

Partie 1

HTML5 - Pour mieux présenter vos pages Web

Le HTML

HTML est l'abréviation de « **HyperText Markup Language** », soit en français « langage de balisage hypertexte ».

Ce langage a été créé en 1991 et a pour fonction de structurer et de donner du sens à du contenu.

C'est un langage de description de pages Web. Il permet de définir le contenu et de présenter le document.

Grâce au **HTML**, on va par exemple pouvoir indiquer au navigateur que tel texte doit être considéré comme un simple paragraphe ou que tel autre est un titre.

Le langage **HTML** est structuré en **balises** qui peuvent ou non accueillir des attributs.

Une page **HTML** est donc un simple fichier texte contenant des balises permettant de mettre en forme la page (les titres, les paragraphes, les images, les liens...) et peut donc très facilement être conçue grâce à un simple éditeur de texte.

Les outils pour préparer et tester notre travail :

Nous avons besoin seulement d'un éditeur de texte pour écrire notre code **HTML**. Nous pourrions très bien prendre le bloc note de Windows, mais pour nous faciliter le travail, nous pouvons télécharger un éditeur un peu plus évolué (Exemple : NotePad++) Et d'un navigateur internet pour tester notre page style Internet Explorer ou Mozilla Firefox ou Chrome ou encore Safari ou ...

Le XHTML

« **Extensible Hypertext Markup Language** », Créé en Janvier 2000, le **XHTML 1.0** a pour but de se conformer à la norme XML (Extensible Markup Language) "tout en gardant" les propriétés du HTML 4.1. Pour simplifier, c'est une reformulation du langage **HTML** à la sauce **XML**.

Pourquoi le XHTML :

- a) **Le principal but est de séparer la forme et le contenu** : La multiplication des produits se connectant à l'internet tel que le PDA, téléphones mobiles multimédias, I-mode ..., va nous obliger à séparer contenu et mise en forme. Grâce à cela, vous n'aurez qu'un seul contenu et plusieurs mises en formes suivant le support de lecture de la page.
- b) **Rendre le document lisible par la machine** : La rigueur du XML va faciliter l'interprétation du document par les machines. Ces machines sont entre autre les robots indexeurs de page Internet.
- c) **Le XHTML est extensible** : Le XHTML va vous permettre d'incorporer un autre langage de balises, tel que le SVG, MathML ou XSLT,... au sein même du document XHTML à condition qu'une DTD (Document Type Definition) qui définit le langage soit introduit au document et que le navigateur soit capable de la traiter.

Le HTML5

La toute nouvelle version du langage HTML soutenue par W3C et qui a longtemps fait parler de lui avant sa finalisation (et même après).

HTML5 est destiné à devenir une révolution du Web. En effet il a proposé des innovations importantes telles que la création de nouvelles balises (dites sémantiques), la prise en charge du flux multimédia (audio et vidéo) sans que le navigateur ne sollicite le service des plugins, les animations 2D et 3D...

Nouveautés de HTML5 :

Vous allez voir durant ce cours qu'entre HTML et HTML5, il y a une grande ressemblance. D'ailleurs la grande majorité des balises de HTML reste valable. Cependant HTML5 a apporté plus de richesse fonctionnelle qu'on peut résumer dans ces points:

- **Nouveau conteneurs:** HTML5 a innové de nouvelles balises (conteneurs) dites sémantiques. Ces balises ont un rendu identique à celui de la balise `<div>`, mais ils ont un sens que le navigateur et les outils de recherche et d'indexation connaissent. D'où le nom de Web sémantique (ou Web3.0).
- **Prise en charge du flux multimédia:** HTML5 permet d'intégrer directement un contenu multimédia (audio, vidéo ou animation) sans que le navigateur n'aie besoin de plugin. Jusqu'ici, pour intégrer une vidéo par exemple, il fallait faire appel à la balise `<object>`. Cette balise invoque le plugin installé sur le navigateur et qui peut comprendre le type de flux intégré. Désormais, le navigateur peut déchiffrer un contenu multimédia d'une façon autonome.
- **De nouveaux champs de formulaire:** HTML5 propose de nouveaux champs de formulaire de la famille `<input>` qui s'avèrent très utiles lors de leur saisie.
- **D'autres fonctionnalités qui mettent en valeur le contenu:** Des fonctionnalités tel que glisser-déposer (drag & drop), accès aux périphériques tel que la webcam... on été également ajoutées.

Fonctionnement des balises

Structure des balises :

Les balises délimitent une fraction du document HTML (paragraphe, titre...). Elles peuvent être imbriquées entre elles et se présentent comme ci-dessous :

```
<balise attribut1= "valeur1" attribut2="valeur2"> Une chaîne de caractères </balise>
```

On peut remarquer qu'une balise peut contenir trois types d'informations :

- Un nom de balise.
- Une liste d'attributs.
- Une chaîne de caractères : Titre, Paragraphe, Citation

Les balises sont toujours écrites en minuscule et fonctionnent généralement par paire: une balise ouvrante `<balise>` , une balise fermante `</balise>` délimitant ensembles le contenu.

Néanmoins quelques balises ne fonctionnent pas par paire. Les balises d'images et de saut de ligne par exemple auront la syntaxe suivante : **<balise />**

Règles d'imbrication :

Dans l'exemple, ci-dessous, la balise 1 est parente de la balise 2.

```
<balise1> <balise2> (...) </balise2> (...) </balise1>
```

Lorsque plusieurs balises sont imbriquées, une règle simple s'applique : une balise ouverte à l'intérieur d'une autre doit obligatoirement être refermée avant la fermeture de sa balise mère.

Définition de Type de Document : Le DTD

La première balise de votre document doit être la balise de **DTD (Document Type Definition)**. Elle sert à indiquer à quelles règles d'écriture obéit le code d'une page web et pour informer le validateur de W3C, de la version du langage utilisé. Il n'est pas nécessaire de retenir les différents DTD. Pour coder en HTML5 il suffit de commencer par :

```
<!DOCTYPE html>
```

Le HTML5 : Structure générale

Structure :

Après la **DTD**, il suffit d'ouvrir la balise **<html>** qui englobera l'ensemble des autres balises de votre page. Cette balise contient deux paires de balises:

- Les balises **<head>** et **</head>** délimitant l'entête du document.
- Les balises **<body>** et **</body>** délimitant le corps du document.

L'entête :

L'entête contient toutes les données relatives au document (le titre du document et sa description par exemple).

a) Le titre

Le plus important dans l'en-tête est le titre de la page, il se définit avec les balises **<title>** :

```
<title> titre de votre page </ title >
```

b) Les balises **<meta>** :

Les balises **<meta>** sont facultatives, mais importantes pour le navigateur, et pour les moteurs de recherche.

✓ Encodage de la page :

```
<meta charset= UTF-8" />
```

L'attribut charset (sur <meta>)

Préciser l'encodage des caractères est primordial pour exploiter la bonne page de code et ne pas se retrouver avec les caractères spéciaux ou accentués. Le choix de l'UTF-8 est désormais préconisé par le W3C pour tous les protocoles échangeant du texte sur internet (dont HTML).

✓ Langue de la page

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

Le corps :

Le corps de votre page est la partie visible du document HTML, il peut être constitué d'un grand nombre de types de balises différents (titres, paragraphes, images, liens,...).

Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre du document</title>
    <meta charset=UTF-8" />
  </head>
  <body>
    <!--je suis un commentaire et je ne serai pas affiché -->
    <h1>Titre de niveau 1</h1>
    
    <br />
    <p> Texte non formaté.</p>
  </body>
</html>
```

Aperçu :



Mise en forme de texte

Titres

Le langage HTML définit 6 niveaux de titres, les balises <h1>, <h2>, <h3>, <h4>, <h5> et <h6> (<h1> et <h6> représentent respectivement la plus et la moins importante). Elles permettent de définir une structuration hiérarchique des paragraphes d'une page.

Paragraphes

Les paragraphes sont délimités par les balises <p> et </p>. Ils ne doivent être employés que lorsque leur contenu ne peut être présenté d'aucune autre manière (liste, tableau,...).

Retour à la ligne

Les retours à la ligne sont définis à l'intérieur d'un paragraphe par la balise unique **
**

Ligne horizontale

Des lignes horizontales peuvent être ajoutées aux paragraphes, elles sont définies par la balise unique **<hr/>**.

Textes préformatés

Les balises **<pre>** permettent d'indiquer qu'une zone de texte doit être présentée par le navigateur, en respectant la mise en page du code source.

Les acronymes

Les Balises **<abbr>** HTML5 – Définit une abréviation ou un acronyme au survol par infobulle grâce à l'attribut **title**.

Les citations

La balise **<blockquote>** définit une citation. Elle spécifie une section qui est citée par une autre source.

Les balises de styles

Texte en gras

Les textes en gras peuvent être définis par les balises **** et ****.

Texte en italique

Les textes en italique peuvent être définis par les balises **** et ****.

Texte marqué

Les textes marqués peuvent être définis par les balises **<mark>** et **</mark>**. Ce lot de balises est utilisé pour faire ressortir du contenu en appliquant un style surligné jaune sur le texte qu'elle décrit.

Texte en indice

Les textes en indice peuvent être définis par les balises **_{** et **}**.

Texte en exposant

Les textes en indice peuvent être définis par les balises **^{** et **}**.

Remarque :

- Il est conseillé d'utiliser les propriétés CSS pour appliquer certains styles à votre contenu ! Tel les balises **<center>**, ****, **<u>**, **<s>**

Les liens

Structure

Les liens hypertextes sont définis par les balises **<a>** associées à l'attribut **href** qui spécifie la source du document cible. Deux types de liens peuvent être définis :

- Liens internes = entre 2 pages d'un même site.
- Liens externes = d'un site vers un autre site.

Liens externes :

Pour des liens externes, on spécifie un chemin absolu. La valeur de l'attribut **href** est l'adresse du site.

```
<a href="http://www.votre-site.fr/index.html"> Accueil </a>
```

Liens internes :

Pour créer des liens internes, on utilise un chemin relatif. 3 cas :

- Même dossier : href = « page2.html »
- Sous-dossier : href = « sous_dossier/page2.html »
- Dossier parent : href=« ../page2.html »
- Lien menant à un autre endroit de la même page web : spécifier un **id** puis #.

Liens spéciaux

Lien vers adresse de messagerie électronique :

```
<a href="mailto:vous@domaine.com">votre texte</a>
```

Lien externe vers un serveur FTP :

```
<a href="ftp:ftp.site.com">votre texte</a>
```

Les images

Les images sont définies par la balise **** associée à l'attribut **src** qui spécifie la source de l'image et à l'attribut **alt** qui spécifie un texte alternatif au cas où l'image ne s'afficherait pas.

```

```

Les tableaux

Il est souvent utile de présenter des informations en ligne et en colonne, les tableaux sont la pour ça. Les tableaux sont définis par les balises suivantes:

- Le tableau est encadré par les balises **<table>**
- Le titre du tableau est encadré par les balises **<caption>**
- Les lignes sont encadrées par les balises **<tr>**
- Les cellules d'en-tête sont encadrées par les balises **<th>**
- Les cellules de valeur sont encadrées par les balises **<td>**

```
<table>
  <caption>Titre du tableau </caption>
  <tr>
    <th> Titre 1 </th>
    <th> Titre 2 </th>
  </tr>
  <tr>
    <td> Valeur 1 </td>
    <td> Valeur 2 </td>
  </tr>
</table>
```

Attributs

- **colspan** définit le nombre de colonnes que la cellule doit occuper.
- **rowspan** définit le nombre de lignes que la cellule doit occuper.

Les listes

Listes de définitions

Les listes de définitions sont définies par les balises suivantes:

- La liste est encadrée par les balises **<dl>**
- Les termes sont encadrés par les balises **<dt>**
- Les définitions sont encadrées par les balises **<dd>**

```
<dl>
  <dt>Terme à Définir</dt>
  <dd>Définition du Terme</dd>
</dl>
```

Listes numérotées

Les listes numérotées sont définies par les balises suivantes:

- La liste est encadrée par les balises ****
- Les termes de la liste sont encadrés par les balises ****

Listes à puces

Les listes à puces sont définies par les balises suivantes:

- La liste est encadrée par les balises ****
- Les termes de la liste sont encadrés par les balises ****

```
<ul>
  <li>Premier terme</li>
  <li>Deuxième terme</li>
</ul>
```

Structurer sa page : ELEMENTS STRUCTURELS DU HTML5

En général, une page web est constituée d'un en-tête (tout en haut), de menus de navigation (en haut ou sur les côtés), de différentes sections au centre... et d'un pied de page (tout en bas).

En HTML 4 il n'y a pas de balises de structuration spécialisées, le contenu est structuré avec des `<div>`, ``.

HTML 5 introduit plusieurs balises pour aider à représenter la structure usuelle des documents :

`<header>`

En-tête de document web. Il s'agit généralement d'une grande bande horizontale, située en haut du site avec un grand titre et/ou un logo.

`<footer>`

Le pied de page. Contenant 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.

`<nav>`

Ce conteneur est destiné à enclore (encercler) un groupe de liens.

`<section>`

Les sections délimitent les parties du contenu. Très basiquement, on peut encadrer une section par une bordure, ou la séparer de ce qui précède par un espace.

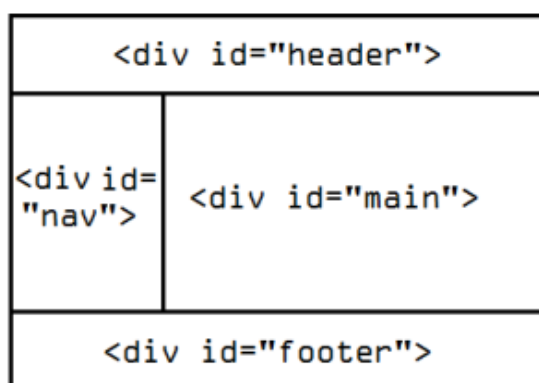
Généralement, pour respecter des bonnes pratiques, on commencera une section avec un élément de **titre**.

`<article>`

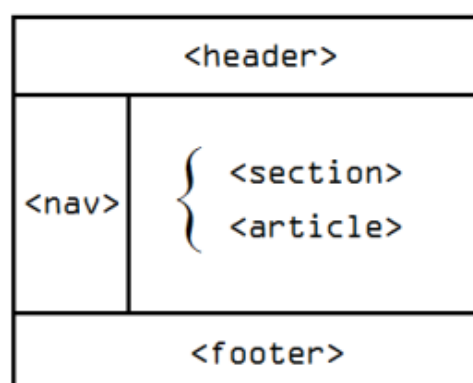
Représente un texte indépendant, une portion de contenu, comme par exemple un article de journal, de blog ou de forum.

`<aside>`

Représente un contenu annexe (non liée) au contenu principal mais qui peut fournir des informations supplémentaires, liées à ce contenu (par exemple une biographie de l'auteur, des liens sur le sujet, bannière publicitaire, liens vers l'extérieur, etc.) et peut définir une barre latérale.



HTML4



HTML5

La vidéo et l'audio en HTML5

Depuis l'arrivée de Youtube et Dailymotion, il est devenu courant aujourd'hui de regarder des vidéos sur des sites web.

Cependant, aucune balise HTML ne permettait jusqu'ici de gérer la vidéo. Il fallait à la place utiliser un plugin, comme Flash. Encore aujourd'hui, Flash reste de loin le moyen le plus utilisé pour regarder des vidéos sur Youtube, Dailymotion, Vimeo et ailleurs. Mais utiliser un plugin a de nombreux défauts : on dépend de ceux qui gèrent le plugin (en l'occurrence, l'entreprise Adobe, qui possède Flash), on ne peut pas toujours contrôler son fonctionnement, il y a parfois des failles de sécurité... Au final, c'est assez lourd.

C'est pour cela que deux nouvelles balises standard ont été créées en HTML5 : **<video>** et **<audio>** !

Insertion d'un élément audio

La balise **<audio>** que nous allons découvrir est reconnue par tous les navigateurs récents, y compris Internet Explorer à partir de la version 9 (IE9).

En théorie, il suffit d'une simple balise pour jouer un son sur notre page :

```
<audio src="musique.mp3"></audio>
```

Vous pouvez compléter la balise des attributs suivants :

- **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, et vous vous demandez peut-être pourquoi cela n'y figure pas par défaut, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript.
- **width** : pour modifier la largeur de l'outil de lecture audio.
- **loop** : la musique sera jouée en boucle.
- **autoplay** : la musique sera jouée dès le chargement de la page. Évitez d'en abuser, c'est en général irritant d'arriver sur un site qui joue de la musique tout seul !
- **preload** : indique si la musique peut être préchargée dès le chargement de la page ou non.

Cet attribut peut prendre les valeurs :

- **auto** (par défaut) : le navigateur décide s'il doit précharger toute la musique, uniquement les métadonnées ou rien du tout.
- **metadata** : charge uniquement les métadonnées (durée, etc.).
- **none** : pas de préchargement. Utile si vous ne voulez pas gaspiller de bande passante sur votre site.

```
<audio src="musique.mp3" controls></audio>
```

L'apparence du lecteur audio change en fonction du navigateur. La figure suivante représente par exemple le lecteur audio dans Google Chrome.



Le lecteur audio dans Google Chrome

Pourquoi ouvrir la balise pour la refermer immédiatement après ?

Cela vous permet d'afficher un message ou de proposer une solution de secours pour les navigateurs qui ne gèrent pas cette nouvelle balise. Par exemple :

```
<audio src="hype_home.mp3" controls>Veuillez mettre à jour votre navigateur !</audio>
```

Ceux qui ont un navigateur récent ne verront pas le message. Les anciens navigateurs, qui ne comprennent pas la balise, afficheront en revanche le texte qui se trouve à l'intérieur.

Je vous conseille de proposer une solution de secours en Flash, comme Dewplayer. Vous placerez le code correspondant à Flash entre les balises <audio> et </audio> : ainsi, les anciens navigateurs afficheront le lecteur Flash, tandis que les nouveaux afficheront le lecteur natif.

On a vu que certains navigateurs ne géraient pas le MP3, comment faire ?

Il faut proposer plusieurs versions du fichier audio. Dans ce cas, on va construire notre balise comme ceci :

```
<audio controls>
  <source src="hype_home.mp3"></source>
  <source src="hype_home.ogg"></source>
</audio>
```

Le navigateur prendra automatiquement le format qu'il reconnaît.

Insertion d'une vidéo

La balise <video> que nous allons découvrir est reconnue par tous les navigateurs récents, y compris Internet Explorer à partir de la version 9 (IE9).

Il suffit d'une simple balise <video> pour insérer une vidéo dans la page :

```
<video src="sintel.webm"></video>
```

Mais, là encore, vous risquez d'être déçus si vous utilisez seulement ce code. Aucun contrôle ne permet de lancer la vidéo !

Rajoutons quelques attributs (la plupart sont les mêmes que pour la balise <audio>) :

- poster : image à afficher à la place de la vidéo tant que celle-ci n'est pas lancée. Par défaut, le navigateur prend la première image de la vidéo mais, comme il s'agit souvent d'une image noire ou d'une image peu représentative de la vidéo, je vous conseille d'en créer une ! Vous pouvez tout simplement faire une capture d'écran d'un moment de la vidéo.
- controls : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript. En ce qui nous concerne, ce sera largement suffisant !
- width : pour modifier la largeur de la vidéo.
- height : pour modifier la hauteur de la vidéo.
- loop : la vidéo sera jouée en boucle.
- autoplay : la vidéo sera jouée dès le chargement de la page. Là encore, évitez d'en abuser, c'est en général irritant d'arriver sur un site qui lance quelque chose tout seul !
- preload : indique si la vidéo peut être préchargée dès le chargement de la page ou non. Cet attribut peut prendre les valeurs :

- auto (par défaut) : le navigateur décide s'il doit précharger toute la vidéo, uniquement les métadonnées ou rien du tout.
- metadata : charge uniquement les métadonnées (durée, dimensions, etc.).
- none : pas de préchargement. Utile si vous souhaitez éviter le gaspillage de bande passante sur votre site.

Voici un code un peu plus complet :

```
<video src="sintel.webm" controls poster="sintel.jpg" width="600"></video>
```

Pourquoi ouvrir et refermer immédiatement après la balise ?

La réponse est la même que pour la balise <audio>. Cela vous permet d'afficher un message ou d'utiliser une technique de secours (en Flash) si le navigateur ne reconnaît pas la balise :

```
<video src="sintel.webm" controls poster="sintel.jpg" width="600">
  Il est temps de mettre à jour votre navigateur !
</video>
```

Comment contenter tous les navigateurs, puisque chacun reconnaît des formats vidéo différents ?

Vous utiliserez la balise <source> à l'intérieur de la balise <video> pour proposer différents formats. Le navigateur prendra celui qu'il reconnaît :

```
<video controls poster="sintel.jpg" width="600">
  <source src="sintel.mp4" />
  <source src="sintel.webm" />
  <source src="sintel.ogv" />
</video>
```

Les iPhone, iPad et iPod ne reconnaissent à l'heure actuelle que le format H.264 (fichier .mp4)... et uniquement si celui-ci apparaît en premier dans la liste ! Je vous recommande donc d'indiquer le format H.264 en premier pour assurer une compatibilité maximale.

Comment afficher la vidéo en plein écran ?

Ce n'est pas possible à l'heure actuelle. En fait, il existe bien un moyen sous Firefox mais il est un peu caché : il faut faire un clic droit sur la vidéo, puis sélectionner « Plein écran ».

Comment protéger ma vidéo, je ne veux pas qu'on puisse la copier facilement !

Ce n'est pas possible. Les balises n'ont pas été conçues pour limiter ou empêcher le téléchargement. C'est assez logique quand on y pense : pour que le visiteur puisse voir la vidéo, il faut bien de toute façon qu'il la télécharge d'une manière ou d'une autre !

N'espérez donc pas empêcher le téléchargement de votre vidéo avec cette technique.

Les lecteurs vidéo Flash permettent de « protéger » le contenu des vidéos mais, là encore, des solutions de contournement existent. De nombreux *plug-ins* permettent de télécharger les vidéos, de Youtube par exemple.

Les formats audio et vidéo Supportés

Les formats audio

Pour diffuser de la musique ou n'importe quel son, il existe de nombreux formats. La plupart d'entre eux sont compressés (comme le sont les images JPEG, PNG et GIF) ce qui permet de réduire leur poids :

- **MP3** : vous ne pouvez pas ne pas en avoir entendu parler ! C'est l'un des plus vieux, mais aussi l'un des plus compatibles (tous les appareils savent lire des MP3), ce qui fait qu'il est toujours très utilisé aujourd'hui.
- **AAC** : utilisé majoritairement par Apple sur iTunes, c'est un format de bonne qualité. Les iPod, iPhone et autres iPad savent les lire sans problème.
- **OGG** : le format Ogg Vorbis est très répandu dans le monde du logiciel libre, notamment sous Linux. Ce format a l'avantage d'être libre, c'est-à-dire qu'il n'est protégé par aucun brevet.
- **WAV (format non compressé)** : évitez autant que possible de l'utiliser car le fichier est très volumineux avec ce format. C'est un peu l'équivalent du Bitmap (BMP) pour l'audio.

Aucun navigateur ne gère tous ces formats à la fois. Retenez surtout la compatibilité pour les MP3 et OGG :

Navigateur	MP3	OGG
Internet Explorer	Oui	-
Chrome	Oui	Oui
Firefox	-	Oui
Safari	Oui	-
Opera	-	Oui

Il n'y a pas de format « idéal » reconnu par tous les navigateurs ?

Eh non ! Heureusement, on pourra proposer différents formats aux navigateurs qui sélectionneront alors celui qu'ils savent lire.

Les formats vidéo

Le stockage de la vidéo est autrement plus complexe. On a besoin de trois éléments :

- **Un format conteneur** : c'est un peu comme une boîte qui va servir à contenir les deux éléments ci-dessous. On reconnaît en général le type de conteneur à l'extension du fichier : AVI, MP4, MKV...
- **Un codec audio** : c'est le format du son de la vidéo, généralement compressé. Nous venons de les voir, on utilise les mêmes : MP3, AAC, OGG...
- **Un codec vidéo** : c'est le format qui va compresser les images. C'est là que les choses se corsent, car ces formats sont complexes et on ne peut pas toujours les utiliser gratuitement. Les principaux à connaître pour le Web sont :

- **H.264** : l'un des plus puissants et des plus utilisés aujourd'hui... mais il n'est pas 100% gratuit. En fait, on peut l'utiliser gratuitement dans certains cas (comme la diffusion de vidéos sur un site web personnel), mais il y a un flou juridique qui fait qu'il est risqué de l'utiliser à tout va.

- **Ogg Theora** : un codec gratuit et libre de droits, mais moins puissant que H.264. Il est bien reconnu sous Linux mais, sous Windows, il faut installer des programmes pour pouvoir le lire.
- **WebM** : un autre codec gratuit et libre de droits, plus récent. Proposé par Google, c'est le concurrent le plus sérieux de H.264 à l'heure actuelle.

Quelle est la compatibilité des codecs vidéo sur les différents navigateurs ? Là encore, vous allez voir que c'est un joyeux bazar :

Navigateur	H.264	Ogg Theora	WebM
Internet Explorer	Oui	-	Oui
Chrome	-	Oui	Oui
Firefox	-	Oui	Oui
Safari	Oui	-	-
Opera	Oui	Oui	Oui

Là encore, aucun format ne sort du lot. Il est conseillé de proposer chaque vidéo dans plusieurs formats pour qu'elle soit lisible sur un maximum de navigateurs.

Pour convertir une vidéo dans ces différents formats, je vous conseille l'excellent logiciel gratuit [Miro Video Converter](#) que vous pouvez télécharger.

Il vous suffit de glisser-déposer votre vidéo dans la fenêtre du programme et de sélectionner le format de sortie souhaité. Cela vous permettra de créer plusieurs versions de votre vidéo !