

# Apprendre HTML5

LES GUIDES  
EFFICACES

## TABLE DES MATIÈRES

Table des matières.....	1
Pourquoi apprendre HTML5.....	1
Comment créer un page web.....	2
La page de base .....	3
Les sections d'une page web.....	5
Navigation.....	5
Article.....	6
Section.....	6
Aparté .....	6
Prendre la bonne décision.....	6
Un segment de code vaut 1000 mots.....	7
Les blocs de texte .....	8
Les éléments .....	8
Exemple de blocs de texte .....	9
Les éléments textuels .....	10
Hyperlien.....	10
Modifier l'importance des mots.....	11
Modification typographique.....	11
Les citations.....	11
Extrait de code.....	12
Autres valeurs sémantiques .....	12
Les attributs.....	13
À quoi servent les attributs? .....	13
Syntaxe des attributs .....	13
Les attributs universels.....	13
Quelques exemples.....	14
Les hyperliens .....	16
De quoi est fait un hyperlien? .....	16
URL.....	16
Les hyperliens absolus et relatifs .....	16
Pourquoi utiliser des liens relatifs?.....	17
Naviguer dans la hiérarchie.....	18
Les liens internes.....	19
Images et média.....	21

Comment faire pour intégrer ces médias à vos pages web? .....	21
Insérer des images .....	21
L'élément <audio> .....	21
L'élément <source> .....	24
L'élément <video> .....	25
Les tableaux .....	28
Planifier .....	28
Les balises .....	28
Des exemples .....	29
D'autres guides .....	31

## POURQUOI APPRENDRE HTML5

HTML5 est un langage relativement simple à apprendre et qui peut vous être utile même si vous n'êtes pas un designer ou un développeur web.

Par exemple, vous faites du marketing et vous envoyez des courriels au format HTML. Si vous apprenez à programmer des pages simples avec HTML et CSS, vous n'aurez plus besoin d'utiliser les services d'un programmeur toutes les fois que vous voulez faire un changement à votre courriel.

Autre situation, il se peut que vous soyez un expert sur un sujet particulier dans votre compagnie et que vous soyez en charge de l'intranet. Bien que la plupart des systèmes de gestion de contenu offrent des interfaces WYSIWYG, une compréhension des langages qui portent votre intranet (soit à tout le moins HTML et CSS) vous donnera une longueur d'avance. Cela vous permettra d'exploiter pleinement les possibilités de votre système de gestion de contenu.

Peut-être êtes-vous intéressé par les applications mobiles? Développer un logiciel comme Word ou Excel demande beaucoup d'efforts, de ressources, des programmeurs et du temps. Mais la plupart des applications qui s'installent sur votre téléphone intelligent sont en fait des pages web relativement simples à programmer (en comparaison à un logiciel qui roule sur Windows ou Mac). Et pour ces petites applications, HTML5 est le langage de base : celui qui est le plus utilisé et qui vous permet de créer une application compatible autant avec iOS qu'Android. Ce guide ne vous permettra pas de programmer une application mobile, mais il vous permettra de franchir la première étape.

HTML5 est un langage de balisage, une fois que vous le connaissez, il vous ouvre la porte (ce sera plus facile d'apprendre) d'autres langages du même genre comme XML.

Finalement, si vous pensez à une carrière en développement web, c'est le tout premier langage à apprendre, cela va de soi.

## COMMENT CRÉER UN PAGE WEB

Ce guide contient des exemples mais ne contient pas de captures d'écran. Dans les prochaines pages, vous verrez de nombreux exemples de code et d'extrait de code.

Votre travail sera de transcrire ces extraits de code dans un éditeur de texte et de les afficher dans un navigateur.

Votre éditeur de texte n'a pas à être très sophistiqué. Par exemple, sous Windows, *Notepad* fera très bien l'affaire. Sur Mac, l'équivalent se nomme *TextEdit* et sur Linux vous pouvez utiliser quelque chose comme *pico* (dans une console) ou *gedit*.

Dans votre éditeur de texte, créez un nouveau document et tapez le texte suivant :

```
Ceci est un test. Ne paniquez pas!
```

Sauvegardez votre nouveau document sous le nom `test.html`. Assurez-vous que l'extension du fichier est bien *html* et non *txt* ou autre chose.

Démarrez votre navigateur préféré et allez dans Fichier – Ouvrir et choisissez le document que vous venez tout juste de créer.

Le texte que vous avez écrit dans votre éditeur de texte est affiché dans le navigateur.

Félicitations! Vous êtes prêt à passer à la prochaine section.

## LA PAGE DE BASE

Voici le code de départ que vous pouvez simplement copier et coller dans votre éditeur de texte pour démarrer la création de votre page web. Avec ces douze lignes, vous avez une page web compatible dans tous les navigateurs.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Titre de la page</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  <body>
    <!-- contenu de la page -->
  </body>
</html>
```

Regardons chacun des éléments du code ci-haut.

```
<!DOCTYPE html>
```

Il s'agit de la déclaration du type de document. Cette ligne dit au navigateur qu'il s'agit d'un document HTML5. Avec cette information, le navigateur saura comment traiter le code qui se trouve dans le document.

```
<html lang="fr">
</html>
```

Il s'agit de l'élément **HTML**, l'élément placé au haut de la hiérarchie d'une page web. Cet élément est fait de deux balises : la balise d'ouverture `<html>` et la balise de fermeture `</html>`. Tous les autres éléments de la page seront placés entre ces deux balises.

```
<head>
</head>
```

Vient ensuite la tête du document, une portion qui est invisible quand on regarde une page web dans un navigateur. C'est dans cette section que le navigateur trouvera beaucoup d'instructions pour préparer l'affichage de la page.

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

Cet élément **META** indique au navigateur quel jeu de caractères utiliser pour afficher le texte qui se trouve dans le document. En utilisant la valeur `utf-8`, vos pages supporteront un très grand nombre de caractères, ce qui est très utile pour créer des pages web en français.

```
<title>Titre de la page</title>
```

L'élément **TITLE** indique au navigateur le titre du document, celui-ci est toujours affiché au haut de la fenêtre ou de l'onglet du navigateur.

```
<link rel="stylesheet" href="style.css">
```

L'élément **LINK** permet de lier des documents externes à une page web. Dans ce cas-ci, nous ajoutons un fichier CSS qui permet de modifier l'apparence des éléments affichés dans une page web. Le CSS est un langage que nous verrons dans un autre guide.

```
<script src="script.js"></script>
```

L'élément **SCRIPT** permet d'importer des instructions plus complexes. JavaScript est un langage de programmation qui permet à une page web d'être interactive. Cette interaction peut être très simple mais elle peut augmenter en complexité jusqu'à créer une application.

```
<body>  
  <!-- contenu de la page -->  
</body>
```

Et finalement, l'élément **BODY** indique au navigateur que tout ce qu'il contient doit être affiché dans la page web.

Notez qu'à l'intérieur de l'élément **body**, il y a une balise particulière :

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

Cette balise sert à insérer des commentaires. Entre les tirets, vous pouvez ajouter des notes qui vous serviront d'aide-mémoire quand vous relirez votre code dans 6 mois ou encore pour vos collaborateurs qui pourraient avoir de la difficulté à comprendre pourquoi vous avez pris une décision particulière. Les commentaires ne sont pas visibles lorsque la page web est affichée dans un navigateur.

Voilà qui fait le tour des éléments d'une page web de base. Évidemment tout cela n'est que la base et il reste beaucoup à apprendre.

Dans la prochaine section, nous verrons les principaux éléments qui permettent de structurer le texte que vous mettez dans une page web.

## LES SECTIONS D'UNE PAGE WEB

Jusqu'à maintenant, nous avons vu comment créer une page web et l'afficher dans un navigateur et nous avons vu la structure d'un document HTML. Nous y avons vu l'élément **HTML** qui contient tous les éléments de la page et les deux parties principales d'un document, la tête avec l'élément **HEAD** et le corps avec l'élément **BODY**. Nous avons aussi vu les principaux éléments qui font partie de la tête d'un document et leur fonction.

Nous allons maintenant voir les principaux éléments qui découpent les sections du corps d'un document HTML.

Le tableau qui suit présente les sections principales d'un document HTML :

Tableau 1 Éléments sectionnels HTML5

SECTION	ÉLÉMENT	DESCRIPTION
NAVIGATION	<code>&lt;nav&gt;</code>	Permet d'accéder à un page ou des parties d'une page.
APARTÉ	<code>&lt;aside&gt;</code>	Pour le contenu qui n'est pas directement lié au thème principal de la page
ARTICLE	<code>&lt;article&gt;</code>	Contenu autonome portant sur le thème de la page.
SECTION	<code>&lt;section&gt;</code>	Partie d'un article qui commence par un titre.

Nous allons maintenant voir en détail la fonction de chacun de ces éléments et dans quel contexte ils doivent être utilisés.

### NAVIGATION

Le principe de base du web, c'est de créer des liens entre les pages. Pas seulement des liens vers les pages de votre propre site mais des liens vers n'importe quelle ressource qui se trouve sur le web.

La navigation est un groupe de lien avec une fonction particulière : accéder aux principales parties d'une page ou de plusieurs pages qui sont toutes regroupées sous un même thème.

Généralement, ces liens pointent tous vers le même site web.

Par exemple, une page de votre site web pourrait avoir deux sections de navigation. La première sert à accéder aux autres pages de votre site. Cette navigation affiche des liens vers votre page d'accueil, votre blog, votre page à propos et votre page portfolio.

La seconde section de navigation se trouve sur une page spécifique qui est un article de 10000 mots sur un sujet précis. Cet article est divisé en 4 sections et chaque élément de cette navigation amène le visiteur directement au début d'une des sections.

Dans cette même page, vous avez un pied de page et celui-ci contient des dizaines de liens vers des pages spécifiques de votre site. Dans ce cas-ci, il ne s'agit pas d'une navigation à proprement parler et ce ne serait pas cohérent d'utiliser une section de navigation pour regrouper ces liens. C'est l'élément **FOOTER** qui serait utile mais nous verrons celui-ci plus loin dans ce guide.



## ARTICLE

L'élément **ARTICLE** contient une partie du document qui peut se distribuer indépendamment de son contexte. Le cas le plus fréquent est le contenu d'une page mais cela pourrait aussi être un outil qui converti les kilogrammes en livres. Pour savoir si ce contenu devrait être à l'intérieur d'une balise `<article>`, posez-vous la question :

*Est-ce que je pourrais consulter ce contenu sur une autre page sans ce qui se trouve autour de celui-ci?*

Si c'est le cas, il s'agit d'un article. Des exemples fréquents de contenu qui sont considérés comme des articles :

- publication dans un forum
- article de magazine ou de journal
- un billet de blog
- un commentaire
- un module interactif

## SECTION

L'élément **SECTION** est celui qu'il faut utiliser si le thème de son contenu devrait apparaître dans une table des matières (par exemple).

Le plus souvent, pour le cas d'articles de blog ou des nouvelles, l'élément **SECTION** devrait contenir un titre et du contenu.

Avant de décider d'utiliser l'élément section, il faut vérifier d'abord s'il s'agit d'un article (voir plus haut). Si l'élément **ARTICLE** ne s'applique pas et que cette partie du document a son propre titre, vous pouvez utiliser l'élément section.

## APARTÉ

L'aparté est une partie du texte qui est lié au contexte mais qui n'en fait pas nécessairement partie. Sur le web, l'aparté s'utilise dans deux contextes :

1. à l'intérieur d'un article
2. à l'intérieur du site web

Dans le premier cas, il s'agira d'un glossaire, d'une citation ou d'autre chose qui est en lien direct avec l'article mais qui n'enlèverait rien à l'article lui-même si cet aparté était retiré.

Pour le deuxième cas, l'exemple le plus fréquent est la barra latérale qui peut contenir des liens vers d'autres articles du même site web, la navigation du site ou même de la publicité (à condition qu'elle soit en lien avec le thème du site).

## PRENDRE LA BONNE DÉCISION

L'avantage et l'inconvénient de la spécification HTML5 est que la définition de chaque élément n'est pas complètement fermée. Cela vous permet d'un côté de faire un usage créatif des éléments qui sont disponibles et rend la spécification compatible avec des ajouts futurs. D'un autre côté, choisir l'élément

approprié dans un contexte particulier est parfois difficile. Un outil qui m'aide à prendre ces décisions est [l'organigramme de html5 doctor](#).

Bien qu'il soit en anglais, il permet de poser une série de questions qui vous guideront dans le choix l'élément qui est le plus approprié à votre situation.

## UN SEGMENT DE CODE VAUT 1000 MOTS

Voici un exemple de code avec les éléments que nous venons de voir.

```
<body>
  <aside>
    <nav>
      <!-- liste de liens vers les pages principales du site ->
    </nav>
  </aside>
  <article>
    <!-- contenu pour la page, en lien avec l'élément title ->
    <section>
      <!-- première section ->
    </section>
    <section>
      <!-- deuxième section ->
    </section>
    <section>
      <!-- ènième section ->
      <aside>
        <!-- citation tirée de l'article ->
      </aside>
    </section>
  </article>
  <aside>
    <!-- liste de liens vers des articles semblables ->
    <!-- publicité ->
  </aside>
</body>
```

## LES BLOCS DE TEXTE

Depuis le début de ce guide, nous avons vu la structure de base d'un document HTML5. Nous avons ensuite vu les éléments qui permettent de créer des sections à l'intérieur du corps d'une page web.

Dans cette section, nous verrons une liste partielle des éléments de niveau *bloc de texte* qui ont une valeur sémantique.

Qu'est-ce qu'une valeur sémantique? Pour simplifier, disons que chaque partie d'un texte a une fonction. Par exemple, un paragraphe regroupe des phrases qui expriment une idée. Dans un texte bien structuré, un paragraphe contient une idée.

Le même concept s'applique aux listes. Bien qu'il soit possible d'énumérer des items à même un phrase, la liste permet de distinguer les différents items d'une énumération et parfois même, dans le cas de la liste numérotée, de les ordonner.

Suivant cette idée, nous verrons ici les éléments les plus fréquemment utilisés pour baliser les blocs de texte dans des pages HTML5.

À la fin de cette section, nous verrons un exemple concret.

## LES ÉLÉMENTS

### PARAGRAPHE

```
<p>
```

Cet élément contient un paragraphe de texte. Tous les paragraphes d'un texte sont séparés par un espace vertical appelé alinéa. Parfois, le premier mot de la première ligne est en retrait pour renforcer la séparation des paragraphes.

### LIGNE HORIZONTALE

```
<hr>
```

Dans les anciennes versions d'HTML, l'élément `HR` plaçait une ligne horizontale dans la page. Dans HTML5, bien que les navigateurs puissent afficher une ligne horizontale, l'élément `HR` a une fonction particulière, il représente un changement de thème entre 2 paragraphes.

## LES LISTES

Dans HTML5, il existe trois types de liste. Certaines peuvent prendre des apparences différentes grâce au CSS. Dans ce guide, nous nous limiterons à l'affichage le plus fréquemment utilisé.

- `<o1>` : liste numérotée
- `<u1>` : liste à puce
- `<d1>` : liste de définition

## LE CONTENU DES LISTES

Chaque liste contient une série d'items. Selon le type de liste, un élément servira à baliser ces items.

- `<li>` : pour les listes numérotées et les liste à puce
- `<dt>` : pour un terme dans une liste de définitions
- `<dd>` : pour une définition dans une liste définitions.

## EXEMPLE DE BLOCS DE TEXTE

Disons que nous avons un court texte à propos des fruits et des légumes et que nous voulons le mettre en HTML5. Voici une des solutions possible.

```
<p>Il existe deux sous-catégories des produits de culture végétale :</p>
<ol>
  <li>Les fruits</li>
  <li>et les légumes.</li>
</ol>
<p>À titre d'exemple, les végétaux suivants font partie des fruits :</p>
<ul>
  <li>les pommes,</li>
  <li>les bananes</li>
  <li>et les oranges.</li>
</ul>
<p>Dans la catégorie des légumes, nous retrouverons les végétaux suivants
:</p>
<ul>
  <li>les patates douces,</li>
  <li>les asperges</li>
  <li>et le brocoli.</li>
</ul>
<hr>
<p>D'une manière générale, nous pouvons, dans le langage courant, définir les
fruits et les légumes comme suit :</p>
<dl>
  <dt>Fruit :</dt><dd> un aliment végétal, à la saveur sucrée, généralement
consommé cru au dessert</dd>
  <dt>Légume :</dt><dd> partie comestible d'une plante potagère</dd>
</dl>
<p>Source : fr.wikipedia.org</p>
```

Notez bien qu'il s'agit d'une solution parmi tant d'autres. Par exemple, l'usage de liste à puce n'est pas nécessaire; il s'agit d'un choix.

Maintenant que nous avons vu la structure d'un document HTML, les différentes sections du corps d'une page et les blocs de texte les plus importants.

Nous verrons les éléments qui s'appliquent à des mots ou des groupes de mots.

## LES ÉLÉMENTS TEXTUELS

Les éléments sémantiques de niveau textuel sont les éléments qui donnent une fonction particulière à un mot ou des groupes de mots à l'intérieur d'une phrase ou à l'intérieur d'un bloc de texte.

Tableau 2 Principaux éléments textuels

ÉLÉMENT	ANGLAIS	FRANÇAIS
<code>&lt;a&gt;</code>	anchor	hyperliens
<code>&lt;em&gt;</code>	emphasis	emphase
<code>&lt;strong&gt;</code>	strong	importance
<code>&lt;small&gt;</code>	small	petits caractères
<code>&lt;cite&gt;</code>	citation	titre
<code>&lt;q&gt;</code>	citation	citation courte
<code>&lt;code&gt;</code>	code	fragment de code
<code>&lt;sub&gt;</code>	subscript	indice
<code>&lt;sup&gt;</code>	superscript	exposant
<code>&lt;i&gt;</code>	italic	italique
<code>&lt;b&gt;</code>	bold	gras
<code>&lt;u&gt;</code>	underline	sous-ligné
<code>&lt;span&gt;</code>	span	autres
<code>&lt;br&gt;</code>	line-break	retour de ligne

## HYPERLIEN

On utilise l'élément `A` pour créer des hyperliens qui mènent à d'autres emplacements dans la même page ou encore vers d'autres ressources sur internet.

Pour être un hyperlien, la balise `<a>` doit avoir à tout le moins un attribut: `href`. Par exemple :

```
<a href="http://alainericgauthier.com">Alain Éric Gauthier</a>.
```

Nous verrons plus loin ce que sont les attributs et à quoi ils servent.

La syntaxe se construit comme suit :

La balise de fermeture de cet élément est obligatoire.

```
<a></a>
```

Le texte qui se trouve entre les deux balises sera affiché sur la page mais aura une présentation différente. Par défaut, la plupart des navigateurs affichent ce texte en bleu et souligné.

```
<a>Texte qui sera en hyperlien</a>
```

L'attribut `href` indique au navigateur la cible de l'hyperlien, c'est à dire vers quelle ressource il pointe. C'est habituellement une autre page web mais ça peut aussi être différents types de fichiers (audio, vidéo) ou encore d'autres types de ressources sur internet (comme un site ftp, etc.).

```
<a href="">Texte qui sera en hyperlien</a>
```

Finalement, il nous reste à ajouter l'URL de la cible.

```
<a href="http://alainericgauthier.com">Texte qui sera en hyperline</a>
```

*Note: Il est possible d'utiliser l'élément a comme ancre dans un document mais nous verrons cela dans une autre section.*

## MODIFIER L'IMPORTANCE DES MOTS

Dans un texte, il est parfois nécessaire de mettre l'emphase sur un ou plusieurs mots d'une phrase ou encore d'en réduire l'importance. Les éléments suivants s'utilisent pour moduler l'importance de mots qu'ils balisent. Il ne faut pas confondre cet usage avec les conventions typographiques (voir Modification typographique).

- **<em>** : pour mettre l'emphase sur une partie du texte
- **<strong>** : pour montrer une partie de texte qui est importante
- **<small>** : pour mettre des mots en petits caractères, comme une clause qu'on ne veut pas que vous lisiez dans un contrat.

## MODIFICATION TYPOGRAPHIQUE

Dans certains contextes, il est parfois nécessaire de respecter des conventions typographiques. Par exemple, dans un manuel d'utilisation, les mots techniques pourraient être en caractère gras et les mots dans une autre langue pourraient être en italique.

Les éléments qui suivent modifient l'affichage du texte mais ne donne pas une fonction particulière à ces mots dans le texte.

- **<sub>**, **<sup>** : pour les indices et les exposants. Évidemment, on s'en sert beaucoup en mathématique mais on peut aussi s'en servir pour les notes de bas de page.
- **<i>** : pour un bout de texte qui se distingue du texte, par exemple des termes techniques ou dans une autre langue ou encore les pensées d'un personnage dans une œuvre de fiction.
- **<b>** : lorsque l'auteur veut utiliser un affichage différent pour certains mots sans que ceux-ci aient une valeur sémantique particulière.
- **<u>** : lorsque l'auteur veut souligner un bout de texte pour suggérer une annotation mais que celle-ci n'est pas spécifiée.
- **<br>** : ajoute un saut de ligne dans le texte, ce qui est différent d'un saut de paragraphe. Utile quand on veut mettre en évidence une adresse ou une équation à l'intérieur d'un paragraphe ou pour séparer les vers un poème.

## LES CITATIONS

Pour les citations assez longues pour créer un paragraphe distinct, vous utiliserez l'élément **BLOCKQUOTE**. Mais si votre citation ne fait que quelques mots et que vous voulez l'insérer à l'intérieur d'une phrase plus longue, vous avez deux éléments à votre disposition :

- **<cite>** : permet de baliser le titre d'une œuvre.

- **<q>** : à utiliser pour les citations courtes qui sont en incise, c'est à dire à l'intérieur d'une phrase.

#### EXTRAIT DE CODE

**<code>** : pour inclure un bout de texte qui est un langage de programmation, n'est pas à utiliser au niveau bloc de texte par contre.

#### AUTRES VALEURS SÉMANTIQUES

Il n'est pas possible de prévoir toutes les situations et c'est pourquoi HTML5 offre la possibilité de créer des conventions propres à une page ou un site web particulier.

**<span>** : c'est l'élément à utiliser si aucun autre élément de niveau textuel n'est pertinent. Par exemple, si certaines phrases d'un paragraphe sont des avertissements, elles peuvent être balisées par un élément **<span>** qui aura un attribut **class="avertissement"**.

Comme nous avons déjà vu à quelques reprises des attributs, nous allons maintenant voir plus précisément ce qu'ils sont et à quoi ils servent.

## LES ATTRIBUTS

Un attribut permet de modifier le comportement d'un élément ou d'y ajouter de l'information. La valeur de l'attribut, ce qui se trouve après le signe d'égalité, sera utilisée différemment selon le type d'attribut. Par exemple, l'attribut `title` permet d'afficher des informations supplémentaires à propos d'un élément.

Disons que vous créez un hyperlien vers un autre site web. Vous pouvez, avec l'attribut `title`, indiquer le nom du site web ou le titre de la page mais vous pouvez aussi y ajouter le nom de l'auteur ou d'autres informations pertinentes liées à ce lien. Quand l'utilisateur place son curseur au-dessus du lien, le texte assigné à l'attribut `title` apparaîtra dans une bulle près du curseur.

La syntaxe d'un élément contenant un seul attribut est la suivante :

```
<[élément] [attribut]="[valeur]">Texte dans la page</[élément]>
```

Pour vous donner un exemple concret semblable à celui décrit plus haut :

```
<a href="http://alainericgauthier.com" title="Site web d'Alain Éric Gauthier">Visitez mon site web</a>
```

## À QUOI SERVENT LES ATTRIBUTS?

Un attribut donne une signification et un contexte à un élément. On s'en sert pour distinguer des éléments identiques qui ont une fonction différente dans un contexte particulier.

Par exemple, vous voulez publier un texte informatif qui comporte deux types de paragraphes, débutants et avancés. Il est possible d'utiliser l'attribut `class` et de définir deux classes : "debutants", "avances".

```
<p class="debutants">Paragraphe pour les utilisateurs débutants.</p>
<p class="avances">Paragraphe pour les utilisateurs avancés.</p>
```

## SYNTAXE DES ATTRIBUTS

Les attributs se présentent sous 4 formes :

1. [attribut]
2. [attribut]=valeur
3. [attribut]='valeur'
4. [attribut]="valeur<sub>1</sub> valeur<sub>2</sub> ... valeur<sub>n</sub>"

Un attribut peut avoir plusieurs valeurs et dans ce cas il faut utiliser des guillemets simples ou doubles et chaque valeur sera séparée d'un espace. L'ordre des attributs n'est pas important.

## LES ATTRIBUTS UNIVERSELS

Les attributs universels s'appliquent à tous les éléments HTML5. Tous les éléments de la norme HTML5 les acceptent et tous les navigateurs savent interpréter ces attributs.

Il existe 22 attributs universels ([source](#)).

Voici la liste des attributs qui sont vraiment utiles pour créer une page web contenant principalement du texte :



**class** : définit une classe ou plusieurs classes pour un élément, la même classe peut être utilisée sur plusieurs éléments identiques ou différents dans une même page.

**id** : définit un identifiant, il ne peut y avoir qu'un seul identifiant par page.

**lang** ou **xml:lang** : définit la langue du texte délimité par l'élément.

**style** : permet d'appliquer du CSS (que nous verrons dans une autre série d'articles) directement à l'élément.

**title** : permet de donner un titre à un élément, cette information sera utilisée, par exemple, par les outils qui permettent de présenter vos pages web à des personnes avec un handicap visuel.

Les attributs ne sont pas obligatoires mais ils sont très utiles pour spécifier des particularités à des éléments.

## QUELQUES EXEMPLES

Vous créez un site web pour un manuel d'instruction qui contient des paragraphes qui mettent en garde l'utilisateur, il s'agit de paragraphes d'avertissement.

```
<p class="avertissement">Attention. Voici un paragraphe d'avertissement.</p>
```

Bien que la syntaxe HTML5 repose sur l'anglais, la langue utilisée pour la valeur des attributs importe peu :

```
id="accueil"
```

Vous pouvez définir la langue d'une page en ajoutant un attribut à l'élément **HTML** ou encore pour une citation dans une autre langue avec l'élément **Q**.

```
<html xml:lang="en">
<q lang="fr">
```

L'attribut **style** permet d'ajouter du CSS (un langage complémentaire à HTML5 qui permet de contrôler la présentation de vos pages web) à un élément :

```
style="text-align:center"
```

Contrairement à ce que ce mot suggère, l'attribut **title** ne sert pas uniquement à ajouter un titre mais il peut contenir des informations à propos du contenu de l'élément. Par exemple, dans le cas d'une citation, il serait possible d'indiquer le nom de l'auteur, l'œuvre liée et le nom du site web où l'information a été trouvée. Pour les navigateurs sur ordinateur, ce texte s'affiche généralement sous-forme de bulle superposée quand le curseur est au-dessus du contenu de l'élément.

Mettons en application ce que nous avons vu jusqu'à maintenant.

```
<section id="citations">
  <h2>Quelques unes de mes citation préférées</h2>
  <p class="avertissement">
    <strong>Avertissement :</strong><br />
```

Les opinions exprimées sur cette page ne sont endossées que par l'auteur. Son employeur et ses proches ne peuvent être tenus responsables des opinions exprimées sur cette page.<p>

<p>À propos de la guerre : <q title="Citation d'Albert Einstein dans Comment je vois le monde">"Ceux qui aiment marcher en rangs sur une musique : ce ne peut être que par erreur qu'ils ont reçu un cerveau, une moelle épinière leur suffirait amplement."</q></p>

<p>À propos du savoir, Peter Drucker à dit : <q lang="en">Today knowledge has power. It control access to opportunities and advancement.</q></p>

<p>À propos de la politique : <q title="Citation de Coluche">La différence qu'il y a entre les oiseaux et les hommes politiques, c'est que de temps en temps les oiseaux s'arrêtent de voler!</q></p>

</section>

Et comme je l'ai déjà mentionné pour d'autres exemples, les choix que j'ai faits ici ne sont qu'une solution parmi tant d'autres. La beauté d'HTML, c'est que nous pouvons l'utiliser de différentes manières pour rendre la communication plus efficace.

## LES HYPERLIENS

Selon un [article de 2013](#), n'importe quelle page web serait au plus à 19 clics de n'importe quelle autre page.

Les résultats de cette recherche ont été [présentés sur le web assez souvent](#) et [ici](#) mais [l'article original se trouve ici](#).

Petite note, vous n'avez pas à payer pour lire toute l'étude, les conclusions principales sont dans *l'abstract*. Mais peu importe si cela est encore vrai aujourd'hui, l'idée est que l'intérêt du web est d'avoir un très grand nombre de pages liées les unes aux autres et que ces liens soient pertinents.

Sur le site du CERN, vous pouvez lire un [article sur la naissance du web](#); vous verrez que les hyperliens sont au cœur de ce qu'est HTML.

Mais trêve de bavardage sur la théorie des hyperliens et l'histoire du web, vous êtes ici pour faire des pages web, n'est-ce pas?

Commençons par des exemples concrets. Dans cette section, vous trouverez des liens vers des pages qui m'ont aidé à rédiger cet article. Vers la fin de l'article, j'ajoute, à titre d'exemple, le code de certains de ces hyperliens.

### DE QUOI EST FAIT UN HYPERLIEN?

Nous avons déjà vu comment est fait un hyperlien. Pour résumer, voici un hyperlien avec le minimum d'information pour qu'il pointe vers une autre page web.

```
<a href="http://info.cern.ch/hypertext/WWW/TheProject.html">Le premier site du web</a>
```

Vous aurez compris qu'un bout de texte est balisé par l'élément `A` et que celui-ci possède un attribut `href` et que la valeur de cet attribut est l'adresse du premier site du web. Une telle adresse se nomme URL.

### URL

Et pour parler des hyperliens, il faut parler des URL.

- Uniform
- Resource
- Locator

Si vous n'en avez pas assez de la théorie et que vous voulez creuser un peu plus ce qu'est un URL, je ne peux que vous recommander [la page wikipedia](#) pour commencer et si vous voulez y aller vraiment à fond, lisez le [Living Standard](#). Si jamais vous en revenez sain d'esprit, je continue avec mes exemples.

### LES HYPERLIENS ABSOLUS ET RELATIFS

Jusqu'à maintenant, je ne vous ai donné que des exemples d'hyperliens absolus. Ces liens ont la syntaxe suivante :

```
protocole://domaine/chemin
```

Le protocole indique au navigateur comment l'information sera transférée du serveur au client. Les protocoles les plus fréquents sur le web sont https et http. Le domaine c'est le nom du site web. Par exemple, le domaine de mon site web est `alainericgauthier.com`. Le chemin (*path* en anglais), décrit l'emplacement du document.

*Note* : selon le type de serveur qui sert les pages web au client (un navigateur), le document peut contenir une extension (le plus souvent .html mais parfois .htm) ou rien du tout. Mais ceci est un détail technique qui dépasse le cadre de cette série.

Les hyperliens relatifs permettent de naviguer vers un autre page du même site web à partir de la page actuelle. Pour l'utilisateur, cela ne fait pas vraiment de différence : cliquer sur un lien absolu ou un lien relatif mènera vers une autre page. Mais pour le navigateur (et le programmeur), le processus est différent.

Quand vous créez un site web, vos documents HTML peuvent être dans le même dossier, dans un sous-dossier du dossier actuel, ou dans un dossier au-dessus du dossier actuel. Les liens relatifs vous permettent d'accéder à ces trois emplacements.

Des exemples vous aideront à mieux comprendre ce que je veux dire.

Disons que sur le domaine `exemple.com`, vous avez deux documents html :

```
index.html
about.html
```

Vous êtes dans le document `index.html` et que vous voulez accéder au document `about.html`. En lien absolus, voici comment accéder à chacun des documents :

Tableau 3 Liens absolu et relatif sans hiérarchie

DOCUMENT	LIEN ABSOLU	LIEN RELATIF
<b>INDEX.HTML</b>	<a href="http://exemple.com/index.html">http://exemple.com/index.html</a>	<a href="#">index.html</a>
<b>ABOUT.HTML</b>	<a href="http://exemple.com/about.html">http://exemple.com/about.html</a>	<a href="#">about.html</a>

Ce qui veut dire que pour créer un lien vers `about.html` dans `index.html`, vous pouvez écrire le code suivant :

```
<a href="about.html">À propos</a>.
```

Notez que le code suivant fonctionnerait aussi :

```
<a href="http://exemple.com/about.html">À propos</a>
```

## POURQUOI UTILISER DES LIENS RELATIFS?

Puisque les deux types de liens permettent de faire exactement la même chose pourquoi devrait-on utiliser des liens relatifs?

La raison principale est que votre site pourrait déménager. Par exemple, si vous travaillez sur un serveur local, il se pourrait que celui-ci ait un nom de domaine différent. Il se pourrait aussi qu'un jour, vous décidiez de migrer votre site vers un nouveau domaine. Par exemple, vous pourriez passer de **exemple.com** à **autre.com**.

Si vous avez des liens absolus, vous devrez tous les réécrire dans chacun de vos documents. Si vous n'avez que deux documents, cela n'est pas très long. Mais un site web respectable aura d'une vingtaine à plusieurs centaines de pages. Vous n'avez pas envie de vous taper la réécriture de milliers de liens et c'est la beauté des liens relatifs, tant que les documents HTML restent au même endroit dans l'architecture des dossiers, vous n'avez rien à modifier.

## NAVIGUER DANS LA HIÉRARCHIE

Parlant d'architecture, le lien relatif que vous avez vu ne vous permet que d'accéder aux documents qui sont dans le même dossier que celui dans lequel vous êtes présentement.

Disons que vous avez un document **index.html** et que vous avez un dossier **produits** qui contient un document nommé **widgets.html**. Et disons que vous êtes encore sur le domaine **exemple.com**. Le tableau suivant vous montre le lien absolu et le lien relatif.

Tableau 4 Liens absolus et relatifs pour différents niveaux dans la hiérarchie

DOCUMENT	LIEN ABSOLU	LIEN RELATIF
<b>INDEX.HTML</b>	<a href="http://exemple.com/index.html">http://exemple.com/index.html</a>	<a href="#">index.html</a>
<b>WIDGETS.HTML</b>	<a href="http://exemple.com/produits/widgets.html">http://exemple.com/produits/widgets.html</a>	<a href="#">/produits/widgets.html</a>

Remarquez que le lien relatif n'est composé que de la partie chemin d'un URL comme je l'ai montré au début de cet article.

Disons que l'architecture de votre site web se complique un peu et que vous avez maintenant les mêmes documents que dans l'exemple précédent mais vous avez aussi un deuxième dossier nommé **contact**.

Dans le dossier racine de votre serveur (le dossier le plus haut dans la hiérarchie) vous aurez donc deux documents HTML (**index.html** et **about.html**) et vous aurez deux dossiers (**produits** et **contact**).

Disons que vous êtes dans le document **widgets.html** qui se trouve dans le dossier **produits** et que vous voulez créer un hyperlien vers le document **president.html** qui se trouve dans le dossier **contact**. Et disons que vous voulez aussi créer un lien vers la page **widgets.html** quand vous êtes dans la page **president.html**. Le tableau suivant vous montre le lien absolu et le lien relatif menant vers l'autre document.

Tableau 5 Liens absolus et relatif dans une hiérarchie complexe

À PARTIR DE ...	LIEN ABSOLU	LIEN RELATIF
<b>WIDGETS.HTML</b>	<a href="http://exemple.com/contact/president.html">http://exemple.com/contact/president.html</a>	<a href="#">../contact/president.html</a>
<b>PRESIDENT.HTML</b>	<a href="http://exemple.com/produits/widgets.html">http://exemple.com/produits/widgets.html</a>	<a href="#">../produits/widgets.html</a>

C'est les deux points qui indiquent au navigateur qu'il faut remonter d'un niveau dans la hiérarchie de dossiers.

Mais ça ne s'arrête pas là, si vous voulez remonter de deux dossiers dans la hiérarchie, vous pouvez utiliser les deux points deux fois.

```
../../index.html
```

## LES LIENS INTERNES

L'élément `A` nous permet de faire une autre sorte de lien, les liens internes au document. Ceux ne sont pas utilisés très souvent mais si vous avez des documents vraiment longs, ils peuvent devenir utiles. Les liens dans une page *faire aux questions* sont un exemple courant sur le web.

Pour créer un lien interne, il faut deux choses :

1. Un hyperlien avec une valeur pour l'attribut `href` qui commence par un dièse (`#`) et
2. connaître la valeur d'un attribut `id` d'un élément vers lequel nous voulons pointer.

Disons que vous avez un document qui contient au haut de la page une navigation interne et trois sections qui commencent toutes par un titre de niveau 2 : `<h2>`.

Comme vous l'avez vu dans l'article sur les attributs universels, tous les éléments peuvent avoir un attribut `id`.

Comme un bout de code vaut parfois mille mots, voici donc le code avec des liens internes qui ne fonctionnent pas encore.

```
<nav>
  <ul>
    <li>Première section</li>
    <li>Deuxième section</li>
    <li>Troisième section</li>
  </ul>
</nav>
<h2>Première section</h2>
<p>Imaginez que ceci est un très loooooong paragraphe.</p>
<h2>Deuxième section</h2>
<p>Imaginez que ceci est un très loooooong paragraphe.</p>
<h2>Troisième section</h2>
<p>Imaginez que ceci est un très loooooong paragraphe.</p>
```

La première chose à faire est d'ajouter la balise `<a>` à l'intérieur des balises `<li>`.

```
<li><a href="">Première section</a></li>
<li><a href="">Deuxième section</a></li>
<li><a href="">Troisième section</a></li>
```

Nous allons ensuite ajouter des attributs `id` à chaque balise `<h2>` et leur donner une valeur différente.

```
<h2 id="un">Première section</h2>
...
<h2 id="deux">Deuxième section</h2>
'''
<h2 id="trois">Troisième section</h2>
```

Il ne reste plus qu'à ajouter une valeur pour chaque attribut `href`. Voici le code final de la page :

```
<nav>
  <ul>
    <li><a href="#un">Première section</a></li>
    <li><a href="#deux">Deuxième section</a></li>
    <li><a href="#trois">Troisième section</a></li>
  </ul>
</nav>
<h2 id="un">Première section</h2>
<p>Imaginez que ceci est un très loooooong paragraphe.</p>
<h2 id="deux">Deuxième section</h2>
<p>Imaginez que ceci est un très loooooong paragraphe.</p>
<h2 id="trois">Troisième section</h2>
<p>Imaginez que ceci est un très loooooong paragraphe.</p>
```

Avec ce que vous avez vu jusqu'à maintenant, vous êtes en mesure de créer 80% des pages qui existent sur le web. Mais il reste encore quelques éléments importants à voir.

## IMAGES ET MÉDIA

Vous le savez, vous le voyez à tous les jours, une page web ne contient pas seulement du texte. À moins que vous n'ayez passé les 10 dernières années dans une grotte sur une île déserte, vous avez sûrement remarqué que certains sites web sont en fait des pages remplies d'images et de vidéo. Et les podcasts et la musique occupent aussi une place non négligeable sur le web.

### COMMENT FAIRE POUR INTÉGRER CES MÉDIAS À VOS PAGES WEB?

D'un certain point de vue, il n'y a pas de différences entre des documents HTML et des fichiers médias. Tout comme pour les documents HTML, les fichiers images, audio et vidéo doivent être entreposés sur un serveur.

Et comme c'est le cas pour les liens entre les pages, les URL relatifs sont très importants pour les fichiers multimédias.

### INSÉRER DES IMAGES

Vous le savez sûrement, vous pouvez insérer des images dans votre texte. Pour ce faire, vous utiliserez l'élément `IMG`. Cet élément a une particularité que nous n'avons pas encore vue, c'est un élément vide. Ce qui veut simplement dire qu'il ne peut pas baliser du texte et que la façon correcte de le mettre en HTML est de mettre la barre oblique de fermeture à la fin de la balise.

```
<img />
```

Évidemment, cet élément accepte tous les attributs universels et il accepte aussi des attributs qui lui sont spécifiques.

#### L'ATTRIBUT ALT

Le contenu de cet attribut sera utilisé pour afficher une description de l'image si jamais celle-ci ne pouvait être affichée dans la page. Une image ne s'affichera pas dans une page s'il y a un problème technique, si le navigateur ne supporte pas un format d'image, si votre URL est brisée ou encore si le visiteur a désactivé l'affichage d'images dans son propre navigateur. Mais encore plus important, il existe des navigateurs spécialisés pour les personnes ayant un handicap visuel et pour ces personnes, le contenu de l'attribut `alt` est très important. Ces navigateurs liront à l'utilisateur le contenu de cet attribut pour que le visiteur sache ce que contient l'image. Selon la norme HTML5, cet attribut doit être présent.

#### L'ATTRIBUT SRC

Le contenu de cet attribut est un URL absolu ou relatif et il pointe vers un fichier image (GIF, JPG, PNG). Si cet URL n'est pas exact, l'image ne sera pas affichée. Selon la norme HTML5, cet attribut doit être présent. À quoi bon ajouter un élément image à un document HTML si ce n'est pas pour ajouter une image à la page?

#### L'ÉLÉMENT <AUDIO>

Il n'y a pas si longtemps, pour intégrer un fichier audio à une page web, il fallait créer un objet en FLASH ou encore utiliser de la programmation en JS et espérer que le visiteur aurait un lecteur audio compatible sur son ordinateur. Bien que la situation actuelle ne soit pas parfaite, il est beaucoup plus facile d'intégrer



de l'audio sur une page web. Il faut simplement être conscient de quelques particularités au niveau de la compatibilité des fichiers audio.

De plus, il ne faut pas oublier que la navigation sur le web, c'est encore une activité silencieuse et c'est une bonne chose. Cela vous est sûrement arrivé, la dernière fois que vous avez utilisé votre ordinateur le volume était assez fort et vous avez oublié de le réduire. Vous arrivez sur une page web et un fichier audio démarre automatiquement. Cette situation est déjà désagréable mais imaginez que vous êtes dans une bibliothèque au moment où ça vous arrive...

La philosophie adéquate avec les médias est de laisser le contrôle à l'utilisateur.

Contrairement à l'élément `IMG`, l'élément `AUDIO` accepte du texte entre ces balises d'ouverture et de fermeture. Dans la plupart des cas, le texte qu'on y ajoute est un message expliquant que ce navigateur n'est pas compatible avec l'élément `AUDIO`.

La plupart des programmeurs pensent en programmeur et vont mettre un message du genre « *Votre navigateur ne supporte pas l'élément audio.* » Mais pour une personne qui ne connaît rien au développement web, ce message ne donne pas beaucoup d'information. Il serait plus utile d'offrir une solution alternative.

Par exemple,

```
Votre navigateur n'est pas en mesure de jouer ce fichier audio à même la page
mais vous pouvez le télécharger à cette adresse :
<a href="http://exemple.com/audio.wav">http://exemple.com/audio.wav</a>.
```

---

## L'INTÉGRATION LA PLUS SIMPLE

Pour insérer un fichier audio dans votre page web, il vous faut à tout le moins le code suivant pour que ce soit utile à vos visiteurs :

```
<audio src="http://exemple.com/audio.wav">
  Votre navigateur est un incapable car il ne peut pas jouer ce fichier audio
  à même la page. Vous n'aurez d'autre choix que de le télécharger à cette
  adresse :
  <a href="http://exemple.com/audio.wav" title="Un fichier
  audio">http://exemple.com/audio.wav.</a>
</audio>
```

---

## LES ATTRIBUTS

L'élément `AUDIO` est supporté par les navigateurs depuis assez longtemps. C'est Internet Explorer qui a ajouté le support en dernier avec la version 9. Donc, si vous devez créer des pages web compatibles avec une version inférieure à IE 9, vous devrez absolument mettre en place une alternative.

Il existe 4 attributs principaux pour l'élément `AUDIO`. De plus, leur support par les navigateurs est identique à l'élément lui-même. Ces quatre attributs vous permettront passablement de contrôle sur le comportement des lecteurs audio dans vos pages web.

---

### SRC

Nous verrons plus loin qu'il y a une autre façon de lier un fichier audio à un lecteur mais cet attribut se comporte exactement de la même manière que pour l'élément **IMG**.

Les trois attributs suivant sont des attributs booléens. En gros, les attributs booléens indiquent si un état est vrai ou faux. Comme les programmeurs aiment bien prendre des raccourcis, les attributs booléens pour les éléments HTML5 se comportent comme suit. Si l'attribut est présent, c'est que son état est vrai (activé) et si l'élément est absent, c'est que cet état est faux (désactivé).

Il est donc inutile de mettre quelque chose comme `autoplay="false"`.

---

## AUTOPLAY

Si cet attribut est présent dans la balise d'ouverture de l'élément **AUDIO**, le fichier audio se mettra à jouer dès que le navigateur sera prêt. Comme mentionné plus haut, assurez-vous que pour le visiteur, accéder à une page qui contient un fichier audio qui démarrera automatiquement ne sera pas une mauvaise surprise.

```
<audio autoplay src="test.wav">
Erreur: quelque chose de terrible s'est produit.
</audio>
```

---

## CONTROLS

Évidemment, quand on voit un lecteur audio sur le web, on s'attend à avoir un minimum de contrôle sur celui-ci. L'attribut **controls**, lorsque présent, donnera au minimum à l'utilisateur les contrôles suivants:

- Jouer
- Pause
- Volume

Voici un exemple de l'intégration de l'attribut **controls**.

```
<audio controls src="panique.wav">
  Il y a un problème avec votre navigateur, vous pouvez maintenant paniquer.
</audio>
```

---

## LOOP

Pour un raison ou pour une autre, vous pourriez avoir une irrésistible envie de faire jouer un fichier en boucle. Pour ce faire, la présence de l'attribut **loop** permet justement de faire jouer ce fichier en boucle.

```
<audio loop src="boucle.wav">
  Savez-vous lacer vos souliers? Votre navigateur ne sait pas faire des
  boucles.
</audio>
```

---

## COMPATIBILITÉ DES FORMATS AUDIO

Finalement, comme si savoir tout cela n'était pas suffisant, tous les navigateurs ne supportent qu'un seul format de fichier audio, le format MP3. Il existe 2 autres formats principaux: ogg et wav. Mais ces formats ne sont pas universels.

Si vous voulez tout de même supporter ces formats, vous aurez donc à offrir plus d'options à vos visiteurs à l'aide d'un autre élément : `<source>`.

Tableau 6 Compatibilité des formats audio

NAVIGATEUR	MP3	WAV	OGG
INTERNET EXPLORER	Oui	Non	Non
CHROME	Oui	Oui	Oui
FIREFOX	Oui	Oui	Oui
SAFARI	Oui	Oui	Non
OPERA	Oui	Oui	Oui

## L'ÉLÉMENT `<SOURCE>`

Cet élément est utilisé à l'intérieur des balises `<audio>` ou `<video>`.

Il offre différentes options aux navigateurs qui sont programmés pour faire le meilleur choix en fonction de la situation et des contraintes de comptabilités.

## LES ATTRIBUTS

Cet élément utilise deux attributs principaux `src` et `type`.

### SRC

Tout comme pour les images et l'élément `AUDIO` lui-même, cet attribut indique un URL pointant vers un fichier audio. Dans l'élément `SOURCE`, cet attribut est obligatoire.

### TYPE

Cet attribut n'est pas obligatoire mais si vous le spécifiez, le chargement de votre page sera un peu plus rapide. Tout ce que vous avez à faire est d'ajouter cet attribut et le type de média en vous fiant au tableau suivant :

Tableau 7 Type MIME par format audio

FORMAT	TYPE
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

Un dernier exemple avec le plus d'options possibles pourrait ressembler à ceci :

```
<audio controls loop autoplay>
  Tout cela ne me dit rien qui vaille. <a href="exemple.mp3">Allez voir
  ailleurs si j'y suis</a>.
  <source src="exemple.wav" type="audio/wav">
  <source src="exemple.ogg" type="audio/ogg">
  <source src="exemple.mp3" type="audio/mpeg">
</audio>
```

## L'ÉLÉMENT <VIDEO>

L'élément **VIDEO** ressemble beaucoup à l'élément **AUDIO** dans son implémentation. Mis à part le fait qu'il offre quelques attributs de plus, le principe est à peu près le même.

Pour intégrer une vidéo dans une page de la manière la plus simple, ce code suffit :

```
<video src="unevideodechaton.webm" controls>
  Votre navigateur est pathétiquement incompatible et horriblement retardé.
  Heureusement, j'ai pensé à vous et vous pouvez télécharger <a
  href="unevideodechaton.webm">ma super vidéo de chaton</a>.
</video>
```

---

## LES ATTRIBUTS

L'élément **VIDEO** utilise les mêmes attributs que l'élément **audio** et il en offre quelques autres.

---

### SRC

Comme pour l'élément **AUDIO**, la valeur de cet attribut pointe vers un fichier vidéo dans un format compatible avec les navigateurs. Nous verrons plus bas les différents formats que tolère chacun des principaux navigateurs.

---

### CONTROLS

Tout comme pour l'élément **AUDIO**, il est préférable d'offrir à vos visiteurs une forme de contrôle sur le défilement de la vidéo. Les contrôles sont:

- Play
- Pause
- Volume

Certains navigateurs peuvent offrir d'autres contrôles, cela dépend du navigateur et de la version.

---

### AUTOPLAY

Cet attribut permet de démarrer la vidéo automatiquement. Veuillez noter qu'en date de publication, cet attribut ne fonctionne pas pour les appareils mobiles.

---

### HEIGHT AND WIDTH

Il est possible de modifier la hauteur ou la largeur originale de la vidéo. Si vous spécifiez les deux, attention de préserver la proportion sinon vous allez modifier la géométrie de l'image.

---

### POSTER

La valeur de cet attribut pointe vers une image qui servira de vignette qui s'affiche quand la vidéo n'est pas encore en mode lecture.

---

### LOOP

La présence de cet attribut indique que la vidéo doit jouer en boucle.

---

## MUTED

La présence de cet attribut indique que le volume de la vidéo (ou le contrôle mute) est à 0. Il n'y aura donc pas de son quand la vidéo démarre.

```
<video src="unevideodechaton.webm" controls height="250px"
poster="chaton.jpeg" autoplay muted loop>
Vous avez pensé à installer un navigateur plus récent? Les mises à jour sont
gratuites... Je vous permets, malgré tout, de télécharger <a
href="unevideodechaton.webm">ma super vidéo de chaton</a>.
</video>
```

---

## COMPATIBILITÉ DES FORMATS VIDÉO

Tout comme pour les fichiers audio, les différents formats vidéo ne font pas tous l'unanimité parmi les navigateurs.

Tableau 8 Compatibilité des formats vidéo

NAVIGATEUR	MP4	WEBM	OGG
INTERNET EXPLORER	Oui	Non	Non
CHROME	Oui	Oui	Oui
FIREFOX	Oui	Oui	Oui
SAFARI	Oui	Non	Non
OPERA	Oui	Oui	Oui

Comme vous pouvez le voir, Internet Explorer et Safari ne supportent pas tous les formats de vidéo. En utilisant l'élément source de la même manière que pour l'élément audio, il est possible d'offrir au navigateur une alternative.

---

## L'ÉLÉMENT <VIDÉO> ET L'ÉLÉMENT <SOURCE>

Tout comme pour l'élément **AUDIO**, si vous spécifiez le type de vidéo avec l'attribut **type**, vous accélérerez le choix de la vidéo pour le navigateur.

Tableau 9 Format vidéo et leur type MIME

FORMAT	TYPE
MP4	video/mp4
WEBM	video/webm
OGG	video/ogg

Voici le code pour la même vidéo que dans l'exemple du début de cette section mais offerte en différents formats.

```
<video controls height="250px" poster="chaton.jpeg" autoplay muted loop>
  <source src="unevideodechaton.webm" type="video/webm">
  <source src="unevideodechaton.ogg" type="video/ogg">
  <source src="unevideodechaton.mp4" type="video/mp4">
```

```
Vous avez pensé à installer un navigateur plus récent? Les mises à jour sont gratuites... Je vous permets, malgré tout, de télécharger <a href="unevideodechaton.webm">ma super vidéo de chaton</a>.</video>
```

Un des plus grand avantage d'HTML5 est de prendre en charge la lecture des fichiers multimédia. Auparavant, la prise en charge de la vidéo et du son était la responsabilité des systèmes d'opération. Ce qui causait bien des soucis aux développeurs web car on ne pouvait jamais avoir la certitude que notre contenu allait être compatible avec les appareils de tous les visiteurs.

Comme les développeurs de navigateur prennent maintenant en charge la lecture de ces types de fichiers, que la comptabilité est clairement définie et qu'elle est la même peu importe la plateforme, cela rend la création de pages web beaucoup plus simples.

De plus, avec les années, deux formats ont vraiment pris la première place:

- mp3 pour les fichiers audio
- mp4 pour les fichiers vidéo

Mais sachez qu'à l'apparition des éléments **VIDEO** et **AUDIO** dans les navigateurs, aucun des formats n'était supporté par tous les navigateurs, il aura fallu quelques années pour voir un format commun accepté par les navigateurs principaux.

Ceci dit, la meilleure pratique reste encore l'utilisation de l'élément **SOURCE** pour offrir des options aux visiteurs qui ont des navigateurs un peu moins à jour.

Et finalement, ne soyez pas aussi sarcastiques que moi avec votre code...

## LES TABLEAUX

Les tableaux servent à organiser de l'information en lignes et en colonnes. Dans ce guide, vous avez déjà vu quelques exemples de tableaux comme par exemple le *Tableau 9 Format vidéo et leur type MIME*.

Il y a très longtemps, les développeurs web utilisaient les tableaux pour faire de la mise en page. Sachez que c'est une très mauvaise idée et le simple fait d'en parler me donne des frissons de dégoût et d'horreur. Pour faire la composition de vos pages web et y placer les éléments, vous devrez apprendre CSS.

### PLANIFIER

Avant de faire un tableau, il faut planifier comment les données qu'il contient seront présentées.

1. *Est-ce que la première ligne sera un titre de colonne?*
2. *Est-ce que la première colonne sera un titre de ligne?*
3. *Combien de colonnes et de lignes seront nécessaires?*

### LES BALISES

Une fois que vous savez combien de colonnes et combien de lignes vous utiliserez (et n'oubliez pas la colonne d'entête et la ligne d'entête dans votre calcul), vous pouvez commencer à placer vos balises.

---

#### L'ÉLÉMENT TABLE

C'est la première balise à placer et elle sert simplement à délimiter le début et la fin du tableau.

Dans la norme HTML5, tous les attributs propres aux tableaux sont obsolètes. Ce sont des attributs qui permettent de définir comment le tableau sera affiché dans la page :

- **align,**
- **border,**
- **cellpadding,**
- **cellspacing,**
- **width**

Ces attributs sont obsolètes parce que dans la philosophie HTML5, la présentation (ou l'affichage) est séparée du contenu sémantique. Essentiellement, le code HTML5 ne contiendra que l'information : c'est à dire le texte, son rôle dans le document (sémantique) et sa structure (la position dans la hiérarchie des éléments).

Il en va de même pour les tableaux HTML5. Le but est de créer la structure du tableau avec HTML et de contrôler l'apparence du tableau avec CSS.

---

#### L'ÉLÉMENT <TH>

Cet élément permet de désigner une cellule qui sert de titre de colonne ou de ligne.

- **<th scope="row">** pour un titre de ligne,
- **<th scope="column">** pour un titre de colonne.

---

## L'ÉLÉMENT TR

Les cellules que cet élément contient seront sur une même ligne. Cet élément peut contenir des éléments `<th>` et `<td>`.

---

## L'ÉLÉMENT TD

Le contenu de cet élément sera une cellule du tableau.

---

## L'ÉLÉMENT CAPTION

À ces éléments, on peut ajouter l'élément `<caption>`. Celui-ci est très utile pour rendre votre tableau utilisable pour les personnes avec un handicap visuel mais aussi pour le référencement dans les moteurs de recherche.

Son contenu sera une description courte du tableau.

## DES EXEMPLES

Maintenant que nous avons vu tous ces éléments, nous pouvons commencer à créer des tableaux simples.

```
<table>
  <caption>
    Première version du navigateur compatible avec l'élément <caption>;
  </caption>
  <tr> <!-- Première ligne -->
    <th scope="row">
      Internet Explorer
    </th>
    <td>
      6.0
    </td>
  </tr>
  <tr> <!-- Deuxième ligne -->
    <th scope="row">
      Firefox
    </th>
    <td>
      2.0
    </td>
  </tr>
  <tr> <!-- Troisième ligne -->
    <th scope="row">
      Chrome
    </th>
    <td>
      1.0
    </td>
  </tr>
</table>
```

---

## DEUXIÈME TABLEAU



```
<table>
  <caption>
    Historique des versions HTML.
  </caption>
  <tr>
    <th scope="column">
      HTML 1.0
    </th>
    <th scope="column">
      HTML 2.0
    </th>
    <th>
      HTML 3.2
    </th>
    <th scope="column">
      HTML 4.0 (XHTML 1.0)
    </th>
    <th scope="column">
      HTML 5.0
    </th>
  </tr>
  <tr>
    <td>
      1991
    </td>
    <td>
      1994
    </td>
    <td>
      1997
    </td>
    <td>
      2000
    </td>
    <td>
      2008
    </td>
  </tr>
</table>
```

En combinant ces éléments, vous pouvez créer des tableaux de très grandes tailles mais vous vous rendez compte qu'il devient parfois difficile de s'y retrouver, particulièrement quand le tableau que vous construisez contient un très grand nombre de colonne. Si vous faites votre code à la main, ce que je recommande fortement pour apprendre, vous pouvez utiliser les commentaires HTML comme je l'ai fait dans le premier exemple pour vous aider à vous repérer.

## D'AUTRES GUIDES

Visitez le site des guides efficaces pour découvrir d'autres guides.

Inscrivez-vous à la liste pour être informé de la publication de nouveaux guides.

<http://lesguidesefficaces.com/>