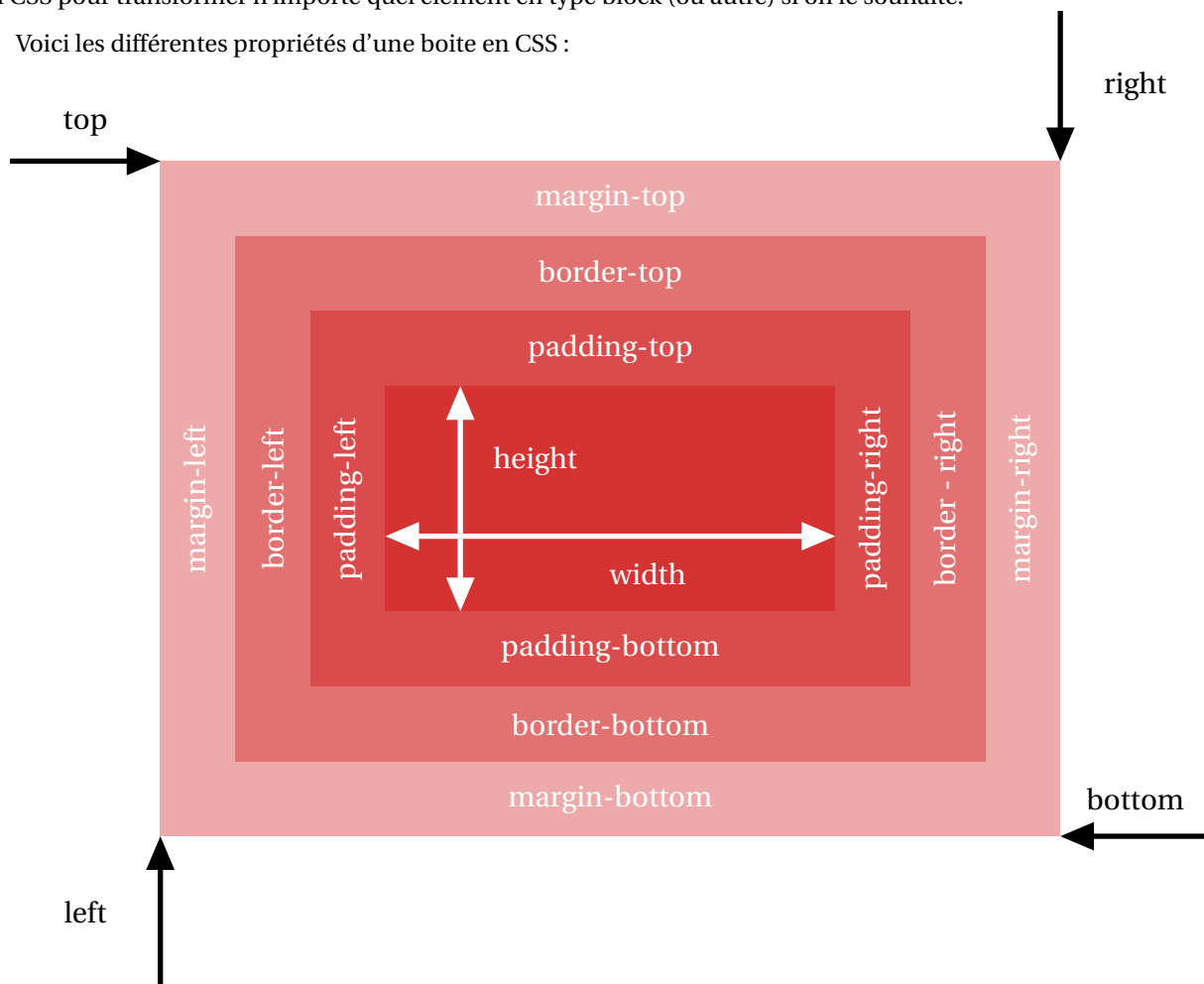
```

Rendu des pages [sans CSS](#) et [avec CSS](#).

## V. CSS et les boîtes

Tous les éléments HTML de type block sont manipulés en tant que boîtes par CSS. Il existe une propriété display en CSS pour transformer n'importe quel élément en type block (ou autre) si on le souhaite.

Voici les différentes propriétés d'une boîte en CSS :

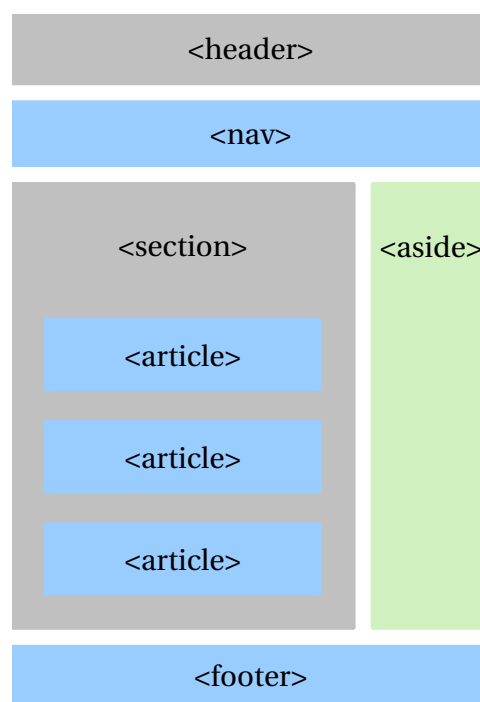


La zone centrale représente le contenu, le padding est l'écart entre le contenu et la bordure, le margin est l'écart avec le bord extérieur de la boîte.

La position d'un élément peut être définie avec la propriété position. Elle peut être relative à la place de l'élément dans le flux de la page ou absolue, c'est à dire définie directement par rapport au bord de la fenêtre.

L'ensemble de ces propriétés, associées aux éléments div, header, nav, section, article, aside et footer permet de construire une belle mise en page en HTML.

Voici un exemple avec un en-tête de page, une barre de navigation, un pied de page et une colonne de menus (à droite de la page dans cet exemple).



## Exemple 4:

Une version sans la barre de navigation :

Le code CSS correspondant :

```

1  html {
2    height: 100%;
3    box-sizing: border-box;
4  }
5  body {
6    box-sizing: border-box;
7    font-family: helvetica, arial
8      , sans-serif;
9    margin: 0em;
10   background-color: beige;
11   min-height: 100%;
12   padding-bottom: 30px;
13 }
14 section {
15   padding: 20px;
16   height: auto;
17   float: left;
18   width: 85%;
19 }
20 header {
21   height : 5%;
22   padding: 20px;
23   text-align: center;
24   background-color: grey;
25 }
26 footer {
27   height : 30px;
28   padding: 10px;
29   text-align: center;
30   background-color: grey;
31   position: absolute;
32   left: 0;
33   right: 0;
34   bottom: 0;
35   clear: both;
36 }
37 aside {
38   padding: 10px;
39   float: right;
40   width: 10%;
41 }
42 h4 {
43   text-align: center;

```

Et le code HTML associé :

```

1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <title>NSI : CSS</title>
5      <meta charset="utf-8" />
6      <link rel="stylesheet" href="style1.css" /
7    </head>
8    <body>
9      <header>
10     Une jolie bannière, un menu, ...
11   </header>
12   <section>
13     <article>
14       <h1>Contenu 1</h1>
15       Voici le contenu de la page, on peut
16         le faire changer grâce aux liens
17         par exemple.
18     </article>
19     <article>
20       <h1>Contenu 2</h1>
21       Il peut y a voir plusieurs articles
22     </article>
23     <article>
24       <h1>Contenu 3</h1>
25       L'informaticien logiciel et
26         programmation.<br />
27       Le travail d'un ...
28       Il propose également des ...
29     <br />
30     <br />
31     Source : wikipédia
32   </article>
33 </section>
34 <aside>
35   <h4>Un menu</h4>
36   <ul>
37     <li><a href="">Page 1</a></li>
38     <li><a href="">Page 2</a></li>
39     <li><a href="">Page 3</a></li>
40   </ul>
41 </aside>
42 <footer>
43   Un pied de page ...
44 </footer>
45 </body>
46 </html>

```

On peut voir le rendu [ici](#)