

HTML et CSS

HTML - Travaux dirigés

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 Tranfert 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).

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>

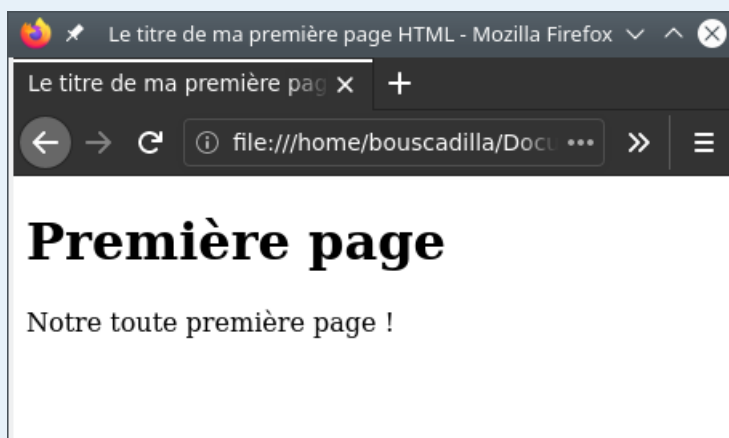
  <!-- Ceci est un gros commentaire -->

</body>
</html>
```

2. **Enregistrer** le fichier sous `premier_exemple.html`.
3. Enfin, **ouvrir** ce fichier avec *Firefox*.

MÉTHODE

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



Capture d'écran du premier exemple

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

À partir de l'exercice précédent, **lister** toutes les balises HTML que tu identifies et pour chacune **indiquer** si c'est une balise fermante, ouvrante ou vide.

MÉTHODE

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.

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).

MÉTHODE

1. Le code source utilise dans l'élément `head` l'attribut `charset`. Cet attribut est défini à "utf-8".
2. "es", "gem", "it", "pl", "pt", "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 :
 1. commence par une déclaration de type `<!DOCTYPE html>`.
 2. contient ensuite un unique élément `html`.
2. L'élément `html` :

1. contient l'attribut `lang` (information utilisée par les navigateurs Web, par exemple pour proposer un correcteur orthographique adapté dans les zones de textes).
 2. contient uniquement deux sous-éléments : `head` et `body`.
3. L'élément `head` :
1. contient les entêtes du document dont au moins l'élément `title` (texte affiché dans l'onglet ou la barre de fenêtre de la page).
 2. 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` :
1. regroupe tous les éléments qui structurent la page.
5. Les balises :
1. Leur nom est écrit en minuscule
 2. Les balises ouvrantes sont **bien parenthésées** : c'est-à-dire fermées par une balise fermante de même niveau d'imbrication. Par exemple :
 - ` <i> </ b> </ i>` est invalide



ACTIVITÉ

Retrouver le nom des 14 balises vides possible.

MÉTHODE

- `<area>`
- `<base>`
- `
`
- `<col>`
- `<embed>`
- `<hr>`
- ``
- `<input>`
- `<link>`
- `<meta>`
- `<param>`
- `<source>`
- `<track>`
- `<wbr>`



ACTIVITÉ

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

MÉTHODE

```
<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.

```
<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>
```



ACTIVITÉ

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

```
In [2]: for i in range(10):
        print(i, "Hello, world!")

0 Hello, world!
1 Hello, world!
2 Hello, world!
3 Hello, world!
4 Hello, world!
5 Hello, world!
6 Hello, world!
7 Hello, world!
8 Hello, world!
9 Hello, world!
```