

Créer un document HTML simple

Structure d'un document HTML

Un document HTML commence par la balise `<html>` et finit par la balise `</html>`. Il contient également un *en-tête* décrivant le titre de la page, puis un *corps* dans lequel se trouve le contenu de la page.

Il est conseillé d'indiquer dans la page HTML le *prologue du type de document*, c'est-à-dire une référence à la norme HTML utilisée, afin de spécifier le standard utilisé pour le codage de la page. Cette déclaration se fait par une ligne du type : `<!DOCTYPE html>`

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Inscrire un titre ici</title>
    <!-- On peut avoir d'autres méta-données ici -->
  </head>
  <body>
    <!-- Ici, on placera tout le contenu à destination
    de l'utilisateur -->
  </body>
</html>
```

Qu'est-ce qu'un document HTML ?

Un document HTML ou une page web est simplement composé de texte structuré au sein de balise (une balise étant un mot-clé encadré par des chevrons ("`<`" "`>`"), par exemple, dans l'exemple précédent `<html>` et `<body>` sont deux des éléments qui sont utilisés). La plupart des balises fonctionnent par paires (on a alors une balise ouvrante `<body>` et une balise fermante correspondante : `</body>`). Un élément est une chaîne de texte entre deux balises.

La plupart des programmes (le plus souvent des navigateurs) traitent ces balises pour générer (ou « rendre ») le site que l'utilisateur peut visualiser.

Étant donné que HTML est un format textuel, il est possible d'écrire des fichiers HTML avec n'importe quel éditeur de texte.

Le doctype

La chaîne de texte suivante est appelée doctype (qui est la contraction, anglaise, de « document » et « type »).

```
<!DOCTYPE html>
```

En ce qui concerne HTML, le *doctype* est un reliquat historique. Si vous ne commencez pas votre document par `<!DOCTYPE html>`, les navigateurs afficheront votre document en mode quirks, le mode utilisé par le navigateur pour afficher les documents anciens, écrits dans les années 1990 lorsque HTML était peu standardisé et que chaque navigateur gérait HTML à sa façon.

Le *doctype* déclenche l'utilisation du mode *standard*. Cela fait que la page sera affichée de façon prévisible et cohérente par rapport aux standards définis.

L'élément `<html>`

Tout le document (en dehors du *doctype*) est contenu entre les balises `<html>` et `</html>`. Il ne peut y avoir qu'un seul élément `<html>` par document.

Il est possible d'oublier `<html>` et le navigateur le comprendra de façon implicite. Cependant, afin de pouvoir manipuler le document dans son ensemble, `<html>` sera nécessaire. Par exemple, on utilise `<html lang="fr">` pour indiquer que le document entier est écrit en français.

Les éléments `<head>` et `<body>`

Au sein de l'élément `<html>`, le document aura une « tête » (*head* en anglais) située entre les balises `<head>` et `</head>` et un « corps » (*body* en anglais) situé entre les balises `<body>` et `</body>`.

La tête de la page contient les méta-données (les données qui décrivent le document) mais pas le contenu principal affiché pour l'utilisateur. Ces méta-données peuvent être utilisées par les moteurs de recherches, pour ajouter des liens vers des feuilles de style CSS, etc.

HTML est très souple à propos de la structure du document, si vous oubliez les éléments `<head>` et `<body>`, ils seront implicitement ajoutés. Voici un document sans `<html>` ni `<head>` ni `<body>` :

```
<!DOCTYPE html>
<title>Ceci est un document HTML</title>
```

```
Coucou monde ! (<i>Hello world!</i>)
```

L'élément `<title>` : le titre du document

Dans la tête du document, on écrira un titre concis et équivoque qui décrit le document. Il faut que le titre ait du sens sans autre contexte :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mon exemple HTML</title>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

Définir le jeu de caractères utilisé

Les ordinateurs enregistrent toutes les informations sous forme de binaire(0 et 1). Lorsqu'il s'agit de représenter des chaînes de caractères, il existe un standard appelé [UTF-8](#) qui permet d'associer des nombres binaires à l'ensemble des glyphes contenus dans les différents langages humains.

D'autres formes d'encodage sont encore utilisées par ailleurs, aussi, pour garantir le résultat obtenu, on déclare explicitement qu'on utilise UTF-8 dans le fichier HTML. C'est pour cette raison que nous avons ajouté `<meta charset="utf-8">` dans le premier exemple.

Découper une page web en sections logiques

Les sections de base

Les sites web ont chacun leur style mais tous ont tendance à partager des composants communs :

- l'en-tête (*header*) :

Il s'agit généralement d'une grande bande horizontale, située en haut du site avec un grand titre et/ou un logo. C'est à cet endroit qu'on a toujours la même information sur le site, quelle que soit la page du site sur laquelle on est.

- la barre de navigation :

Celle-ci permet de faire référence aux différentes sections du site via des menus ou des onglets. Comme pour l'en-tête, la barre de navigation sera généralement présente sur l'ensemble du site. De nombreux concepteurs web considèrent que la barre de navigation fait partie intégrale de l'en-tête mais ce n'est pas strictement nécessaire.

- le contenu principal

Une zone importante, située au centre dont le contenu est spécifique selon la page visitée.

- le panneau ou la barre latéral-e (*sidebar*)

Il contient des informations périphériques, des liens, des citations, des publicités. Généralement, ce panneau dépend du contenu principal de la page mais on trouve également des panneaux latéraux qui agissent comme des menus de navigation secondaires.

- le pied de page (*footer*)

C'est ici qu'on trouvera les informations de contact, de copyright... C'est un endroit où on place des informations communes à l'ensemble du site mais qui ne sont pas primordiales pour le site en tant que tel. Le pied de page est parfois utilisé par les outils de référencement afin de fournir un accès rapide à certaines parties du site.

Pour structurer une page avec cette sémantique, HTML fournit des balises dédiées :

- Pour l'en-tête : [`<header>`](#)
- Pour la barre de navigation : [`<nav>`](#)

- Pour le contenu principal : [`<main>`](#) ainsi qu'une association d'éléments [`<article>`](#), [`<section>`](#) et [`<div>`](#) pour créer des sous-sections
- Pour la barre latérale : [`<aside>`](#)
- Pour le pied de page : [`<footer>`](#)

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 :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Un document HTML formel</title>
  </head>
  <body>
    <!-- Du contenu pour l'utilisateur ici -->
  </body>
</html>
```

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

```
<!DOCTYPE html>
<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.

Comment intégrer une image à une page web ?

Pour mettre une image simple sur une page web, nous utiliserons l'élément ``. C'est un élément vide (ce qui signifie qu'il ne contient ni texte ni balise de fermeture) qui demande au moins un attribut pour fonctionner — `src` (souvent appelé par son nom entier: *source*). L'attribut `src` contient un chemin pointant vers l'image que vous voulez intégrer, qui peut être une URL absolue ou relative, de la même manière que l'élément `<a>` `href=` attribue des valeurs.

Donc, par exemple, si votre image s'appelle `img.jpg`, et qu'elle est située dans le même répertoire que votre page HTML, vous pouvez intégrer cette image comme ceci (URL relative) :

```

```

Et si cette image se trouve dans un sous-répertoire `images` situé dans le même dossier que la page HTML (ce que Google recommande pour [SEO](#)/dans un but d'indexation et d'optimisation de la recherche), alors vous l'intégrerez comme ceci :

```

```

Vous pouvez intégrer l'image en utilisant son URL absolue, par exemple :

```

```

Ce n'est pas très efficace, cela fait travailler le navigateur plus qu'il ne devrait, il cherche l'adresse IP depuis le serveur DNS à chaque fois etc... Vous devriez autant que possible garder vos images du site sur le même serveur que la page HTML.

Texte alternatif

Le prochain attribut que nous allons étudier est `alt`. Sa valeur est supposée être un descriptif sous forme de texte de l'image, à utiliser dans les cas où l'image ne peut être affichée. Exemple : le code au-dessus pourrait être modifié de cette manière :

```

```

La manière la plus simple de tester votre texte `alt` est de mal épeler votre nom de fichier intentionnellement. Si dans l'exemple, la photo était épelée `imgg.jpg`, le navigateur ne l'afficherait pas mais afficherait le texte `alt` à la place.

Largeur et hauteur (width-height)

Vous pouvez vous servir des attributs `width` et `height` pour spécifier la largeur et la hauteur de votre image. Vous pouvez trouver la largeur et la hauteur de différentes manières. Sur Mac, par exemple, vous pouvez utiliser `Cmd + I` pour afficher l'info relative au fichier image. Pour revenir à notre exemple, nous pourrions faire ceci :

```

```

Titre d'images

Comme décrit dans le chapitre [Création d'hyperliens](#), vous pouvez aussi ajouter un attribut `title` aux images, pour fournir un supplément d'information si nécessaire. Dans notre exemple, nous pourrions faire ceci :

```

```

Tableau :

```
<table border="1">
  <thead>
    <tr>
      <th>Column heading 1</th>
      <th> Column heading 2</th>
      <th> Column heading 3</th>
    </tr>
  </thead>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
    <td>9</td>
  </tr>
</table>
```