

Les balises HTML et leur rôle

Apprendre le développement web > Apprendre le HTML : guides et didacticiels > Les balises HTML et leur rôle

Français ▼

Sur cette page

Pédagogie active

Aller plus loin

Prochaines étapes

Cet article aborde les bases de HTML : les balises et comment les utiliser.

Prérequis :

Vous devriez au préalable connaître la différence entre une page web et un site web et savoir comment créer un document HTML simple.

Objectifs :

Apprendre ce que sont les balises HTML et comment les utiliser correctement dans un document HTML.

HTML (*HyperText Markup Language*) est un langage descriptif utilisé pour structurer le contenu d'une page (ses textes, ses images, ses liens, etc.).

Un document HTML est un fichier texte qui contient des balises (ou *tag* en anglais). Ces balises doivent être utilisées d'une certaine façon pour décrire correctement la structure du document. Les balises indiquent au navigateur comment afficher le document, certaines balises permettent d'intégrer différents médias comme des images, des vidéos ou des musiques parmi le texte de la page.

Le navigateur n'affiche pas les balises telles quelles. Lorsqu'un utilisateur visite une page web, son navigateur *analyse* (ou *parse* en anglais) le document et *l'interprète* afin d'afficher la page web correctement. Par exemple, si le document contient une balise ``, le navigateur chargera l'image associée et affichera l'image *à la place* de la balise HTML.

La syntaxe des balises

Les balises HTML respectent une syntaxe simple et stricte :

- Un **chevron ouvrant** (`<`)
- Le **nom de la balise**
- Des attributs** (optionnels). Un espace, suivi du nom de l'attribut, d'un signe égal (=) et d'une valeur entre doubles quotes ("").
- Un **chevron fermant** (`>`)

Voici quelques exemples :

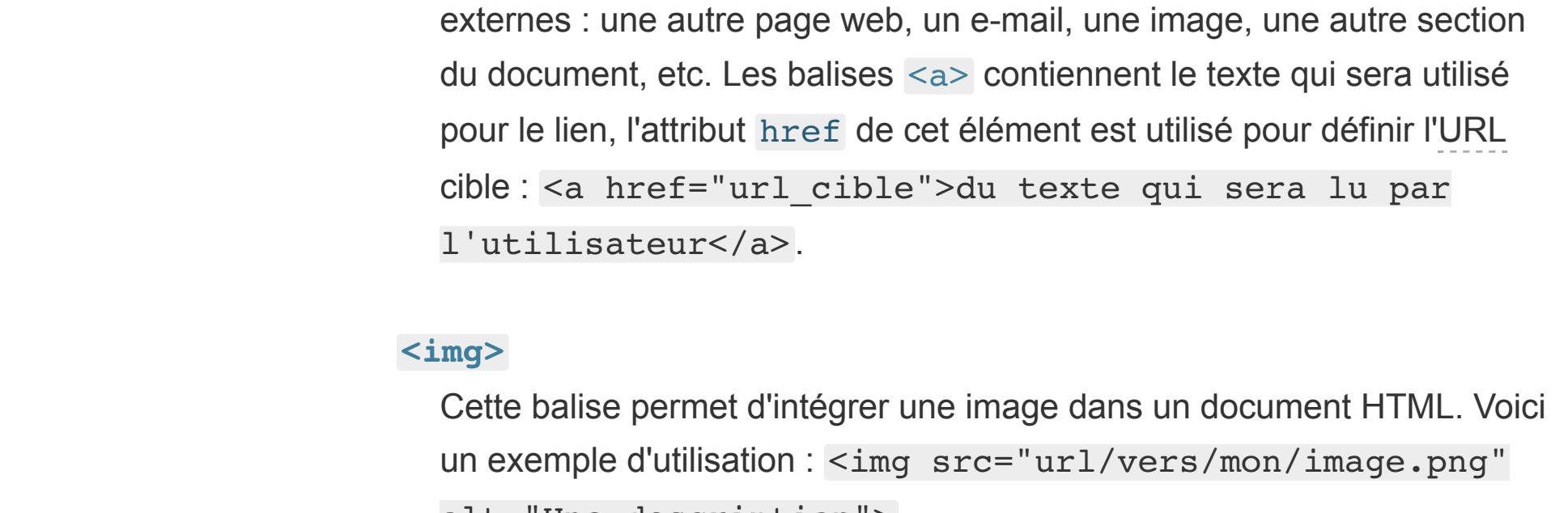
```
<article>
<meta charset="utf-8">

```

Les éléments HTML

Généralement, les balises fonctionnent par paires. La première balise est la balise ouvrante et la seconde est la balise fermante. Une balise fermante *doit* avoir le même nom que la balise ouvrante correspondante. De plus, une balise fermante doit contenir une barre oblique entre le chevron initial et le nom de la balise. Ainsi, si `<p>` est une balise ouvrante, `</p>` sera la balise fermante correspondante.

Un **élément HTML** se compose d'une balise ouvrante, d'un contenu textuel et d'une balise fermante :



La rigidité des conventions pour la fermeture des balises est utile afin de pouvoir imbriquer différents éléments les uns dans les autres :

```
1 <p>
2   Voici le début du paragraphe
3   <strong>
4     ici un texte important au sein du paragraphe
5   </strong>
6   et là, la fin du paragraphe.
7 </p>
```

Bonne pratique : Si vous omettez les balises fermantes, le navigateur aura les coudées franches pour déterminer comment clôturer un élément. Ces règles sont bien définies mais souvent contre-intuitives (donc difficiles à deviner). Afin de vous épargner des pertes de temps à venir, préférez fermer les éléments.

Les balises HTML

HTML contient environ 140 [balises](#) qui fournissent au navigateur des indications sur le sens d'un élément, son interprétation ou son affichage. Entre autres choses, les balises permettent de fournir des méta-données pour le document HTML, de mettre en avant certaines phrases, d'ajouter des fichiers multimédias ou de gérer des formulaires en ligne.

Voici quelques-unes des balises les plus fréquemment utilisées :

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

Ces balises permettent de définir des titres de différents niveaux : `h1` pour les grands titres et `h6` pour les titres des sections très spécifiques.

`<p>`

La balise utilisée pour créer des paragraphes. Ces paragraphes sont généralement (automatiquement) séparés par des sauts de ligne.

`<a>`

Cette balise est utilisée afin de créer des liens vers des ressources externes : une autre page web, un e-mail, une image, une autre section du document, etc. Les balises `<a>` contiennent le texte qui sera utilisé pour le lien, l'attribut `href` de cet élément est utilisé pour définir l'URL cible : `du texte qui sera lu par l'utilisateur`.

``

Cette balise permet d'intégrer une image dans un document HTML. Voici un exemple d'utilisation : ``

`<div>` et ``

Ces balises n'ont pas de signification particulière, elles permettent simplement de séparer des sections d'un document. La plupart du temps, ces balises sont utilisées pour la mise en forme et le script (dont nous discuterons plus tard).

``, `` et ``

Ces balises sont utilisées pour créer des listes. `` permet de définir une liste non-ordonnée et `` de définir une liste ordonnée. Pour chacune de ces listes, ce sera la balise `` qui permettra de définir un élément de la liste. Voici un exemple avec `` :

```
1 <ul>
2   <li>Café</li>
3   <li>Thé</li>
4   <li>Lait</li>
5 </ul>
```

N'hésitez pas à expérimenter avec ces balises. Dans les articles suivants, nous verrons comment intégrer des fichiers multimédias, des formulaires, des tableaux, etc. Dans tous les cas, vous pouvez faire vos propres tests avec [n'importe quelle balise HTML](#). Assurez-vous de bien choisir la bonne balise pour ce que vous souhaitez faire : la sémantique des éléments HTML est très importante, notamment pour les moteurs de recherches qui analysent et classent votre site.

Pédagogie active

Cet article ne contient pas encore de matériel interactif. *N'hésitez pas à contribuer.*

Aller plus loin

Pour l'instant, nous avons vu quelques règles simples concernant les éléments et les balises HTML. Mais c'est bien connu, chaque règle a ses exceptions.

Les balises « auto-fermantes » (ou balises vides)

Nous avons vu précédemment qu'un élément HTML était composé de texte entre deux balises. Cependant, certaines balises ne contiennent pas de texte. L'exemple le plus simple est ``. Le navigateur remplace la balise `` avec l'image indiquée, ignorant le texte de l'élément. Pour cette raison, `` n'a pas de balise fermante.

```
1 Here some text before an image element
2 
3 Here some text after an image element
```

Note : Attention à ne pas mélanger les éléments et les balises. Parfois, une balise seule permet de définir un élément mais la plupart du temps, les balises fonctionnent par paires.

Les éléments spéciaux

En HTML, il existe deux éléments spéciaux qui n'ont pas de balise. Ces éléments sont essentiels pour tout document HTML.

Le *doctype*

Le *doctype* (pour « type de document ») est une déclaration formelle, placée au tout début d'un document HTML. Elle indique que le document est écrit avec du HTML standard. Le doctype se présente de cette façon :

```
1 <!DOCTYPE html>
```

Si vous ne commencez pas votre document par `<!DOCTYPE html>`, les navigateurs afficheront votre document en *mode quirks*. Le mode *quirks* est le mode utilisé par le navigateur pour afficher les documents anciens ou malformés, écrits dans les années 1990 lorsque HTML était peu (voire pas du tout) standardisé et que chaque navigateur gérait HTML à sa façon.

Les commentaires

Les commentaires sont des éléments très particuliers. Ce sont des notes que vous pouvez utiliser pour annoter votre code HTML. Le navigateur n'affichera pas ces commentaires dans la page web (cependant, le code source peut être lu par n'importe qui et les commentaires seront donc publics comme le reste de la page).

En HTML, les commentaires sont écrits avec du texte contenu entre `<!--` et `-->`

```
1 <!-- Ceci est un commentaire. Il ne sera pas affiché -->
2
3 Ce texte s'affichera dans le navigateur.
```

Structure d'un document HTML

La structure de base d'un document HTML est définie avec un ensemble de balises spéciales. Les éléments définis dans ces balises ne doivent pas apparaître plus d'une fois dans le document (excepté pour l'élément `<title>`). Le navigateur gèrera les cas où ces éléments ne sont pas fournis (ce qui n'est pas conseillé).

`<html>`

Cet élément définit la *racine* du document. Chaque document HTML ne contient qu'une *racine*. Tous les autres éléments doivent être placés dans cet élément.

`<head>`

Cet élément définit la *tête* du document. Le navigateur n'affichera pas cet élément qui ne contient que des méta-données, dont le titre et des informations descriptives. Les navigateurs pourront utiliser ces méta-données pour améliorer l'ergonomie de la page (nous le verrons par la suite).

`<body>`

Cet élément définit le *corps* du document. Il n'y a qu'un seul élément `body` dans un document HTML et celui-ci est toujours placé après la tête. L'utilisateur verra tout ce qui est placé dans cet élément.

`<title>`

Cet élément définit le *titre* d'un document. Le *titre* est le seul élément HTML obligatoire et il est placé dans la tête. En effet, le titre est une des méta-données utilisée par le navigateur (il est utilisé pour le titre de la fenêtre, le titre de l'onglet et aussi dans les résultats d'un moteur de recherche).

Formel ou valide ?

Voici le document HTML formel le plus simple qu'on puisse écrire :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Un document HTML formel</title>
5   </head>
6   <body>
7     <!-- Du contenu pour l'utilisateur ici -->
8   </body>
9 </html>
```

Si on retire les différentes balises optionnelles, on obtient alors le document HTML *valide* le plus simple qui puisse être écrit :

```
1 <!DOCTYPE html>
2 <title>Et voilà un tout petit document HTML</title>
```

Bonne pratique : Nous vous recommandons d'utiliser la structure formelle. Si vous ne séparez pas clairement le contenu de `<head>` du contenu de `<body>`, il sera plus facile de commettre des erreurs qui entraîneront un comportement étrange du navigateur.

Les éléments qui se chevauchent

Pour finir notre aperçu sur les éléments HTML, précisons que les éléments peuvent être *imbriqués* mais ne peuvent pas se *chevaucher*.

Comment faire se chevaucher des éléments ? Voici un exemple :

```
1 <p>
2   le début de mon paragraphe
3   <strong>
4     du texte important
5   </p>
6 <p>
7   un deuxième paragraphe
8   </strong>
9   et l'élément strong a été bouclé
10  juste avant
11 </p>
```

Dans l'exemple ci-dessus, l'élément `strong` chevauche ceux des paragraphes. Ceci est une erreur (qui n'est pas autorisée par HTML). Il faut plutôt écrire :

```
1 <p>
2   mon paragraphe commence ici
3   <strong>
4     du texte important
5   </strong>
6 </p>
7 <p>
8   <strong>
9     un deuxième paragraphe
10  </strong>
11  dont le début était important aussi
12 </p>
```

Dans des cas simples comme celui-ci, le navigateur pourra deviner ce qui était voulu mais généralement de telles erreurs pourront avoir des répercussions importantes (par exemple avec CSS ou JavaScript que nous verrons bientôt). Par exemple, si nous avions utilisé une balise `<div>` à la place des balises `<p>` et `` :

```
1 <div>
2   le début de mon paragraphe
3   <div>
4     du texte important
5   </div>
6   <div>
7     un deuxième paragraphe
8   </div>
9   la fin du paragraphe
10 </div>
```

Comment savoir si les éléments se chevauchent ou non ? C'est impossible car il n'y a plus de chevauchement mais une imbrication d'éléments `div` ! Attention donc si vous souhaitez maîtriser le comportement de vos futurs documents HTML.

Vérifier le code HTML

Une bonne pratique consiste à valider son code pour s'assurer qu'il est correct. Le meilleur outil pour cela est [le validateur HTML du W3C](#). C'est un outil en ligne qui analyse votre document HTML et vous indique si celui-ci respecte les spécifications HTML. Si le document est invalide, le validateur fournira des indications quant aux erreurs qu'il a trouvées.

Note : Étant donné la méthode utilisée par le validateur, il est préférable de se concentrer sur la première erreur relevée dans l'analyse. En effet, une seule erreur peut en déclencher d'autres en cascades. Si le résultat de l'analyse comporte une myriade d'erreurs, commencez par résoudre la première, cela permettra éventuellement de résoudre des erreurs suivantes.

Prochaines étapes

Grâce aux notions abordées dans cet article, vous devriez être en mesure d'écrire votre premier document HTML complet et d'obtenir une première version d'un site web fonctionnel.

- Pour que cette page soit (plus) belle, vous pouvez lire [l'article sur les propriétés CSS et comment les utiliser](#).
- Pour savoir comment publier votre page web, vous pouvez lire [comment transférer ses fichiers vers un serveur web](#).

⌕

Dernière modification :

23 mars 2019, by MDN contributors

Recevez le meilleur du développement web

Get the latest and greatest from MDN delivered straight to your inbox.

Cette lettre d'information est uniquement disponible en anglais pour l'instant.

Sign up now