



Webdesigner

Une formation de Bruxelles Formation CEPEGRA

HTML 5

Hypertext Markup Language

PRÉAMBULE

Faisons une petite incursion de l'autre côté du miroir de CSS, dans le langage de structure HTML.

La version 5 nous promet plein de nouvelles choses : nouvelles balises sémantiques, nouvelles fonctionnalités pour nos chers formulaires, l'appel à du contenu multimédia (audio et vidéo) sans passer par flash ou javascript, une nouvelle syntaxe et le tout en vraiment simplifié par rapport aux précédentes versions de notre ami HTML.

Le W3C prévoit de clôturer une fois pour toute les specs d'HTML 5 pour 2014 (environ ☺) mais en attendant, le W3C encourage les développeurs Web à utiliser HTML 5 dès maintenant et voici donc la raison de la naissance de ce mini-module consacré à l'avenir de la structuration de nos pages Web à travers l'évolution d'un langage qui a déjà connu quelques versions jusqu'à présent.

Nous verrons les nouveautés essentielles et à notre portée de ce nouveau langage lors de ce cours mais pour être complet voici toutes les nouveautés apportées par HTML 5 :

- Une grammaire entièrement revue et simplifiée ;
- De nouveaux éléments sémantiques et de nouveaux attributs ;
- La reconnaissance de vidéos ou de sons sans plug-ins ;
- Une gestion étendue des formulaires ;
- La possibilité de se localiser géographiquement et de profiter de cette information ;
- La création de dessins ou de graphiques à l'aide de l'élément <canvas> ;
- Le stockage de données sur son ordinateur pour les exploiter hors ligne ;
- La possibilité d'intervenir sur les éléments en les modifiant à la volée ou en les déplaçant (drag & drop), etc.



TABLE DES MATIÈRES

PRÉAMBULE	2
HTML5 : LA RÉVÉLATION	4
• Nouvelle grammaire.....	4
Simplification du !DOCTYPE	4
Une syntaxe permissive	4
• Nouvelles balises sémantiques	5
<header> et <hgroup>	5
<footer>	5
<nav>	5
<aside>	5
<section>	5
<article>	5
<figure>	6
• Nouvelles balises multimédia	7
<audio>	7
<video>	8
• Des formulaires améliorés.....	9
De nouveaux types de champs	9
Range	10
date, datetime, month, week, time	10
search	10
De nouveaux attributs intelligents	11
Placeholder	11
Autofocus	11
Autocomplete	11
Required	12
• Des attributs puissants.....	12
Draggable	12
Contenteditable	12
• Vers un HTML5 « transitionnel » ?	13

HTML5 : LA RÉVÉLATION

● Nouvelle grammaire

Avec HTML 5, vous allez voir qu'écrire des pages devient de plus en plus facile, simple et à portée de tous.

Si vous venez d'une application de l'XHTML 1.0 Strict, la bonne nouvelle c'est que ça ne va pas changer grand-chose pour vous : vous pourrez toujours coder comme précédemment, c'est juste qu'il y a maintenant une façon plus simple d'envisager les choses et qui sera également correcte !

Simplification du !DOCTYPE

En HTML 5, la syntaxe du doctype a complètement été revue et simplifiée :

```
<!doctype html>
```

Mais ce n'est pas tout ! D'autres éléments du <head> de notre document ont aussi été simplifiés, comme la balise <meta> qui précise l'encodage des caractères du document, on ne doit plus écrire que ceci dorénavant :

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

Une syntaxe permissive

En HTML 5, les règles syntaxiques sont de plus en plus permissives. Par exemple, on conserve la permissivité syntaxique de l'HTML 4 : majuscules autorisées pour les noms d'éléments et d'attributs, apostrophes simples ou doubles non obligatoires pour signaler les valeurs.

Plus fort encore ! Les balises <p>, , <optgroup>, <option>, <td>, <th> ne nécessitent pas de balise fermante pour être valides. Seule la version XHTML 5 oblige à fermer ces éléments.

Toujours plus fort ! Certains éléments ne nécessitent ni balise fermante ni balise ouvrante. C'est le cas de <html>, <head>, <body>. Cela signifierait que la présence même de ces éléments devient implicite.

Notons finalement que l'attribut type (qu'on rentre principalement sur les balises <script> et <style>) devient superflu.

Ce qui voudrait dire par exemple que le code suivant serait parfaitement valide :

```
<!doctype html>
<meta charset="utf-8">
<title>Titre de la page</title>
<p>Hello World !</p>
```

Mais bon ce n'est pas parce que c'est toléré qu'il faut forcément le faire ... Je vous invite quand même à garder une certaine rigueur dans la structure de vos pages web. Gardez donc le réflexe de bien écrire vos balises habituelles.

● Nouvelles balises sémantiques

En plus de la mini-révolution au niveau de la simplification de l'écriture d'éléments tels que le Doctype ou des balises meta, HTML5 ouvre la voie à de nouvelles possibilités en terme de balisage sémantique.

De quoi enfin se passer de nos chères balises `<div>` ? Pas forcément mais en tous cas, vous verrez que vous en aurez nettement moins l'utilité. La balise `<div>` restant l'élément qu'il faut choisir si vous n'avez pas d'autre choix en termes de balise de type block et qu'aucune autre des balises déjà existantes mais aussi des nouvelles balises ci-dessous ne fait l'affaire.

Voici une liste des nouvelles balises sémantiques que vous allez pouvoir utiliser dès maintenant :

`<header>` et `<hgroup>`

La balise `<header>` représente le bloc d'en-tête d'une section ou d'une page. Il remplace évidemment vos précédentes balises `<div id="header">`, mais ne doit pas forcément être considéré comme un élément unique dans votre document : toute section est susceptible de disposer de sa propre balise `<header>`.

`<hgroup>`, elle, sert simplement à grouper des éléments de type « titres » (`<h1>` à `<h6>`) ensemble.

`<footer>`

La balise `<footer>` regroupe les contenus du pied d'une section ou d'un document (pied de page) et est destiné à recueillir les informations concernant l'auteur, les mentions légales, etc. Comme pour la balise `<header>`, vous pourriez retrouver plusieurs fois la balise `<footer>` sur la même page.

`<nav>`

La balise `<nav>` regroupe les liens de navigation considérés comme majeurs ou jugés suffisamment pertinents. Ceux-ci peuvent être internes ou externes à la page.

`<aside>`

La balise `<aside>` représente une portion de contenu contextuelle, directement ou indirectement liée aux éléments qui l'entourent, comme un bloc d'archives relatives au contenu précédent. Par extrapolation, cet élément désigne fréquemment les barres latérales classiques du document et peut remplacer l'ancien `<div id="sidebar">`.

`<section>`

La balise `<section>` est là pour désigner un bloc générique de contenu ayant la même thématique. Cela pourrait concerner les chapitres, en-têtes et pieds de page, ou toute autre partie dans un document. L'élément `<section>` peut contenir des éléments de titre `<h1>` à `<h6>` pour avoir une meilleure définition de la structure du document.

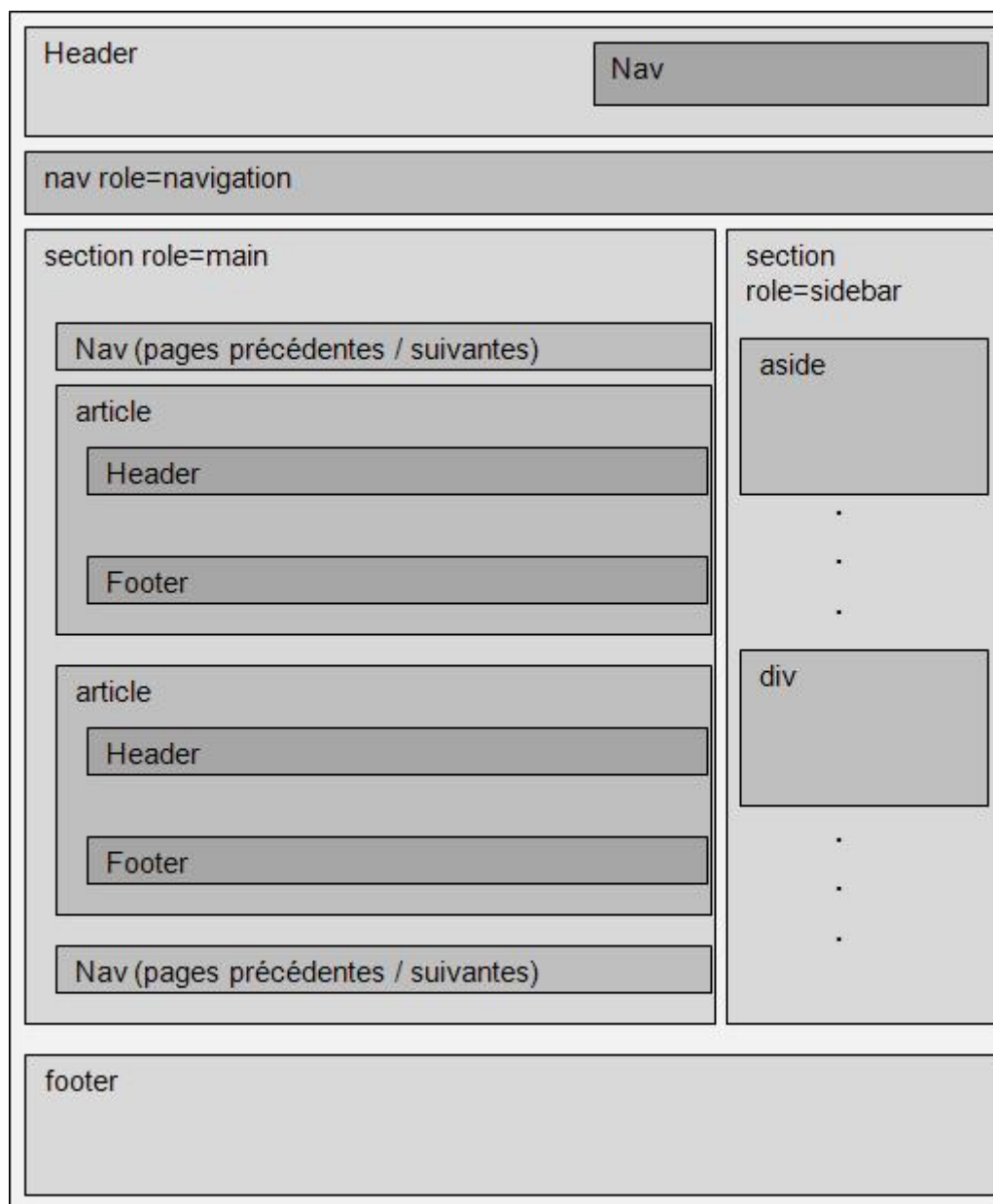
`<article>`

La balise `<article>` désigne une partie du document potentiellement autonome dans le sens où elle pourrait être reprise ou réutilisée, comme un article de journal, de blog ou de forum.

<figure>

La balise <figure>, selon le W3C est à employer combinée à <figcaption> pour structurer des illustrations, photos, diagrammes et portions de codes. <figcaption> désignera la légende de la figure.

Voici une représentation graphique de ce que pourrait être une structure HTML5 de nos jours :



Cependant, il reste encore un petit problème : certains navigateurs n'affichent généralement pas ces éléments correctement (ils sont identifiés comme étant des éléments inline plutôt que block).

Internet Explorer 8 est encore plus difficile puisque, rien que pour lui, il va falloir aussi ajouter une petite couche de javascript pour que tout cela fonctionne.

Concrètement, il suffira d'appeler un script externe dès le début de la page de manière spécifique au navigateur de Microsoft :

```
<!--[if lt IE 9]>
<script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

Tandis que pour les autres navigateurs, il suffira d'indiquer ceci côté CSS :

```
article, aside, details, figcaption, figure, footer, header, hgroup, nav,
section { display: block; }
```

● Nouvelles balises multimédia

En plus des nouveautés liées à la sémantique, HTML5 s'ouvre au multimédia grâce à deux nouvelles balises : <audio> et <video>.

<audio>

La balise <audio> permet la lecture d'un fichier audio, aux formats classiques, sans recourir à un player extérieur ou propriétaire (flash ou javascript). Sa syntaxe est aussi courte qu'efficace :

```
<audio src="la-danse-des-canards.mp3"></audio>
```

On peut ajouter des attributs facultatifs : `autoplay` (lecture automatique) et `controls` (affichage des boutons de contrôle de lecture).

Le tout combiné donnerait donc, pour une musique lancée dès le lancement de votre page :

```
<audio src="la-danse-des-canards.mp3" autoplay controls></audio>
```

Attention, au niveau des formats de fichiers à fournir, c'est un peu là que le bât blesse pour le moment : chaque navigateur a ses petits préférés et, de manière générale, tous sont tombés d'accord sauf ... Internet Explorer !

En fournissant un fichier .mp3 et un fichier .ogg, vous devriez pouvoir baliser tous les navigateurs sauf IE6, 7 et 8 ! Au niveau syntaxique voici comment s'articule la syntaxe quand vous avez plusieurs fichiers à injecter :

```
<audio>
  <source src="la-danse-des-canards.mp3">
  <source src="la-danse-des-canards.ogg">
</audio>
```

Chaque navigateur a donné un look différent à son player :



<video>

Même principe que pour la balise <audio>, ici la balise <video> va vous permettre, à terme, de vous passer des plugins propriétaires pour placer une vidéo sur vos pages web.

Au niveau syntaxique, on reste dans la simplicité :

```
<video src="the-dark-knight-rises.mp4"></video>
```

Pareil que pour <audio>, les attributs `controls` et `autoplay` sont encore présents.

Un troisième attribut est intéressant dans ce cas-ci car il permet de pré-loader la vidéo en imaginant que l'utilisateur va la regarder par la suite, il s'agit de `preload`.

Au niveau des formats à utiliser, il vous faudra en prévoir 2 pour couvrir tous les navigateurs : du mp4 et de l'ogv. Au niveau syntaxique, ça donne ceci :

```
<video>
  <source src=" the-dark-knight-rises.mp4" type="video/mp4">
  <source src=" the-dark-knight-rises.ogv" type="video/ogg">
</video>
```

Attention

Si vous désirez que votre fichier vidéo soit visible sur un périphérique Apple (iPhone, iPod et iPad), il faut absolument commencer par le fichier .mp4 sinon le fichier ne sera pas lu du tout.

Au niveau du skin des players, c'est la même chose que pour la balise <audio> mais avec la vidéo incrustée par-dessus et la possibilité sur certains navigateurs de passer la vidéo en full-screen (cool!).

• Des formulaires améliorés

Dans cette jungle de nouveautés, même les formulaires ont été revus, corrigés et surtout améliorés.

Pas mal de nouveautés voient le jour : de nouveaux types de champs, de nouvelles fonctions, des tests d'expressions régulières sur leur contenu textuel et bien d'autres choses encore ...

De nouveaux types de champs

HTML5 introduit de nouveaux types de champs de formulaires. Parmi ceux-ci :

- `email` : le champ requiert un contenu au format d'adresse électronique.
- `url` : le champ accueille des URL absolues.
- `tel` : le champ est destiné aux numéros de téléphone
- `number` : le champ accepte uniquement les caractères numériques.
- `color` : le champ est prévu pour les chaînes représentant une valeur de couleur.
- `range` : le champ devient un curseur défilant, ou slider.
- `Search` : le champ désigne un champ de recherche.
- `date`, `datetime`, `month`, `week` et `time` : ces champs affichent des datepickers pour les unités qu'ils représentent.

Ces différents types viennent s'ajouter aux valeurs classiques de HTML4 à savoir `submit`, `image`, `text`, `radio`, `button`, `checkbox`, `hidden`, `file` et `password`.

Au niveau syntaxique, rien ne change dans l'écriture de l'`input` :

```
<input type="email" id="email">
```

Chacun des types de champ obéit à une norme qui définit les valeurs acceptées ou rejetées. Ainsi, il est possible de vérifier quels éléments sont invalides à l'aide des sélecteurs CSS3 `:valid` et `:invalid` :

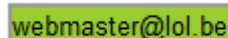
CSS

```
input[type="email"]:valid {background:yellowgreen;}  
input[type="email"]:invalid {background:#F03;}
```

HTML

```
<input type="email" id="email">
```

Un champ valide :



Un champ non valide :



CSS3, associé à HTML5, ont donc réussi à comprendre que ce qui se trouvait dans le champ était valide dans le premier cas et non valide dans le second cas, car l'adresse mail n'est pas complète.

Range

Les champs de type range se présentent sous la forme d'un curseur défilant, ou slider.

Au niveau syntaxique, il s'agit de nouveau de la balise `<input>` :

```
<input type="range" min="0" max="50" value="0">
```

Ce qui génère, sous chrome, un curseur de ce type :



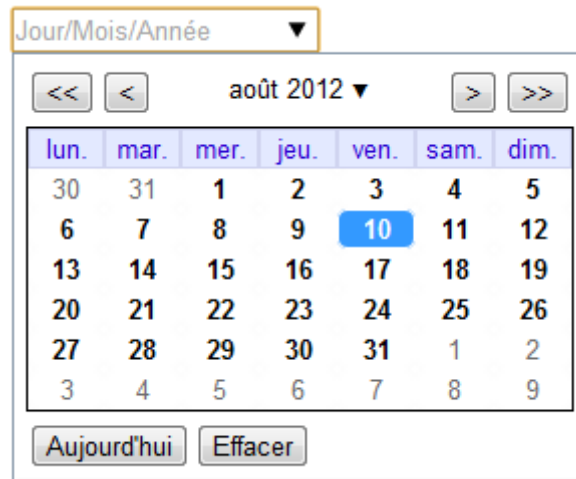
Vous aurez remarqué que certains attributs sont là pour définir les propriétés du curseur :

- `min` et `max` : définit la valeur minimale et maximale disponible sur la barre (il y a donc 50 états différents dans ce cas-ci).
- `value` : l'endroit où se trouve le curseur par défaut (ici 0 donc au début de la ligne).
- `step` : définit le nombre d'unités passées à chaque fois (une valeur de 2 ferait passer votre curseur de 0 à 2 puis 4, puis 6, etc.).

date, datetime, month, week, time

On en parlait un peu plus haut, des nouveaux types d'input pour préciser un « moment » ont vu le jour : `date`, `datetime`, `month`, `week` et `time`.

Voici un exemple d'un input de type « date » sur chrome avec datepicker :



search

Voici un type d'input qui désigne donc un champ de recherche.

Petit particularité, sous Chrome, une petite croix apparait quand vous commencez à écrire dans le champ pour pouvoir le réinitialiser :



De nouveaux attributs intelligents

Avec ces nouveaux types de champs débarquent aussi de nouveaux attributs « intelligents » qui vous permettent : de pré-écrire dans un champ, de mettre le focus automatique sur un champ présent dans une page, d'activer l'auto-complétion dans certains champs, de faire en sorte que certains champs soient obligatoires et d'autres pas...

Placeholder

Commençons avec `placeholder` qui va vous permettre de pré-remplir vos champs de type input par du texte qui va disparaître dès que vous commencerez à écrire dedans.

Autre effet magique du champ marqué d'un `placeholder`, si vous le videz à nouveau alors qu'il était rempli, il récupère la valeur indiquée initialement par le `placeholder`, this is madness !

Voici à quoi ressemblerait un champ email rempli avec un placeholder qui a pour valeur « Votre email » :

```
HTML
<input type="email" placeholder="Votre email">
```



Autofocus

L'attribut `autofocus` place votre curseur directement dans un champ quand votre page est entièrement chargée.

Même code que ci-dessus mais avec un autofocus placé en attribut :

```
HTML
<input type="email" placeholder="Votre email" autofocus>
```



Autocomplete

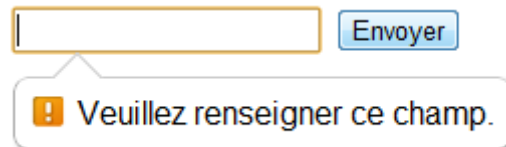
L'attribut `autocomplete` affiche une boîte contenant les dernières entrées de formulaire que vous avez saisies et qui ont été conservées en mémoire sur votre machine. La valeur par défaut est `on` (activé), mais il est possible de masquer ces termes à l'aide de la valeur `off` (pour le champ de répétition de l'email, en général, c'est la technique qui est employée).

Required

Les champs bénéficiant de l'attribut `required` doivent absolument être remplis lors de la soumission du formulaire, sans quoi la validation est refusée et le champ concerné est mis en évidence.

Pour le bout de code suivant, voici le rendu sur Chrome si vous ne remplissez pas le champ requis :

```
HTML
<form>
  <input type="text" required>
  <input type="submit" value="Envoyer">
</form>
```



• Des attributs puissants

Et ce n'est pas tout !

HTML5 permet aussi une autre flopée de choses : rendre un élément du DOM draggable (déplaçable) au click, rendre du contenu éditable directement, utiliser la géolocalisation, faire des glisser-déposer, et bien d'autres choses. Nous allons seulement en voir deux ou trois car cet aspect-là d'HTML5 est encore relativement peu supporté et nécessite parfois des connaissances en javascript complémentaires.

Draggable

L'attribut `draggable` (ou `draggable="true"`) donne à un élément de votre DOM la possibilité d'être « drag and drop » grâce à HTML5. Pour réussir l'effet complet, vous avez néanmoins besoin également d'une petite couche de javascript.

Contenteditable

L'attribut `contenteditable` est reconnu depuis longtemps par IE et il indique une zone éditable par l'utilisateur. Il peut en changer son contenu et manipuler ainsi les chaînes de caractères.

```
<p contenteditable="true">Ha bein ça alors, ce texte est éditable !</p>
```

Ce qui nous donne :

Ha bein ça alors, ce texte est éditable (c'est super ^^) !

● Vers un HTML5 « transitionnel » ?

HTML5 marque le coup en simplifiant un bon lot de la rigueur qui nous avait été ordonnée par l'xhtml 1.0 Strict mais il ouvre aussi des portes concernant les nouvelles technologies contemporaines en décrivant mieux le contenu affiché sur une page web, en améliorant la gestion des périphériques et en favorisant l'intégration des applications web.

La démocratisation de ce langage facilitera dans un avenir « relativement » proche l'interopérabilité des documents HTML, mais aussi leur accessibilité universelle, voire leur référencement (grâce aux toutes nouvelles balises sémantiques).

En attendant, quelle position adopter par rapport à cette nouvelle spécification qui n'est pas encore officielle ?

- S'il s'agit seulement de se simplifier la vie en passant par une syntaxe HTML5 plus élémentaire (Doctype abrégé, etc.) et plus courte (plus besoin de type= ..., plus besoin de / pour les balises auto-fermantes, etc.) alors le risque d'incompatibilité est carrément nul et rien ne vous empêche d'appliquer dès aujourd'hui ces nouveautés à vos pages Web.
- Si vous désirez utiliser les nouvelles balises sémantiques (<aside>, <footer>, <article>, etc.) le risque est un peu plus grand : non seulement vous prenez le pari que vos visiteurs sur IE disposent tous de Javascript (ok ça c'est souvent le cas), mais vous pourriez altérer l'accessibilité à vos documents en optant pour un nouvel élément HTML5 qui n'est pas encore standardisé.
- Les nouveaux champs de formulaires comptent parmi les éléments les moins reconnus par les navigateurs (mais ça progresse bien dans les versions récentes). Mais vous pouvez néanmoins les employer (<input> avec les type email, url, search, number ou date) dès à présent et, s'ils ne sont pas compris, ils seront tout simplement traités comme des champs <input> classiques de type text.
- Pour ce qui est de l'utilisation des API multimédia comme <audio>, <video> ou <canvas>, certains sites les utilisent déjà comme Google ou Youtube par exemple. Mais tout ceci reste encore un peu instable au niveau des spécifications et surtout il faut prévoir beaucoup d'alternatives quand vous désirez mettre un média en ligne pour qu'il soit accessible par tous. Bref, méfiance !

En plus des nouveautés que nous avons vues ensemble, notons encore **ces points importants** au niveau de la structure même de nos pages :

- Plusieurs éléments H1 sont applicables à différents niveaux de la hiérarchie.
- Les éléments de lien (<a>..<) peuvent maintenant contenir des éléments de type bloc.
- Les éléments <body> et <head> ne sont plus nécessaires ! Mais je vous les recommande toujours vivement !
- Les attributs ALT sur les balises ont été remis en cause mais aucune décision n'a été prise les concernant à l'heure actuelle.
- Utiliser les balises <audio>, <video> et <canvas> demande aux navigateurs un réel support sinon elles ne fonctionneront pas, c'est pourquoi chaque navigateur a un style de player différent et une façon plus ou moins complète d'intégrer ces nouvelles fonctionnalités.

Restez donc encore prudents dans vos choix liés à l'utilisation d'HTML5 !