

# La balise video

## 1 Préparation de la page HTML

Faire et refaire, c'est toujours travailler. Comme pour l'audio, il nous faut une page HTML, une autre ...

### 1.1 Mise en place de la page

Il nous faut travailler avec un éditeur de code HTML, mais un simple éditeur de texte peut faire l'affaire. Les captures d'écran sont faites avec le Notepad++, mais ce n'est qu'un éditeur parmi d'autres.

Vous devez créer un nouveau fichier, dans le dossier pagesVideo, de votre répertoire personnel, le nommer index.html et lui donner les attributs d'un fichier HTML :

```

index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5  <body>
6  </body>
7  </html>

```

à savoir une définition de type : `<!DOCTYPE html>` et bien évidemment le jeu de balises `<html>` encadrant les balises `<head>`, `<body>`.

**Attention** à bien indiquer un encodage en **UTF8** pour avoir un affichage correct des caractères accentués.

Rien que du classique, il n'est nullement précisé que le code utilisé est en HTML version 5, juste du HTML. C'est un des avantages du HTML, les balises sont posées, disponibles, mais si le navigateur n'est pas capable de les traiter, il les ignore totalement, c'est donc sans risque !

### 1.2 Premiers éléments

Un titre pour la page, pour faire joli dans le navigateur, cela fera toujours quelque chose à afficher si jamais on se trompe !

Balise `<title>` dans l'en-tête, Page de test vidéo fera un très bon titre !

```

index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Page de test vidéo</title>
5  </head>
6  <body>
7  </body>
8  </html>

```

Testez votre page dans votre navigateur préféré, après l'avoir enregistrée dans votre répertoire personnel, menu Exécution / Launch in ...

Satisfaits, vous n'avez pas perdu la main à force d'utiliser des CMS ?

Bon, on accélère.  
(moins drôle la seconde fois !)

Le but étant de simplement créer une page pour tester la balise video, on va faire simple, pas d'appel à une feuille de style, pas de script, rien que du HTML pur et dur, toujours et encore !

Pas besoin d'une balise paragraphe, car l'avantage de la vidéo, c'est d'être justement visible, quand cela fonctionne.

## 2 Insertion de la balise video

### 2.1 <video>

La balise video est presque aussi simple que la balise audio et comporte elle aussi que peu d'attributs. Comme pour audio, l'attribut `src` est essentiel :

`src` qui prend pour valeur le chemin d'accès au fichier, en relatif ou absolu : `src="chemin du fichier vidéo"`

Bien évidemment, vous avez pensé à déplacer les fichiers des vidéos encodées dans le dossier pagesVideo ! Revenons à notre fichier, modifiez le code selon les indications de la capture d'écran suivante :

# La balise video

```

5 | </head>
6 | <body>
7 | <video src="Clement1.mp4">
8 | </body>
9 | </html>

```

Plus simple, il n'y a pas ! Enregistrez votre fichier HTML et demandez son exécution depuis votre navigateur préféré ... Alors ? Ben oui, on l'a vu tout à l'heure, le fichier MP4 est bien là,

chargé à priori, mais la lecture ne s'effectue pas !

Comme pour l'audio, le fichier « Clement1.mp4 » est correctement cité dans la page, présent dans le même dossier que le fichier HTML, mais si l'on voit bien la première image de la vidéo, la lecture ne se fait pas ! Il va falloir compléter la balise avec les attributs qui vont bien.

## 2.2 Quelques attributs

Dans les attributs de notre balise, le premier, par ordre alphabétique, se nomme `autoplay`, quelle chance, cet argument est également valable pour une vidéo. Cet attribut est évidemment de type booléen. Ce qui nous donne :

```

3 | <head>
4 |   <title>Page de test vidéo</title>
5 | </head>
6 | <body>
7 |   <video src="Clement1.mp4" autoplay>
8 | </body>
9 | </html>

```

Enregistrez, exécutez ... Mieux non, sauf pour ..... !  
Ah oui, quand même !

## 2.3 Encore un peu plus ?

Les problèmes liés au format de vidéo décodables par les navigateurs sont plus présents, à l'heure actuelle, en vidéo qu'en audio (cf document Test navigateurs). Pour être sûr d'être vu, quel que soit le navigateur utilisé, il est conseillé d'utiliser un pointage vers au moins deux versions de fichier encodés différemment, ou même les trois pour être tranquille. Cela s'obtient en utilisant l'élément `source`. Cet élément est à utiliser lorsque plusieurs encodages différents sont mis à disposition du navigateur Internet. Dans ce cas, le navigateur utilise le premier dans la liste qui correspond à un codec connu du navigateur, ce qui s'écrit alors :

```

<video autoplay>
  <source src="chemin_fichier_encodé_format_H264.mp4">
  <source src="chemin_fichier_encodé_format_OggTheraVorbis.ogv">
  <source src="chemin_fichier_encodé_format_WebM.webm">
</video>

```

Ce qui, dans le cas de notre page, devient :

```

6 | <body>
7 |   <video src="Clement1.mp4" autoplay>
8 |     <source src="Clement1.mp4">
9 |     <source src="Clement1.ogv">
10 |    <source src="Clement1.webm">
11 |   </video>
12 | </body>

```

Enregistrez votre page, testez la dans les 5 navigateurs. Tous sont capables de lire le fichier vidéo !

Et on peut encore faire mieux, en insérant un message d'avertissement pour le cas où le navigateur de l'utilisateur ne prend pas en charge le HTML5 (trop ancien), à l'intérieur de la balise `video` :

```
<p>Ce fichier vidéo n'est pas jouable, mettez votre navigateur à jour pour en profiter !</p>
```

## 2.4 Toujours plus

Proposer plusieurs formats de données peut-être utile, mais on peut aussi indiquer de façon explicite le format utilisé de façon à accélérer la traduction de la page : pas besoin d'ouvrir le fichier pour voir ce qu'il y a dedans si on me le précise par ailleurs. C'est le rôle de l'attribut `type`, voir dans la capture suivante.

# La balise video

Et si des incertitudes subsistent, il est conseillé d'utiliser les arguments `Content-Type` et `codecs` (cf RFC4281 : <http://tools.ietf.org/html/rfc4281>)

Et ce n'est pas tout !

L'attribut `controls` est aussi disponible, il permet l'affichage d'une barre de contrôle de type lecteur vidéo. Cet attribut est lui-aussi de type booléen : si je l'écris, je le vois, si je l'oublie tant pis pour moi.

Modifiez votre fichier HTML comme indiqué dans la capture suivante avant de l'enregistrer et de la tester.

```

7 <video src="Clement1.mp4" controls>
8   <source src="Clement1.mp4" type="video/mp4">
9   <source src="Clement1.ogv" type="video/ogg">
10  <source src="Clement1.webm" type="video/webm">
11  <p>Ce fichier vidéo n'est pas jouable, mettez votre navigateur à jour pour en profiter !</p>
12 </video>

```

Pour être exhaustif, il existe d'autres attributs pour la balise `video`.

`loop` permet une lecture en boucle du fichier audio, peut parfois, dans de rares cas, être utile, votre vidéo se bouclant par exemple (certaines publicités le font très bien).

`preload`, comme son nom le laisse supposer, autorise le préchargement du, ou des fichiers, vidéo.

Ces deux attributs sont de type booléen.

`height` et `width` utilisées ensemble permettent d'imposer des dimensions différentes des dimensions propres de la vidéo, attention au ratio ! (la solution : ne préciser qu'une seule des deux valeurs)

`muted` permet d'avoir une vidéo réduite au silence, pendant la lecture, à l'utilisateur de modifier grâce à la barre de contrôle (si affichée !)

Ce qui nous donne, par exemple :

```

6 <body>
7   <video src="Clement1.mp4" controls width="640" loop muted>
8     <source src="Clement1.mp4" type="video/mp4">
9     <source src="Clement1.ogv" type="video/ogg">
10    <source src="Clement1.webm" type="video/webm">
11    <p>Ce fichier vidéo n'est pas jouable, mettez votre navigateur à jour pour en profiter !</p>
12  </video>
13 </body>
14 </html>

```

Enregistrez puis testez votre page dans chaque navigateurs. Satisfaits ?

## 2.5 Encore plus loin

Un dernier attribut est disponible pour la balise `video`, il s'agit de l'attribut `poster` qui va nous permettre de remplacer la première image de la vidéo, qui, par défaut est affichée dans le lecteur et souvent noire ou peu représentative, par une image de notre choix. Cette image peut être une capture faite lors de la lecture de la vidéo, ou tout autre chose comme le montre l'exemple suivant.

```

6 <body>
7   <video src="Clement1.mp4" controls width="640" loop muted poster="titre1.png">
8     <source src="Clement1.mp4" type="video/mp4">
9     <source src="Clement1.ogv" type="video/ogg">
10    <source src="Clement1.webm" type="video/webm">
11    <p>Ce fichier vidéo n'est pas jouable, mettez votre navigateur à jour pour en profiter !</p>
12  </video>
13 </body>
14 </html>

```

Le fichier `titre1.png` se trouve dans le dossier `Ressources`.

Si vous avez testé votre page avec les 5 navigateurs majeurs, vous avez certainement remarqué les différences sensibles au niveau des barres de lecture du fichier vidéo, ainsi que les disparités au niveau des fonctionnalités offertes par ces barres ou les menus contextuels. Voir les feuilles 2 et 3 de test navigateurs.

Et si on en veut plus, avoir des boutons différents, ou tous les mêmes quel que soit le navigateur ?

La réponse est simple : non, pas avec HTML5 en l'état, pas maintenant, peut-être un jour ...

Ou alors, on fait appel aux deux copains de HTML, JS et CSS ! Mais là, c'est une autre histoire ...