

Chap. 1 – HTML et CSS

Dans ce chapitre, tu vas :

- découvrir le `HTML` : langage des pages Web
- découvrir les feuilles de style : le `CSS`

1.1 – Découvrir le langage HTML

Comme tu l'as vu en seconde, le *Web* est un ensemble de documents reliés entre eux par des *liens hypertexte*. Ces documents peuvent donc être parcourus de façon non linéaire en **actionnant** (clic, touch) les liens hypertextes.

Le terme Web englobe aussi toute l'infrastructure hébergeant cette collection mondiale de documents :

- les langages informatiques permettant d'écrire les documents hypertextes : **HTML** et **CSS** ;
- une architecture client-serveur utilisant le protocole `HTTP` (*Hypertext Transfer Protocol*) ou sa version sécurisée `HTTPS` (*HTTP secure*) ;
- des langages de programmation côté serveur dont le but est de générer du `HTML` ;
- des langages de programmation côté client dont le but est de permettre l'écriture d'application Web hautement interactives (par exemple des jeux).

REMARQUE

Le chapitre a pour but de présenter les langages `HTML` et `CSS`. Dans cette première partie, tu vas commencer par le **langage HTML**. Ces langages sont présentés en dehors de toute considération de réseau, de client ou de serveur.

Premier exemple

Le format **HTML** (*Hyper Text Markup Language*) est un format textuel permettant de décrire :

- le contenu
- la structure

d'un document ou d'une page Web.



ACTIVITÉ

Tu vas maintenant créer la première page HTML de ce cours.

1. **Ouvrir** un éditeur de texte et **copier** le code ci-dessous.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8"/>
  <title>Le titre de ma première page HTML</title>
</head>
<body>
  <h1>Première page </h1>
  <p>Notre toute première page !</p>

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

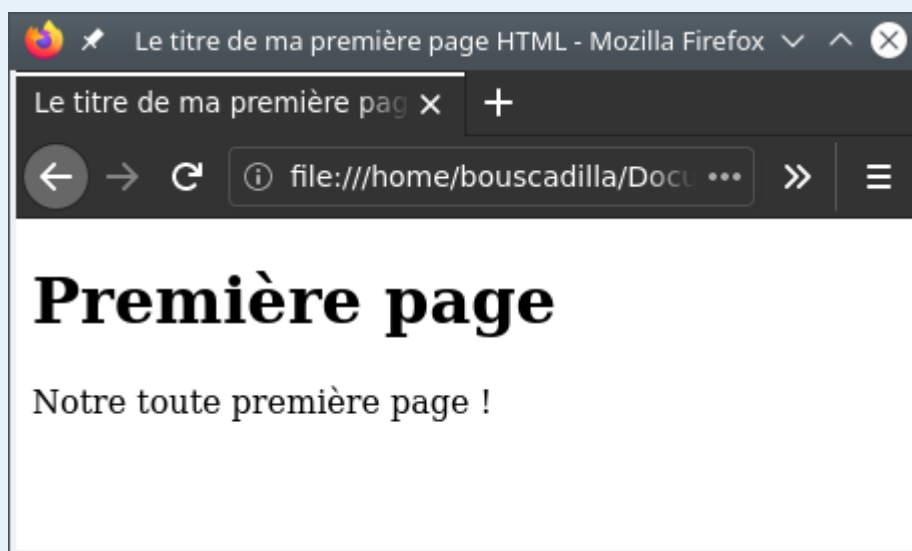
</body>
</html>
```

2. **Enregistrer** le fichier sous `premier_exemple.html`.

3. Enfin, **ouvrir** ce fichier avec *Firefox*.

CORRECTION

Le code précédent devrait afficher quelques chose qui ressemble à cela :



Balises et attributs

Le code source possède une structure donnée par les **balises HTML** qui sont des suites de caractères délimitées par < et >.

Par exemple :

- <head> est une *balise ouvrante*
- </h1> est une *balise fermante* (à cause de la barre oblique /).

Un **élément** est constitué d'une paire de balise ouvrante et fermante ainsi que du contenu situé entre les deux. Par exemple :

- title est l'élément constitué par `<title>Le titre de ma première page HTML</title>`.

La balise `<meta charset="utf-8" />` est une **balise vide** car elle est à la fois ouvrante et fermante.

Certaines balises et leurs contenus correspondent directement à des éléments graphiques. Par exemple `<h1>Première page</h1>` se retrouve comme un titre de la page en gras.



ACTIVITÉ

Reprendre le code de l'activité précédente, **lister** toutes les balises HTML que tu identifies et pour chacune **indiquer** si c'est une balise fermante, ouvrante ou vide.

CORRECTION

Balise ouvrante	Balise fermante	Balise Vide
<code><head></code>	<code></head></code>	
<code><html></code>	<code></html></code>	
<code><title></code>	<code></title></code>	
		<code><meta /></code>
<code><body></code>	<code></body></code>	
<code><h1></code>	<code></h1></code>	
<code><p></code>	<code></p></code>	

Dans une balise, on peut utiliser des **attributs**. Les attributs sont associés à des valeurs.

Exemple

Par exemple, pour indiquer que la page HTML est écrite en français, on va utiliser l'attribut `lang` dans la balise ouvrante `<html lang="fr">`. En écrivant `lang="fr"`, on stipule que l'attribut `lang` a pour valeur "fr". Si on avait utilisé la valeur "en", alors la langue officielle du document aurait été l'anglais.



ACTIVITÉ

1. Dans le code source de l'exercice précédent, **trouver** un autre attribut et **indiquer** la balise à laquelle il est associé.
2. **Trouver** 6 valeurs possibles pour l'attribut `lang` correspondants à 6 langues parlées dans l'union européenne (autres que le français et l'anglais).

CORRECTION

1. Le code source utilise dans l'élément `head` l'attribut `charset`. Cet attribut est défini à "utf-8".
2. Voici quelques exemples de langues possibles : espagnole `es`, allemande `de`, italienne `it`, polonaise `pl`, portugaise `pt` ou encore lithuanienne `LT`.

Validation du code

Le format HTML est standardisé par le W3C. Les règles décrivant les documents valides sont complexes (plusieurs centaines de pages!).

Voici quelques principes généraux.

Structure du document

La structure doit être similaire à celle de l'exercice :

1. Le document :

- commence par une déclaration de type `<!DOCTYPE html>`.
- contient ensuite un unique élément `html`.

2. L'élément `html` :

- contient l'attribut `lang` (information utilisée par les navigateurs Web, par exemple pour proposer un correcteur orthographique adapté dans les zones de textes).
- contient uniquement deux sous-éléments : `head` et `body`.

3. L'élément `head` :

- contient les entêtes du document dont au moins l'élément `title` (texte affiché dans l'onglet ou la barre de fenêtre de la page).
- La balise vide `<meta charset="utf-8">` indique l'encodage de caractère (format de texte) utilisé pour le document. Sauf exception, il vaudra toujours `utf-8`.

4. L'élément `body` :

- regroupe tous les éléments qui structurent la page.

5. Les balises :

- Leur nom est écrit en minuscule
- Les balises ouvrantes sont **bien parenthésées** : c'est-à-dire fermées par une balise fermante de même niveau d'imbrication. Par exemple le code ` <i> </ b> </ i>` est invalide



ACTIVITÉ

En effectuant une recherche sur internet, **retrouver** le nom des 14 balises vides possible.

CORRECTION

Voici la liste des 14 balises vides acceptées :

```
<area>, <base>, <br>, <col>, <embed>, <hr>, <img>, <input>, <link>,
<meta>, <param>, <source>, <track>, <wbr>.
```

Balises de structures

Sections et paragraphes



ACTIVITÉ

Indiquer le nom des balises qui permettent de définir les titres de section, sous-sections, etc.

CORRECTION

Pour structurer le texte en chapitres, sections, parties, etc., il faut utiliser les balises h... :

```
<h1> Titre de premier niveau </h1>
<h2> Titre de deuxième niveau </h2>
<h3> Titre de troisième niveau </h3>
```

```
<h4> Titre de quatrième niveau </h4>
<h5> Titre de cinquième niveau </h5>
<h6> Titre de sixième niveau </h6>
```



ACTIVITÉ

Indiquer le nom de l'élément qui permet de créer un nouveau paragraphe.

CORRECTION

Pour créer un nouveau paragraphe (avec saut à la ligne), il faut utiliser les balises `<p> ... </p>`.

Voici un exemple d'utilisation de cet élément :

```
<p> Un premier paragraphe
    on y met
    du texte    n'importe comment
    et il est affiché comme
    il
    faut </p>
<p> Et si on comment un nouveau paragraphe
    on remarque enfin un retour à la ligne! <p>
```


Listes et tableaux



ACTIVITÉ

Donner un exemple de liste énumérée et un exemple de liste non-énumérée.

CORRECTION

Voici un exemple de liste énumérée :

```
<ol>
  <li> Se lever </li>
  <li> Déjeuner </li>
  <ol>
    <li> prendre un thé </li>
    <li> manger quelque chose </li>
  </ol>
</ol>
```

ce qui affiche :

1. Se lever
2. Déjeuner
 1. prendre un thé
 2. manger quelque chose

Voici un exemple de liste non énumérée :

```
<ul>
  <li> Une première chose à faire </li>
  <li> Une seconde chose à faire </li>
```

```
<ul>
  <li> d'abord A </li>
  <li> ensuite B </li>
</ul>
</ul>
```

ce qui donne :

- Une première chose à faire
- Une seconde chose à faire
 - d'abord A
 - ensuite B



ACTIVITÉ

Donner le nom des balises :

1. qui délimite un tableau
2. qui commence une nouvelle ligne du tableau
3. qui encadre une colonne.

Donner une exemple simple de tableau.

CORRECTION

Les balises à utiliser pour les tableaux sont :

1. `table` pour délimiter un tableau
2. `tr` pour les lignes

3. td pour les colonnes

Voici un exemple de tableau :

```
<table>
  <tr>
    <td> Janvier </td> <td> Février </td> <td> Mars</td>
  </tr>
  <tr>
    <td> 31 </td> <td> 28 </td> <td> 31 </td>
  </tr>
</table>
```

qui donne alors :

Janvier	Février	Mars
31	28	31

Balises de texte

À la différence des balises de structures, les **balises de textes** ne produisent pas de nouveaux bloc de texte formaté. Le flot du texte n'est pas interrompu.

Ces balises servent à indiquer la nature du texte.



ACTIVITÉ

Identifier pour chaque balise son rôle :

–

- `<i>`
- `<u>`
- `<mark>`
- `<small>`
- `<code>`
- `<sub>`
- `<sup>`

Écrire un petit texte utilisant ces balises.

CORRECTION

Voici le rôle de ces balises :

- `` : mise en gras
- `<i>` : mise en italique
- `<u>` : souligner
- `<mark>` : surligner
- `<small>` : texte en petit caractères
- `<code>` : police fixe utilisée souvent pour du code informatique
- `<sub>` : texte en indice
- `<sup>` : texte en exposant

Mettre en forme un texte est très simple. Il suffit d'utiliser quelques balises. On peut écrire de la chimie CO_2 ou de la physique $6,02 \times 10^{-23}$ sans faire d'erreur.

Liens et ressources

La fonctionnalité **essentielle** des documents HTML est la possibilité d'utiliser des **liens hypertextes**. Ce sont les textes cliquables qui apparaissent dans un style souligné par défaut et qui permettent d'accéder à d'autres ressources (pages, média, etc.).

Pour insérer un lien, il faut :

- utiliser la *balise* `<a>`
- définir son *attribut* `href` avec pour valeur l'emplacement de la ressource que l'on souhaite pointer entre guillemets
- définir le *contenu de la balise* qui sert à afficher le lien

Exemple

Voici un lien hypertexte vers le fichier `expose_gp1.html` :

```
<a href="expose_gp1.html">Accéder à <b>notre</b> exposé</a>
```

qui affiche cela :

[Accéder à **notre** exposé](#)

L'attribut `href` pointe vers le fichier qui est dans le même dossier que le document visité.

Il est possible d'insérer directement dans la page des ressources externes.

Pour insérer une image :

- balise ``
- attribut `src` : chemin d'accès à l'image
- attribut `alt` : légende de l'image
- attribut `width` : largeur de l'image

- attribut `height` : hauteur de l'image

Exemple

Pour insérer l'**image** `logo_html.svg` qui est dans le sous-répertoire `res` et lui imposer une largeur de 240px, il faut écrire :

```

```

ce qui donne :



Balises neutres, classes et identifiants

Pour la définition du style graphique, il est nécessaire d'utiliser les balises neutres, les classes et les identifiants. Ces derniers permettent d'étiqueter les éléments de la page et ainsi de les personnaliser.

Il existe deux **balises neutres** qui n'ont pas d'effet particulier :

- la balise de structure `<div>` dont le comportement par défaut est identique à `<p>`

- la balise de texte `` laisse le texte inchangé.

Chaque balise d'une page Web peut posséder un **identifiant** unique grâce à l'attribut `id`.

Enfin, l'attribut `class` contient une suite de noms de classes séparés par des espaces. Une classe est un ensemble de balises qui possèdera certaines caractéristiques graphiques communes.

Exemple

Voici un extrait de page contenant un titre de section et une liste non numérotée.

La balise de titre contient l'identifiant unique "titre1" et les items de la liste appartiennent à la classe "pair" ou "impair" en fonction de leurs positions.

```
<h4 id="titre4">Voici la todo list</h4>
<ul>
  <li class="impair">Acheter du pain</li>
  <li class="pair">Arroser les plantes</li>
  <li class="impair">Apprendre le HTML</li>
  <li class="pair">Dormir</li>
</ul>
```