

Balises HTML

3 min (en anglais)

Vous connaissez maintenant tous les éléments de base et la configuration dont vous avez besoin pour structurer une page HTML et ajouter différents types de contenu. Avec l'aide de CSS, vous allez très bientôt créer de beaux sites web !

Bien que certaines balises aient un objectif très spécifique, comme les balises d'image et de vidéo, la plupart des balises sont utilisées pour décrire le contenu qu'elles entourent, ce qui nous aide à modifier et à styliser notre contenu par la suite. Il y a apparemment un nombre infini de balises à utiliser (beaucoup plus que ce que nous avons enseigné). Savoir quand utiliser chacun d'entre eux dépend de la façon dont vous souhaitez décrire le contenu de votre HTML. Des balises descriptives et bien choisies sont l'une des clés d'un développement web de haute qualité. Une liste complète des balises HTML disponibles est disponible [dans la documentation de Mozilla](#).

Passons en revue ce que vous avez appris dans cette leçon :

1. La déclaration doit toujours être la première ligne de code de vos fichiers HTML. Cela permet au navigateur de savoir à quelle version de HTML il peut s'attendre. `<!DOCTYPE html>`
2. L'élément contiendra tout votre code HTML. `<html>`
3. Les informations sur la page Web, comme le titre, appartiennent à la page de la page. `<head>`
4. Vous pouvez ajouter un titre à votre page web en utilisant l'élément, à l'intérieur de l'en-tête. `<title>`
5. Le titre d'une page Web apparaît dans l'onglet d'un navigateur.
6. Les balises d'ancrage () sont utilisées pour créer des liens vers des pages internes, des pages externes ou du contenu sur la même page. `<a>`
7. Vous pouvez créer des sections sur une page Web et y accéder à l'aide de balises et en ajoutant des s aux éléments auxquels vous souhaitez accéder. `<a>id`
8. Les espaces entre les éléments HTML facilitent la lecture du code sans modifier l'apparence des éléments dans le navigateur.

9. L'indentation permet également de faciliter la lecture du code. Il rend visibles les relations parent-enfant.
10. Les commentaires sont écrits en HTML à l'aide de la syntaxe suivante :

```
<!-- comment -->
```

Prenez le temps de modifier l'espace de travail que vous avez créé et observez comment il change !

TABLEAUX HTML

Révision

1 min (en anglais)

Beau travail ! Dans cette leçon, nous avons appris à créer un tableau, à y ajouter des données et à le sectionner en parties plus petites qui le rendent plus facile à lire.

Passons en revue ce que nous avons appris jusqu'à présent :

- L'élément crée une table. `<table>`
- L'élément ajoute des lignes à une table. `<tr>`
- Pour ajouter des données à une ligne, vous pouvez utiliser l'élément. `<td>`
- Les en-têtes des tableaux clarifient la signification des données. Les en-têtes sont ajoutés avec l'élément. `<th>`
- Les données de table peuvent s'étendre sur des colonnes à l'aide de l'attribut. `colspan`
- Les données de table peuvent s'étendre sur des lignes à l'aide de l'attribut. `rowspan`
- Les tableaux peuvent être divisés en trois sections principales : une tête, un corps et un pied de page.
- La tête d'une table est créée avec l'élément. `<thead>`
- Le corps d'une table est créé avec l'élément. `<tbody>`
- Le pied de page d'un tableau est créé avec l'élément. `<tfoot>`
- Toutes les propriétés CSS que vous avez apprises dans ce cours peuvent être appliquées aux tables et à leurs données.

FORMULAIRES HTML

Révision

1 min (en anglais)

Beau travail d'interaction avec l'élément extrêmement commun et utile ! `<form>`

Dans cette leçon, nous avons passé en revue :

- Le but d'un est de permettre aux utilisateurs de saisir des informations et de les envoyer. `<form>`
- L'attribut 's détermine l'emplacement des informations du formulaire. `<form>action`
- L'attribut de détermine la manière dont les informations sont envoyées et traitées. `<form>method`
- Pour ajouter des champs permettant aux utilisateurs de saisir des informations, nous utilisons l'élément et définissons l'attribut sur un champ de notre choix : `<input>type`
 - Paramètre sur crée un champ de ligne unique pour la saisie de texte. `type="text"`
 - La valeur sur crée un champ de ligne unique qui censure la saisie de texte. `type="password"`
 - Le paramètre sur crée un champ de ligne unique pour la saisie numérique. `type="number"`
 - Le paramètre sur crée un curseur permettant de sélectionner parmi une plage de nombres. `type="range"`
 - La définition de crée une seule case à cocher qui peut être associée à d'autres cases à cocher. `type="checkbox"`
 - Le paramètre sur crée une case d'option qui peut être associée à d'autres cases d'option. `type="radio"`
 - La définition de et l'ajout de l'attribut associeront le avec un élément si le of et le of sont identiques. `type="text"list<input><datalist>list<input>id<datalist>`
 - Paramètre pour créer un bouton d'envoi. `type="submit"`
- Un élément est renseigné avec des éléments et affiche une liste déroulante de sélection. `<select><option>`
- Un élément est renseigné avec des éléments et fonctionne avec un pour effectuer une recherche dans les choix. `<datalist><option><input>`

- Un élément est un champ de saisie de texte qui possède une zone personnalisable. `<textarea>`
- Lorsque a est soumis, les champs qui acceptent l'entrée et les champs de ces champs sont envoyés par paires. `<form>namevalue`

L'utilisation de l'élément en conjonction avec les autres éléments énumérés ci-dessus nous permet de créer des sites qui prennent en considération les désirs et les besoins de nos utilisateurs. Profitez-en pour mettre en pratique ce que vous avez appris ! `<form>`

Instructions

Si vous voulez vous mettre au défi :

- Créez un formulaire avec plusieurs champs qui acceptent la saisie de l'utilisateur.
- Ajoutez deux ensembles distincts de cases d'option ou de cases à cocher.
- Recherchez comment ajouter un *espace réservé* (hint hint) à un champ de saisie. Un espace réservé s'affiche lorsque le champ n'a pas, il disparaît une fois que l'utilisateur a tapé ou sélectionné quelque chose dans le champ `value`

VALIDATION DU FORMULAIRE

Révision

<1 min

Travail génial d'ajouter la validation côté client à s ! `<form>`

Récapitulons rapidement :

- Les validations côté client se produisent dans le navigateur avant que les informations ne soient envoyées à un serveur.
- L'ajout de l'attribut à un élément lié à l'entrée permet de valider que le champ de saisie contient des informations. `required`

- L'affectation d'une valeur à l'attribut d'un élément d'entrée numérique valide une valeur minimale acceptable. `min`
- L'affectation d'une valeur à l'attribut d'un élément d'entrée numérique valide une valeur maximale acceptable. `max`
- L'affectation d'une valeur à l'attribut d'un élément de saisie de texte validera un nombre minimum acceptable de caractères. `minlength`
- L'affectation d'une valeur à l'attribut d'un élément de saisie de texte validera un nombre maximal acceptable de caractères. `maxlength`
- L'affectation d'une expression régulière à fait correspondre l'entrée à l'expression régulière fournie `pattern`
- Si les validations d'un ne réussissent pas, l'utilisateur reçoit un message expliquant pourquoi et ne peut pas être soumis. `<form><form>`

Ces vérifications rapides permettent de s'assurer que les données saisies sont correctes et sûres pour nos serveurs. Cela permet également de donner aux utilisateurs un retour immédiat sur ce qu'ils doivent corriger au lieu d'avoir à attendre qu'un serveur renvoie ces informations.

HTML SÉMANTIQUE

En-tête et navigation

5 min (en anglais)

Jetons un coup d'œil à quelques éléments sémantiques qui aident à la structure d'une page Web. Un `<header>` est un conteneur généralement destiné aux liens de navigation ou au contenu d'introduction contenant des en-têtes. `<h1><h6>`

L'exemple ci-dessous montre en action : `<header>`

```
<header>
  <h1>
    Tout ce que vous devez savoir sur la pizza !
  </h1>
</header>
```

Cela peut être comparé au code ci-dessous qui utilise une balise au lieu d'une balise : `<div><header>`

```
<div id="header">
  <h1>
    Tout ce que vous devez savoir sur la pizza !
  </h1>
</div>
```

En utilisant une balise, notre code devient plus facile à lire. Il est beaucoup plus facile d'identifier ce qu'il y a à l'intérieur des balises parentes de , par opposition à une balise qui ne fournirait aucun détail sur ce qu'il y avait à l'intérieur de la balise. `<header><h1><div>`

Un `<nav>` permet de définir un bloc de liens de navigation tels que des menus et des tables des matières. Il est important de noter qu'il peut être utilisé à l'intérieur de l'élément mais peut également être utilisé seul. `<nav><header>`

Jetons un coup d'œil à l'exemple ci-dessous :

```
<header>
  <nav>
    <ul>
      <li><a href="#home">Accueil</a></li>
      <li><a href="#about ">À propos </a></li>
    </ul>
  </nav>
</header>
```

En l'utilisant comme un moyen d'étiqueter nos liens de navigation, il sera plus facile non seulement pour nous, mais aussi pour les navigateurs Web et les lecteurs d'écran de lire le code. `<nav>`

Maintenant que nous avons appris à connaître les éléments and, ajoutons-les dans notre code ! `<header><nav>`

Instructions

1.

Dans l'éditeur de code, recherchez la balise et remplacez-la par. `<div id="header"><header>`

Note : Lorsque vous modifiez une balise d'ouverture, vous devez trouver la balise de fermeture correspondante et la modifier également. Si ce n'est pas le cas, vous verrez du rouge dans votre code pour indiquer l'erreur.

Coincé ? Obtenir un indice

2.

Maintenant, trouvez la balise et remplacez-la par. `<div id="nav"><nav>`

HTML SÉMANTIQUE

Principal et pied de page

4 min (en anglais)

Deux autres éléments structurels sont `<main>` et `<footer>`. Ces éléments, ainsi que `<nav>` et `<header>` aident à décrire l'emplacement d'un élément sur la base des normes de développement Web conventionnelles. `<main><footer><nav><header>`

L'élément `<main>` est utilisé pour encapsuler le contenu dominant au sein d'une page Web. Cette balise est distincte de la balise `<footer>` et le d'une page web puisque ces éléments ne contiennent pas le contenu principal. En utilisant plutôt qu'un élément, les lecteurs d'écran et les navigateurs Web sont mieux à même d'identifier que tout ce qui se trouve à l'intérieur de la balise constitue la majeure partie du contenu. `<nav><main><div>`

Alors, à quoi ressemble l'intégration dans notre code ? C'est une excellente question. `<main>`

```
<Principal>
  <en-tête>
    <h1>Types de sports</h1>
  </en-tête>
  <article>
    <h3>Baseball</h3>
    <p> Le premier match de baseball a eu lieu à Cooperstown, dans
l'État de New York, à l'été 1839.
    </p>
  </article>
</main>
```

Comme nous l'avons vu ci-dessus, contient une balise et avec des éléments enfants qui contiennent les informations les plus importantes liées à la page. `<main><article><header>`

Le contenu au bas de l'information sur le sujet est connu sous le nom de *pied de page*, indiqué par le `<footer>` élément. Le pied de page contient des informations telles que :

- Coordonnées
- Informations sur les droits d'auteur
- Conditions d'utilisation
- Plan du site
- Référence aux liens vers le haut de la page

Par exemple :

```
<footer>
  <p>Envoyez-moi un e-mail à Codey@Codecademy.com</p>
</footer>
```

Dans l'exemple ci-dessus, le pied de page est utilisé pour contenir les informations de contact. La balise est distincte de l'élément et généralement située au bas du contenu. `<footer><main>`

HTML SÉMANTIQUE

L'élément d'aparté

2 min (en anglais)

Maintenant que nous avons couvert le corps du HTML sémantique, concentrons-nous sur ce qui peut aller dans le corps. Les deux éléments sur lesquels nous allons nous concentrer maintenant sont et. `<section><article>`

`<section>` Définit les éléments d'un document, tels que les chapitres, les titres ou toute autre zone du document ayant le même thème. Par exemple, le contenu ayant le même thème, comme les articles sur le cricket, peut être placé sous un seul fichier. La page d'accueil d'un site Web peut être divisée en sections pour l'introduction, les actualités et les coordonnées. `<section>`

Voici un exemple d'utilisation :<section>

```
<section>
  <h2>Faits amusants sur le cricket</h2>
</section>
```

Dans le code ci-dessus, nous avons créé un élément pour encapsuler le code. Dans nous avons ajouté un élément en tant qu'en-tête.<section><section><h2>

Le <article> contient du contenu qui a du sens par lui-même. Peut contenir du contenu tel que des articles, des blogs, des commentaires, des magazines, etc. Une balise aiderait quelqu'un qui utilise un lecteur d'écran à comprendre où le contenu de l'article (qui peut contenir une combinaison de texte, d'images, d'audio, etc.) commence et se termine.<article><article>

Voici un exemple d'utilisation :<article>

```
<section>
  <h2>Fun Facts About Cricket</h2>
  <article>
    <p>Un seul match de cricket peut durer jusqu'à 5 jours.</p>
  </article>
</section>
```

Dans le code ci-dessus, l'élément contenant un fait sur le cricket a été placé à l'intérieur de l'élément. Il est important de noter qu'un élément peut également être placé dans un élément en fonction du contexte.<article><section><section><article>

HTML SÉMANTIQUE

L'élément d'aparté

2 min (en anglais)

Le <aside> est utilisé pour marquer des informations supplémentaires qui peuvent améliorer un autre élément, mais qui ne sont pas nécessaires pour comprendre le contenu principal. Cet élément peut être utilisé avec d'autres éléments tels que ou. Certaines utilisations courantes de l'élément sont les suivantes :<article><section><aside>

- Bibliographies
- Notes
- Commentaires
- [Extraire des guillemets](#)
- Encadrés éditoriaux
- Informations complémentaires

Voici un exemple d'utilisation aux côtés de : `<aside><article>`

```
<article>
  <p>La première Série mondiale a été disputée entre Pittsburghet
  Boston en 1903 et était une série de neuf matchs. </p>
</article>
<aside>
  <p>
    Babe Ruth a déclaré un jour : « On se souvient des héros, mais les
    légendes ne meurent jamais. »
  </p>
</aside>
```

Comme indiqué ci-dessus, les informations contenues dans le sont le contenu important. Pendant ce temps, l'information contenue dans le améliore l'information, mais n'est pas nécessaire pour la comprendre.

`<article><aside><article>`

HTML SÉMANTIQUE

Figure et légende de la figure

5 min (en anglais)

Avec, nous avons appris que nous pouvons mettre des informations supplémentaires à côté d'un élément de contenu principal, mais que se passe-t-il si nous voulons ajouter une image ou une illustration ? C'est là qu'il faut intervenir. `<aside><figure><figcaption>`

`<figure>` est un élément utilisé pour encapsuler des médias tels qu'une image, une illustration, un diagramme, un extrait de code, etc., qui est référencé dans le flux principal du document.

```
<figure>
  
</figure>
```

Dans ce code, nous avons créé un élément afin de pouvoir encapsuler notre balise. Dans , nous avons utilisé la balise pour insérer une image sur la page Web. Nous avons utilisé l'attribut dans la balise afin de pouvoir lier la source de l'image. `<figure></figure>src`

Il est possible d'ajouter une légende à l'image à l'aide de `<figcaption>`

`<figcaption>` est un élément utilisé pour décrire le média dans la balise. Habituellement, ira à l'intérieur. C'est différent de l'utilisation d'un élément pour décrire le contenu ; Si nous décidons de changer l'emplacement de , la balise de paragraphe peut être déplacée de la figure tandis que a se déplacera avec la figure. Ceci est utile pour regrouper une image avec une légende.`<figure><figcaption></figure><p></figure><figcaption>`

```
<figure>
  
  <figcaption>Cette image montre des personnages
d'Overwatch.</figcaption>
</chiffre>
```

Dans l'exemple ci-dessus, nous avons ajouté un dans l'élément pour décrire l'image de l'exemple précédent. Cela permet de regrouper le contenu avec le contenu.`<figcaption><figure></figure><figcaption>`

Bien que le contenu soit lié au flux principal du document, sa position est indépendante. Cela signifie que vous pouvez le supprimer ou le déplacer ailleurs sans affecter le flux du document.`<figure>`

SEMANTIC HTML

Audio and Attributes

5 min

Now that we learned about text-based content, let us dig into `<audio>`! Surely everyone needs —how else would you listen to your Korean hip hop? `<audio>`

The `<audio>` element is used to embed audio content into a document. Like `<video>`, it uses `src` to link the audio source.

```
<audio>
  <source src="iAmAnAudioFile.mp3" type="audio/mp3">
</audio>
```

In this example, we created an `<audio>` element. Then we created a `<source>` element to encapsulate our audio link. In this case, `iAmAnAudioFile.mp3` is our audio file. Then we specified the type by using `type="audio/mp3"` and named what kind of audio it is. Although not always necessary, it's recommended that we state the type of audio as it helps the browser identify it more easily and determine if that type of audio file is supported by the browser.

We linked our audio file into the browser but now we need to give it controls. This is where *attributes* come in. [Attributes](#) provide additional information about an element.

Attributes allow us to do many different things to our audio file. There are many attributes for but today we're going to be focusing on `controls` and `src`.

- `controls`: automatically displays the audio controls into the browser such as play and mute.
- `src`: specifies the URL of the audio file.

As you might have noticed, we already used the `src` attribute in the example code above. Most attributes go in the opening tag of `<audio>`. For example, here's how we could add both autoplay functionality and audio controls:

```
<audio autoplay controls>
```

You can find other attributes here: [Useful attributes](#).

Instructions

1.

In the code editor, create an `<audio>` tag with the `controls` attribute directly under the commented line.

2.

1.

2.

Add a source for the audio inside the tag by using this code: `<audio>`

```
<source src="" type="audio/mp3">
```

Use the following URL as the source for the

audio: `https://content.codecademy.com/courses/SemanticHTML/dogBarking.mp3`

HTML SÉMANTIQUE

Vidéo et intégration

6 min (en anglais seulement)

Comme nous l'avons vu dans l'exercice précédent, le contenu multimédia peut être un complément utile à un site Web. En utilisant un élément, nous pouvons ajouter des vidéos à notre site Web. Le `<video></video>` indique clairement qu'un développeur tente d'afficher une vidéo à l'utilisateur.

Voici quelques attributs qui peuvent modifier la lecture d'une vidéo :

- `controls`: Une fois ajouté, un bouton de lecture/pause sera ajouté à la vidéo ainsi qu'un contrôle du volume et une option plein écran.
- `autoplay`: L'attribut qui permet la lecture automatique d'une vidéo dès que la page est chargée.
- `loop`: cet attribut permet de lire la vidéo en continu en boucle.

Vous trouverez ci-dessous un exemple d'utilisation de l'attribut `controls`

: `<video>`

```
<video src="coding.mp4 » contrôles>Vidéo non prise en charge</vidéo>
```

Dans le code ci-dessus, un fichier vidéo nommé est en cours de lecture. La mention « Vidéo non prise en charge » ne s'affiche que si le navigateur n'est pas en mesure d'afficher la vidéo. `coding.mp4`

Une autre balise qui peut être utilisée pour incorporer du contenu multimédia dans une page est la balise, qui peut intégrer n'importe quel contenu multimédia, y compris des vidéos, des fichiers audio et des gifs provenant d'une source externe. Cela signifie que les sites Web qui ont un bouton

d'intégration ont une certaine forme de contenu multimédia qui peut être ajouté à d'autres sites Web. La balise est une balise à fermeture automatique, contrairement à l'élément. Notez que est une balise obsolète et que d'autres alternatives, telles que , et , doivent être utilisées à sa place, mais sont enseignées à des fins d'héritage. `<embed>``<embed>``<video>``<embed>``<video>``<audio>```

Ci-dessous, nous allons voir comment il est utilisé en action. `<embed>`

```
<embed src="download.gif" />
```

Dans l'exemple ci-dessus, est utilisé pour ajouter un gif à partir d'un fichier local connu sous le nom de . L'intégration peut être utilisée pour ajouter des fichiers locaux ainsi que du contenu multimédia directement à partir d'autres sites Web. `<embed>`download.gif

Instructions

1.

Ajoutez l'URL de la vidéo suivante avec l'attribut et enveloppez-la dans une balise vidéo d'ouverture et de fermeture. `controls`

```
https://content.codecademy.com/courses/SemanticHTML/dog-video.mp4
```

Entre la balise d'ouverture et la balise de fermeture, vous pouvez ajouter un texte qui s'affichera si la vidéo ne peut pas être chargée. Ce texte est également utile pour les lecteurs d'écran ! `<video>`

Coincé? Obtenir un indice

2.

Ajoutez le gif suivant dans votre code à l'aide de la balise et de l'attribut

: `<embed>`src

```
https://content.codecademy.com/courses/SemanticHTML/dog-on-beach.gif
```

HTML SÉMANTIQUE

Révision

1 min (en anglais)

Félicitations pour avoir terminé cette leçon sur le HTML sémantique ! Maintenant que vous connaissez les avantages du HTML sémantique et comment l'utiliser, vous pouvez intégrer des éléments sémantiques dans votre site web pour le rendre plus accessible et rendre le code plus facile à lire.

Passons en revue certains des sujets que nous avons abordés tout au long de la leçon :

- Le HTML sémantique introduit du sens à une page par le biais d'éléments spécifiques qui fournissent un contexte quant à ce qui se trouve entre les balises.
- Le HTML sémantique est une norme moderne qui rend un site Web accessible aux personnes qui utilisent des lecteurs d'écran pour traduire la page Web et améliore le référencement de votre site Web.
- `<header>`, et créez la structure de base de la page Web. `<nav><main><footer>`
- `<section>` Définit les éléments d'un document, tels que les chapitres, les titres ou toute autre zone du document ayant le même thème.
- `<article>` contient du contenu qui a du sens en soi, comme des articles, des blogs, des commentaires, etc.
- `<aside>` contient des informations qui sont liées au contenu principal, mais qui ne sont pas nécessaires pour comprendre l'information dominante.
- `<figure>` Encapsule tous les types de supports.
- `<figcaption>` est utilisé pour décrire le média dans `<figure>`
- `<video>`, et sont utilisés pour les fichiers multimédias. `<embed><audio>`

Maintenant, appliquez ces connaissances pour devenir un meilleur développeur Web.