

HTML

" Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.

Ce langage permet :

- d'écrire de l'hypertexte
- de structurer sémantiquement la page
- de mettre en forme le contenu
- de créer des formulaires de saisie
- d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques
- de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). " ([Wikipedia](#))

Dans le développement web "moderne", le HTML est principalement utilisé pour structurer la page, on peut se le représenter comme l'ossature notre site web. La mise en page et l'interactivité seront prise en charge respectivement par le CSS et le Javascript.

Le HTML n'est pas à proprement parlé un langage de programmation qu'on doit compiler comme le c# par exemple. C'est plutôt un ensemble de balise (mot clé) qui sera interprété par le navigateur qui va afficher la page. Il peut donc y avoir des différences d'affichage selon le navigateur utilisé malgré qu'on tends de plus en plus vers une uniformité. C'est cependant une réalité avec laquelle le développeur web doit composer.

Balise HTML

Le langage HTML utilise des balises pour indiquer au navigateur de quelle façon afficher le contenu de la page. Une balise débute toujours par le caractère < suivi du nom de la balise. On peut ajouter ensuite des attributs à chaque balise, certain général et d'autre plus spécifique.

On va voir plus loin les attributs mais voici un exemple de syntaxe d'une balise avec des attributs :

```
<input type="text" id="nom" name="nom" value="Mathieu">
```

La balise se terminera par le caractère >. Tout le contenu qui suivra sera affecté par cette balise. Pour chaque ouverture de balise, on doit ajouter une autre balise de fermeture. Pour ce faire on utilise le même mot clé mais précédé du caractère /. Ex:

```
<title>Le titre de ma page</title>
```

Dans cet exemple le texte 'Le titre de la page' sera affiché comme un titre parce qu'il est englobé dans la balise **title**.

Certaines balises n'ont pas besoin de fermeture, comme par exemple la balise **img** pour afficher une image :

```

```

Structure HTML5 d'une page web

Le code qui suit représente la structure de base d'une page web

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Le titre de ma page</title>
  </head>
  <body>
    <p>Un paragraphe qui sera affiché à l'utilisateur</p>
  </body>
</html>
```

Voyons plus en détail chacune de ces balises

< !DOCTYPE html >

Historiquement, cette balise servait à indiquer au navigateur quelles règles il devait utiliser pour bien afficher la page. On continue de l'inclure pour des questions de rétrocompatibilité. Cette balise ne doit pas être "fermée".

< html >

La page dans son ensemble. Toutes les autres balises html doivent se retrouver à l'intérieur de celle-ci.

< head >

On va ajouter ici tout ce qu'on inclure dans la page web mais qui ne sera pas visible par l'utilisateur.

- Des mots-clé pour le référencement
- Des liens vers les fichiers css, javascript et autres librairies utilisées.

< meta charset="utf-8">

Pour spécifier le jeu de caractère à utiliser pour bien afficher le texte de la page. Tout comme la balise |Doctype, on l'inclus par défaut sans se poser d'autres questions. Cette balise ne doit pas être "fermée".

< title >

Pour définir le titre de la page, qui sera afficher dans l'onglet du navigateur.

< body >

C'est ici qu'on va placer le contenu de la page qui sera afficher par le navigateur et visible par l'utilisateur.

Je vous conseille de vous créer un "template" de cette structure de base dans l'IDE que vous utiliser pour coder afin de ne pas toujours le recopier à chaque nouvelle page web que vous créé.

Block vs Inline

Les éléments HTML ne sont pas tous affiché de la même façon par le navigateur. Il y a deux dispositions possibles, **block** ou **inline**, et chaque élément en a un des deux par défaut.

Block

Un élément ayant la valeur **block** commence toujours une nouvelle ligne et prendra toute la largeur disponible. Ex : **div**, **p**.

Inline

Avec Inline l'élément est disposé à la suite des autres sans commencer une nouvelle ligne. Ex : **a**, **span**.

Source

- [Documentation Mozilla](#)
- [Tutoriel HTML W3Schools](#)