**HTML**

**H**yper**T**ext **M**arkup **L**anguage (HTML) est le code utilisé pour structurer une page web et son contenu. Par exemple, le contenu de votre page pourra être structuré en un ensemble de paragraphes, une liste à puces ou avec des images et des tableaux de données. Comme le suggère le titre, cet article vous fournit les bases de compréhension du HTML et de ses fonctions.

HTML n'est pas un langage de programmation. C'est un langage de balises qui définit la structure de votre contenu. HTML se compose d'une série d'[**éléments**](https://developer.mozilla.org/fr/docs/Glossary/Element), utilisés pour entourer, ou envelopper, les diverses parties du contenu pour les faire apparaître ou agir d'une certaine façon. Les [balises](https://developer.mozilla.org/fr/docs/Glossary/Tag) entourantes peuvent être rendues par un mot ou une image lien hypertexte vers quelque chose d'autre, un texte en italique, une police plus grande ou plus petite, et ainsi de suite.

## **Qu'est-ce qu'une balise en HTML ?**

Une **balise**[**HTML**](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/) est un élément de base du système de codage HTML utilisé, en grande partie, pour la création d'une page Web. Ce sont les **balises HTML** qui permettent de hiérarchiser et de structurer le contenu d'une page Web grâce à une mise en forme particulière.

Il existe une multitude de **balises HTML** utilisées par les développeurs, webmasters, rédacteurs web, etc. Les plus connues et les plus simples à utiliser sont les balises de titrage, de mise en gras ou de mise en italique, etc. Toutes sont invisibles pour l'internaute qui navigue sur le Web de façon classique (on peut les apercevoir en activant un mode de navigation avec affichage du [code source](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203623-code-source-definition-traduction/)). Toutes sont délimitées par les signes < et >.

Le HTML repose sur une syntaxe simple et rigide, dont l'unité de base est la balise. Les balises vont toujours par deux, une pour ouvrir l'élément, l'autre pour le fermer.  Elles se construisent toujours de la même façon :

* un chevron < pour ouvrir la balise
* Le nom de l'élément contenu dans la balise
* Un attribut, si nécessaire, pour qualifier l'élément. Par exemples la source ou la balise alt d'une image. Pour ajouter l'attribut, à l'intérieur de la balise : espace, nom de l'attribut (src, alt, etc), signe égal (=) et une valeur s'il y a lieu, placée entre double quotes (""). Exemple : <img src="myImage.png">
* un chevron > pour fermer la balise

La balise d'ouverture contient le nom de l'élément et l'attribut. Une balise de fermeture termine l'élément, elle ne reprend que le nom de l'élément accompagné d'un slash (/). Par exemple, la balise paragraphe : <p> contenu du paragraphe </p>.

### [**Anatomie d'un document HTML**](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics#anatomie_dun_document_html)

* <!DOCTYPE html> — le doctype. Au début de HTML, dans les années 1991-1992, les doctypes étaient utilisés pour faire référence à des ensembles de règles qu'on pouvait utiliser pour dire qu'un document était du HTML « valide » et détecter les erreurs de balisage. Cependant, ceux-ci ne sont plus utilisés aujourd'hui et sont juste présents pour s'assurer que la page puisse fonctionner y compris sur les anciens navigateurs. Pour le moment, c'est tout ce qu'il y a à savoir à propos des doctypes.
* <html></html> — l'élément <html>. Cet élément encadre tout le contenu de la page. Cet élément est parfois appelé l'élément racine.
* <head></head> — l'élément <head>. Cet élément est utilisé comme un container pour toutes les choses qui font partie de la page HTML mais qui ne sont pas du contenu affiché. C'est dans cet élément qu'on mettra des [mots-clés](https://developer.mozilla.org/fr/docs/Glossary/Keyword), une description de la page qui apparaîtra sur les moteurs de recherche, les liens vers les fichiers CSS à utiliser pour la mise en forme, les déclarations des jeux de caractères à utiliser et ainsi de suite.
* <body></body> — l'élément [<body>](https://developer.mozilla.org/fr/docs/Web/HTML/Element/body). Cet élément est celui qui contient tout le contenu que vous souhaitez afficher pour qu'il soit vu par les visiteurs : cela peut être du texte, des images, des vidéos, des jeux, des pistes audio jouables, et ainsi de suite.
* <meta charset="utf-8"> — Cet élément définit le jeu de caractères qui devrait être utilisé pour le document et indique que c'est utf-8. utf-8 regroupe l'ensemble des caractères connus utilisés dans les différents langages humains. Généralement, utf-8 permet de gérer n'importe quel texte que vous pourriez utiliser sur la page. Il n'y a pas de raison de ne pas le définir, et il permet d'éviter certains problèmes plus tard.
* <title></title> — L'élément [<title>](https://developer.mozilla.org/fr/docs/Web/HTML/Element/title) définit le titre de votre page. C'est ce titre qui apparaîtra sur l'onglet lorsque la page sera chargée. C'est également ce titre qui sera utilisé pour décrire la page lorsque vous la placez dans vos marques-pages.

**CSS**

Les CSS (Cascading Style Sheets en anglais, ou « feuilles de style en cascade ») sont le code utilisé pour mettre en forme une page web. Les bases des CSS présentent ce qu'il faut savoir pour commencer. Nous répondrons à des questions comme : Comment rendre mon texte rouge ou noir ? Comment faire apparaître mon contenu à tel endroit de l'écran ? Comment décorer ma page web avec une image ou une couleur d'arrière-plan ?

### [**Anatomie d'une règle CSS**](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/CSS_basics#anatomie_dune_r%C3%A8gle_css)

Cette structure s'appelle **un ensemble de règles** (ou seulement « une règle »). Les différentes parties se nomment :

**Sélecteur**

C'est le nom de l'élément HTML situé au début de l'ensemble de règles. Il permet de sélectionner les éléments sur lesquels appliquer le style souhaité (en l'occurence, les éléments p). Pour mettre en forme un élément différent, il suffit de changer le sélecteur.

**Déclaration**

C'est une règle simple comme color: red; qui détermine les **propriétés\*\*** \*\*de l'élément que l'on veut mettre en forme.

**Propriétés**

Les différentes façons dont on peut mettre en forme un élément HTML (dans ce cas, color est une propriété des éléments p). En CSS, vous choisissez les différentes propriétés que vous voulez utiliser dans une règle CSS.

**Valeur de la propriété**

À droite de la propriété, après les deux points, on a **la valeur de la propriété**. Celle-ci permet de choisir une mise en forme parmi d'autres pour une propriété donnée (par exemple, il y a d'autres couleurs que red pour la propriété color).

### [**Les types de sélecteurs**](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/CSS_basics#les_diff%C3%A9rents_types_de_s%C3%A9lecteurs)

Il y a différents types de sélecteurs. Dans les exemples précédents, nous n'avons vu que **les sélecteurs d'élément** qui permettent de sélectionner les éléments HTML d'un type donné dans un document HTML. Mais ce n'est pas tout, il est possible de faire des sélections plus spécifiques. Voici quelques-uns des types de sélecteur les plus fréquents :

| **Nom du sélecteur** | **Ce qu'il sélectionne** | **Exemple** |
| --- | --- | --- |
| Sélecteur d'élément (parfois appelé « sélecteur de balise » ou « sélecteur de type ») | Tous les éléments HTML d'un type donné. | p sélectionne tous les <p> |
| Sélecteur d'ID | L'élément d'une page qui possède l'ID fourni (pour une page HTML donné, on ne peut avoir qu'un seul élément pour un ID donné). | #my-id sélectionne <p id="my-id"> ou <a id="my-id"> |
| Sélecteur de classe | Les éléments d'une page qui sont de la classe donnée (pour une page donnée, il est possible d'avoir plusieurs éléments qui partagent une même classe). | .my-class sélectionne <p class="my-class"> et <a class="my-class!"> |
| Sélecteur d'attribut | Les éléments de la page qui possèdent l'attribut donné. | img[src] sélectionne <img src="monimage.png"> mais pas <img> |
| Sélecteur de pseudo-classe | Les éléments donnés mais uniquement dans un certain état (par exemple quand on passe la souris dessus). | a:hover sélectionne <a> mais uniquement quand le pointeur de la souris est au-dessus du lien. |