

Apprendre le HTML

Plan du cours

- Introduction
- Créons notre premier fichier
- La syntaxe
- La structure d'une page HTML
- Balises textuelles
- Le lien hypertexte

- Les images
 - Les tableaux
 - Les listes
 - Les éléments block et inline
 - Les formulaires
 - Les bonnes pratiques
-

HTML est un langage de balisage qui permet de définir la structure d'une page Web. Il constitue le langage de bases d'un site internet.

Créons notre premier fichier

Nous allons commencer par créer un fichier avec comme extension .html.

```
mkdir /var/www/projet_html  
vim /var/www/projet_html/accueil.html
```

Nous avons besoin d'un éditeur de texte pour écrire nos premières lignes de code en HTML. Nous pouvons utiliser les éditeurs par défaut comme nano ou vi.

Mais il est préférable d'utiliser des éditeurs plus avancés tels que Visual Code.

Nous allons donc télécharger Visual Code a cette adresse.

<https://code.visualstudio.com/>

Une fois le téléchargement terminé, nous allons l'installer avec cette commande:

```
sudo dpkg -i  
~/Téléchargements/code_1.39.1_amd64-.deb
```

Nous pouvons maintenant ouvrir le fichier HTML en faisant:

```
code /var/www/projet_html/accueil.html
```

À noter: qu'il est aussi possible d'utiliser d'autre éditeur de text tel que Sublime Text ou Atom

La syntaxe

HTML est un langage de balisage. Il utilise les balises pour structurer des pages.

Les balises sont des instructions entourées par les signes “<” et “>”.

Exemple: <p>,

Certaines balises disposent d’une balise fermante qui ajoute un “/” pour spécifier les limites de celle-ci.

Exemple: <p>le contenu de la balise</p>

En plus des balises, on trouve souvent dans un fichier html des commentaires.

Les commentaires sont des instructions qui n’auront aucun effet sur le document.

Ils servent à ajouter des instructions ou des explications sur certaines parties du document.

Le concept de commentaire est présent dans toutes les langages de programmation.

Il s’agit d’une bonne pratique qui permet d’expliquer une portion de code à ses collaborateurs.

Pour écrire un commentaire en HTML on utilise la syntaxe suivante

<!-- le contenu de notre commentaire -->.

Tout le texte entre <!-- et --> sera considéré comme un commentaire et donc, ne s’affichera pas sur la page Web.

Les commentaires peuvent aussi être utilisés pour désactiver une partie du code.

Structure d'une page HTML

Voici ci-dessous la structure minimum d'une page HTML.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>Document</title>
  </head>
  <body>
    </body>
</html>
```

La balise `<!DOCTYPE html>` permet de préciser le type de document.

Les balises `<html>` entourent tout document HTML. Ils permettent de définir la langue du site avec l'attribut **lang="fr"**.

Les balises `<head>` contiennent les informations relatives aux sites. Ils peuvent contenir plusieurs balises `<meta>`. Il existe plusieurs types de balise meta. La balise meta charset permet de préciser l'encodage du site.

Toujours dans les balises `<head>`, nous avons la balise `<title>` qui va permettre de définir le texte à afficher dans l'onglet du site.

Et enfin nous avons les balises `<body>` qui vont afficher le contenu du site. C'est entre ces balises que nous allons mettre les balises et contenus qui vont constituer notre page Web

Balises textuelles

Pour formater du texte nous utiliserons différentes balises en fonction des besoins.

La balise `<p>` permet de créer un paragraphe dans notre page. Elle peut prendre autant de texte nécessaire. Et tout contenu à la suite de cette balise sera affiché sur une nouvelle ligne.

La balise `
` permet d'insérer un saut de ligne. Elle fait partie de ces balises qui n'ont pas de balise fermante.

La balise `<hr>` va afficher une ligne de séparation.

Les balises `` et `` permettent de mettre un texte en gras.

La balise `<u>` va souligner le texte qu'elle contient.

La balise `<i>` permet de formater un texte en italique.

Pour mettre du texte en valeur nous avons une liste de balises commençant par la lettre 'h' suivi des chiffres allant de 1 à 6. Ce sont les balises d'en-tête

Exemple: `<h1>Titre du site</h1>`

H1 a un effet large et gras sur son contenu. H2 est moins large. Ainsi de suite jusqu'à la balise `<h6>` qui a un effet similaire de la balise ``.

La balise `<sup>` va afficher son contenu en exposant

La balise `<sub>` positionne le texte en indice

Les liens hypertextes

Les liens hypertexte permettent de naviguer d'une page à une autre, d'un site Web à un autre ou à différents endroits au sein d'une page Web.

La syntaxe est la suivante:

```
<a href="adresse_url" titre="une description du lien">Le nom du  
lien</a>
```

La balise utilisée est `<a>`; elle a comme attribut href et title.

- ❖ L'attribut href va contenir l'adresse ou l'URL cible.

Pour une URL cet attribut sera comme suite :

```
href="http://google.com"
```

Pour un lien interne (adresse vers un fichier sur la même machine) on aura: `href="/lien/vers/lefichier.html"`. Cette adresse peut aussi être absolue.

Et pour cibler un endroit sur la même page nous allons utiliser une ancre. Pour définir une ancre il faut d'abord préciser l'identifiant de l'élément cible: `<balise id="id_ancre">...</balise>`

Et pour naviguer vers cette ancre on ajoute un # à l'identifiant cible:

```
<a href="#id_ancre">lien</a>
```

- ❖ L'attribut **title=""** permet d'ajouter une description au lien.

Les images

Pour insérer une image dans notre page nous allons utiliser la balise `img`

```

```

Cette balise utilise deux paramètres: `src` et `alt`.

L'attribut `src` définit le chemin vers l'image. Ce chemin peut être absolue ou relatif. Il peut aussi être une URL.

L'attribut `alt` permet d'ajouter une description de l'image qui sera affichée dans les cas où le chemin vers l'image n'est pas valide.

La balise `img` peut prendre deux autres attributs qui sont `width` et `height`
`Width` définit la largeur de l'image et `height` va définir sa hauteur.

Les listes

Les listes permettent, comme leur nom le suggère, d’afficher une succession d’éléments plus ou moins liés.

Elles peuvent être ordonnées ou non ordonnées.

Pour créer une liste non ordonnée nous utilisons la balise ``. Celle-ci contiendra des balises `` qui seront les différents composants de la liste.

```
<ul>  
  <li>element liste</li>  
  <li>element liste</li>  
  <li>element liste</li>  
  <li>element liste</li>  
</ul>
```

Pour créer une liste ordonnée le procédé est le même qu’avec les listes non ordonnées; excepté que nous utiliserons la balise `` à la place de ``.

Par défaut, la balise `` va ordonner les éléments de la liste en utilisant des entiers.

1. element un
2. element deux
3. element trois
4. ...

Nous pouvons cependant changer ce comportement avec l’attribut “type” de la balise. Cet attribut va permettre d’utiliser des lettres ou des chiffres romains à la place des entiers.

```
<ol type="I">  
  <li>Element un</li>  
  <li>Element deux</li>  
</ol>
```

Les options de l’attribut “type” sont: **I,A,a,i**

Les tableaux

Les tableaux en html sont un autre moyen d'organiser du contenu. Ils permettent une organisation en ligne et en cellules de différents éléments.

La balise pour créer un tableau est `<table>`. Cette balise va par la suite contenir d'autres les balises qui vont définir les lignes et les cellules du tableau.

La balise `<tr>` crée une ligne du tableau. Elle ne doit contenir que des balises `<td>`. Les balises `<td>` sont chargées de créer les cellules.

```
<table>
  <tr><!-- premiere ligne -->
    <td>table data</td>
    <td>table data</td>
    <td>table data</td>
  </tr>
  <tr><!-- second ligne -->
    <td>table data</td>
    <td>table data</td>
    <td>table data</td>
  </tr>
</table>
```

La balise `<th>` est comme la balise `<td>`, elle définit les cellules du tableau. La différence avec `<td>` est qu'elle va mettre en valeur son contenu (texte en gras). Elle est souvent utilisée sur la première ligne d'un tableau pour définir les entêtes de celle-ci.

Par défaut une cellule n'occupe qu'une ligne et une colonne. Cependant les balises `<td>` et `<tr>` peuvent prendre deux attributs qui vont permettre de modifier ce comportement par défaut.

- ❖ `colspan`: cet attribut va définir le nombre de colonnes qu'une cellule va occuper
- ❖ `rowspan`: cet attribut définit le nombre de lignes qu'une cellule peut occuper.

```
...
<td colspan="3" rowspan="2">
...
```

Les balises blocks Vs inline

Chaque balise en HTML à un affichage par défaut. Cette affichage est soit en block ou en inline.

- ❖ Un élément en block va toujours commencer sur une nouvelle ligne.

Elle va ensuite prendre toute la largeur de la ligne. Les balises p, h1,h2,h3,h4,h5,h6, ul, li, table ou encore hr sont des éléments en block.

- ❖ Les éléments inline ne vont prendre que l'espace nécessaire pour afficher leur contenu. Ils ne débutent une nouvelle ligne que si c'est nécessaire.

Les balises img, a, b, ou strong sont des éléments inline.

Les formulaires

Les formulaires en HTML vont permettre d'afficher une interface qui va proposer aux utilisateurs d'entrer des données.

Pour créer un formulaire nous utiliserons la balise `<form>`. Elle prend les attributs méthode et action. L'attribut méthode définit la méthode d'envoi du formulaire. Sa valeur est soit GET ou POST. Et l'attribut action va préciser le lien vers le fichier qui est chargé de traiter le formulaire.

NB: Il faut cependant préciser que HTML n'est pas en mesure de traiter les données d'un formulaire. Nous utiliserons d'autres langages de programmation pour cela.

Dans la balise `<form>`, nous pouvons utiliser plusieurs autres balises qui vont créer des champs pour recueillir des informations.

- ❖ La balise `<label>` permet d'ajouter un titre devant un champ.
- ❖ La balise `<input>` est la balise la plus utilisée pour définir les éléments d'un formulaire. Elle change d'état en fonction de la valeur de son attribut type.
- ❖ `<input type="text">` va afficher un champ de texte à une seule ligne.
- ❖ `<input type="email">` va retourner le même type de champ à l'exception qui va afficher une erreur si son contenu n'a pas le format d'une adresse email.
- ❖ `<input type="number">` ce type champ ne va accepter que des chiffres. Il prend aussi deux attributs, max et min qui vont définir la valeur maximale et la valeur minimale à accepter.
- ❖ `<input type="checkbox">` va afficher une case à cocher
- ❖ `<input type="radio">` va afficher un bouton d'option. Elle n'est jamais utilisée seule puisqu'elle propose de faire un choix sur une liste d'options. Elle utilise l'attribut name pour réunir les différentes options.

Les formulaires

`<input type="file">` permet de choisir un fichier à uploader. L'utilisation de ce type de champ nécessite que l'on ajoute l'attribut `enctype` à la balise `<form>`. Cet attribut sera défini comme suite: `enctype="multipart/form-data"`.

L'absence de cet attribut fera que l'utilisateur pourra charger le fichier mais, ce dernier, ne sera jamais envoyé au fichier de traitement.

En plus des différentes balises `input`, nous avons la balise `<textarea>`. Elle va créer une zone de texte multi ligne. Elle prend en paramètre les attributs `cols` et `rows` qui vont définir respectivement sa largeur et sa hauteur.

Un attribut qui est souvent utilisé dans les champs de texte est l'attribut `placeholder`. Il permet d'afficher du texte pour aider à comprendre quelle est la valeur souhaitée.

`<input type="text" placeholder="Entrez votre nom de famille">`

Pour soumettre le formulaire il faudra créer un bouton. Il existe deux méthodes pour créer un bouton:

`<input type="submit">` la valeur `submit` va créer un bouton pour soumettre le formulaire. À la place de `submit` nous pouvons utiliser `reset` pour réinitialiser tous les champs du formulaire.

`<button type="submit"></button>` cette balise se comporte comme la balise `input`. À la différence qu'elle prend une balise fermante. Elle peut aussi être utilisée pour réinitialiser le formulaire avec la valeur `reset`.

Les balises div et span

Les balises div et span n'ont aucun effet spécifique sur leur contenu.

Elles ont le rôle de conteneur. Elles n'ont aucun attribut spécifique.

En fait leur utilisation prend du sens avec d'autres langages tels que le CSS ou le JavaScript.

- ❖ `<div>` est un élément block. On l'utilise donc pour regrouper des éléments devant occuper toute la largeur de la page.
- ❖ `` par contre, est un élément inline. Elle ne prend que l'espace nécessaire à son contenu.

Quelques bonnes pratiques

L'activité nommée SEO (Search Engine Optimization) a pour objectif d'améliorer le trafic sur une page Web.

Les bases de cette activité reposent grandement sur une utilisation optimale des balises HTML.

Ainsi, pour une bonne SEO, nous penserons à suivre les règles suivantes:

- ❖ Renseigner la balise title. Le contenu de la balise title sera le texte qu'affichent les moteurs de recherche dans leurs résultats. Cette balise doit donc être attractive. Il est conseillé de ne pas dépasser 65 caractères (10 0 12 mots) et d'utiliser des mots clés.
- ❖ Ajouter une balise meta description. C'est une balise que nous n'avons pas vue. Elle est le résumé qui s'affiche en bas du titre dans les résultats de la recherche. Comme la balise title, elle doit inciter l'utilisateur à cliquer sur le lien.
- ❖ Utiliser les balises d'en-têtes (h1 à h6). Ces balises permettent aux moteurs de reconnaître les titres et sous-titres de la page.
- ❖ Renseigner l'attribut alt dans les balises ``. Cet attribut permet au moteur de recherche d'identifier l'image. Cependant, sa valeur n'est pas une longue rédaction; mais juste deux à trois mots pour décrire l'image.

- ❖ Utiliser des balises sémantiques. HTML5 a introduit de nouvelles balises qui servent exclusivement à indiquer le type de leur contenu. Nous pouvons utiliser la balise `<div>` pour arranger les différents segments de notre site. Mais il est mieux de faire usage des balises `<header>`, `<nav>`, `<main>`, `<section>` et `<footer>`. Ces balises ont un comportement identique à la balise `<div>`.
 - `<header>` est utilisé pour contenir l'entête du site (à ne pas confondre avec la balise `<head>`).
 - `<footer>` contiendra, naturellement le pied de page.
 - `<main>` contiendra alors le contenu du site.

Ces trois balises doivent être unique dans le document HTML.

- La balise `<nav>` sera utilisée pour contenir le menu du site Web.
- Et la balise `<section>` est utilisée pour les différentes sections du site.

Conclusion

Nous n'avons pas fait le tour de toutes les balises HTML. Pour tester toutes les balises, je vous invite à visiter le lien suivant:

<https://www.codeshttp.com/toutes-les-balises-html-liste-complete-avec-syntaxe.html>

Cependant il faut préciser que le HTML ne suffit pas pour concevoir un site. Les connaissances partagées sur ce document ne suffisent pas pour gérer le design de notre page Web. Pour cela il nous faudra apprendre un autre langage, le CSS; qui est la suite logique dans notre apprentissage de la programmation Web.

Liens utiles

- ❖ <https://www.codeshttp.com/toutes-les-balises-html-liste-complete-avec-syntaxe.html>
- ❖ <https://developer.mozilla.org/fr/docs/Web/HTML>
- ❖ https://www.w3schools.com/TAGs/tag_doctype.asp
- ❖ <https://www.lafabriquedunet.fr/seo/articles/balises-html-seo/>