

Chapitre: le langage HTML

1. Historique:

- 1970 - 1980 : développement des réseaux informatiques
=> USA: ARPANET Les ordinateurs apprennent à communiquer entre eux en utilisant des langages (des protocoles) tels que TCP/IP.
- 1990 : C'est la naissance du protocole HTTP qui permet à des machines d'échanger du texte "amélioré" : **l'hypertext**.
=> Tim Berners LEE (*anglais*) et Robert CAILLIAU (*belge*) sont à l'origine du projet.
=> texte amélioré => texte avec lequel l'utilisateur peut interagir

Editer un fichier html

Pour écrire un fichier de texte amélioré, il suffit de lui coller l'extension **.html**

Exemple: **premier.html**

2. Le langage HTML

=> [Site des Développeurs de Mozilla](#)

HTML : HyperText Markup Language

Il s'agit d'un langage déclaratif qui indique au navigateur (Firefox, Chrome, ...) ce que l'utilisateur doit voir.

Pour cela, on utilise un éditeur de texte/code (Visual Studio Code, SublimText, Vim...). On (le développeur) se sert des balises pour indiquer ce que l'on veut.

Il existe deux grands types de balises:

- les balises paires/couplées

```
<body></body>
<p></p>
<h1></h1>
```

- les balises orphelines

```
<img>
<br>
<meta>
```

Remarques:

- La structure minimale d'un fichier html devrait ressembler à:

```
<html>
  <head>
    <!--Ici la partie pour les machines-->
  </head>
  <body>
    <!--Ici la partie visibles des utilisateurs-->
  </body>
</html>
```

- Une balise `<Y></Y>` ouverte après une balise `<X></X>` doit être fermée avant.

```
<!-- Correct -->
<x> <y> </y> </x>
<!-- Correct -->
<x> </x> <y> </y>
<!-- Incorrect -->
<x> <y> </x> </y>
```

- Un utilisateur peut accéder au code HTML d'une page depuis son navigateur avec la combinaison `ctrl+U`. Il est possible d'inspecter (clic droit) un élément d'une page.
- Quelques balises simples à connaître:
 - `<p></p>` pour un paragraphe avec saut de ligne
 - `` pour une liste non ordonnée
 - `` pour une liste ordonnée
 - `` pour les items d'une liste
 - `<a>` pour créer des liens

Remarque: Très souvent les balises ouvrantes (ou orphelines) possède un ou plusieurs **attributs**. Certains attributs sont spécifiques à un type de balise, d'autres sont généraux.

Ex:

- spécifiques: `href` pour `<a>`
`=>`
- généraux: `id` et `class`
`<p id="untruc" class="unAutreTruc"></p>`

Pour des raisons pratiques, on classe souvent les fichiers dans des dossiers différents. Le dossier courant s'appelle `./` mais il n'est pas nécessaire de toujours l'indiquer. La recherche se fait toujours par défaut dans `./`

Remarque:

Bien que conseillé, le symbole `./` est facultatif. On peut alors atteindre tous les fichiers ou les dossiers qui sont plus bas dans l'arborescence.

Pour remonter dans l'arborescence et atteindre le dossier du niveau juste supérieur, on écrit `../` Pour remonter de deux niveaux, c'est `../../` etc.

Un chemin vers un fichier commençant par `./` ou `../` est appelé **chemin relatif**.

3. Un fichier `index.html`

En général, la page d'accueil d'un site se nomme `index.html`

=> voir fichier correspondant

- **Avec des images:**

C'est la balise orpheline `` qui permet d'insérer une image. Elle s'accompagne obligatoirement de l'attribut `src` avec l'adresse vers le fichier image. Parfois, on ajoute l'attribut `alt` en cas d'échec de chargement de l'image.

Ex: ``

Le chemin relatif `../img/planete.jpeg` signifie que pour atteindre le fichier `planete.jpeg` depuis le dossier courant :

- il faut remonter un rang plus haut dans l'arborescence `../`
- Descendre dans le dossier `img/`

- **Avec des tableaux.....sans bordures**

Pour insérer un tableau, on utilise la balise `<table></table>` puis on définit une ligne `<tr></tr>` puis on définit une case `<td></td>`.

On peut également définir des cases d'en-têtes avec `<th></th>`.

Pour donner plus de style (bordure, couleur, position,...) à une page, il faut utiliser le langage de style `'CSS'`.

- **Avec des titres de niveaux 1 à 6**

On utilise la balise `<h></h>` où `n` est compris entre 1 et 6. Par défaut, le navigateur donne une taille de police supérieur aux titres de niveaux 1

- **Avec des vidéos :**

- Si la vidéo est hébergée sur notre machine, on utilisera le tag `<video></video>`

Ex:

```
<video width="320" height="240" controls>
<source src="../img/movie.mp4" type="video/mp4">
</video>
```

- Si la vidéo est hébergée sur un serveur externe, on utilisera le tag `<iframe></iframe>` avec le chemin `embed/codeDeLaVideo`

Ex:

```
<iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>
</video>
```

- des pieds de pages, des menus, etc....

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My First HTML Page</title>
</head>
<body>
  <h1>Welcome to My Page!</h1>

  <h2>About Me</h2>
  <p>Hello! I'm learning HTML and this is my first webpage. I'm excited
to learn about all the different HTML elements and how to use them to
create interesting web content.</p>

  <p>This page demonstrates various HTML elements like lists, tables,
links, and more. I hope this helps other beginners understand how HTML
works!</p>

  <h2>Check Out My GitHub</h2>
  <a href="https://github.com" target="_blank">Visit My GitHub
Profile</a>

  <h2>My Favorite Plates</h2>
  <ol>
    <li>Spaghetti Carbonara</li>
    <li>Grilled Salmon</li>
    <li>Chocolate Cake</li>
  </ol>

  <h2>Daily Meals</h2>
  <ul>
    <li>Breakfast</li>
    <li>Lunch</li>
    <li>Dinner</li>
  </ul>

  <h2>Class Roster</h2>
  <table>
    <tr>
      <th>Firstname</th>
      <th>Name</th>
```

```
        <th>Age</th>
    </tr>
    <tr>
        <td>John</td>
        <td>Smith</td>
        <td>25</td>
    </tr>
    <tr>
        <td>Sarah</td>
        <td>Johnson</td>
        <td>22</td>
    </tr>
</table>

<h2>My Favorite Music</h2>
<iframe width="560" height="315"
src="https://www.youtube.com//embed/K2snTkaD64U" allow="accelerometer;
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>

<h2>Random Image</h2>


</body>
</html>
```