

Leçon 5-2 – HTML5 : contenus embarqués

s1 -----

Dans cette leçon, nous allons introduire les balises de contenu embarqué. On appelle contenu embarqué un fichier que la page HTML va afficher. Ces balises permettent d'afficher notamment des images ou de la vidéo ainsi que de jouer du son.

s2 -----

On examinera ici successivement les balises

- image
- map
- figure

qui ne sont pas des nouvelles balises HTML5 et les balises

- audio
- et video

qui, quant à elles, sont nouvelles.

s3 -----

Quelques précisions sur la balise `` que nous avons déjà rencontrée semaine 1.

La balise `` permet d'afficher une image stockée dans un fichier externe. Différents formats de compression sont possibles (GIF, PNG, JPEG...). Ils présentent les uns et les autres des avantages et des inconvénients que vous pourrez aisément trouver de façon détaillée sur le web.

Les attributs de cette balise `img` sont listés dans ce tableau. Le plus important d'entre eux étant évidemment l'attribut `src` qui définit le chemin relatif du fichier à afficher, c'est-à-dire à partir de l'endroit où est localisé le fichier HTML.

L'attribut `alt` est optionnel. Il permet de définir un texte alternatif qui sera affiché par exemple si l'image est manquante ou lu par un outil de synthèse vocale.

Les attributs `width` et `height` redéfinissent respectivement la largeur et la hauteur de l'image en pixels. Ces attributs sont optionnels. En leur absence, l'image est affichée à sa taille originelle.

L'attribut `ismap` est un booléen qui permet d'envoyer au serveur les coordonnées de l'endroit de l'image où un clic a été effectué. Nous ne le testerons pas ici.

L'attribut `usemap` est lié à la définition de zones cliquables par-dessus une image. Nous allons voir ça dans la diapositive suivante.

Avant cela, je vous propose de mettre en oeuvre l'exemple donné ici avec une image de votre choix, que vous placerez dans un dossier appelé `images` situé au même niveau que le fichier HTML. Dans un second temps, vous pouvez tester l'effet des attributs `width` et `height`.

s4 -----

Un mot sur la balise `<map>`.

La balise `<map>` permet de définir des zones cliquables de forme rectangulaire, circulaire ou polygonale par-dessus une image.

Examinons l'exemple ci-dessous. Dans la balise `img`, on a rajouté l'attribut `usemap` (que l'on a découvert précédemment) qui précise quelle map sera utilisée. Ici on utilisera la map appelée `map1` (attention à ne pas oublier l'hashtag). On définit ensuite la map avec ici deux zones cliquables ; la première est un rectangle dont le coin supérieur gauche est en 0,0 (c'est-à-dire le coin supérieur gauche de l'image définie dans la balise `img`) et dont le coin inférieur droit est en 40,40 (ces dimensions sont en pixels) ; la seconde zone est un cercle de centre 150,150 et de rayon 20 (ces dimensions sont en pixels).

Un clic sur la zone rectangulaire affiche la page `p1.html` ; un clic sur la zone circulaire affiche la page `p2.html`.

Je vous propose de mettre en oeuvre l'exemple donné ici avec une image de votre choix. Il vous faudra bien entendu créer les pages `p1.html` et `p2.html`.

s5 -----

Un mot rapide sur la balise `<figure>`.

La balise `<figure>`, utilisée en association avec une image, permet de rajouter une légende. Je vous propose de mettre en oeuvre l'exemple donné ici avec une image de votre choix.

s6 -----

L'audio, comme la vidéo, a eu sur le web un passé tourmenté, en raison des faibles débits disponibles qui n'autorisaient pas une restitution de qualité. De toute façon la norme HTML n'avait pas prévu l'emploi de tels éléments dans une page web. Avec l'avènement du haut débit ont surgi des formats de compression divers (associés à des programmes de compression et décompression, que l'on appelle codecs) et autant de plug-ins à installer afin que les navigateurs puissent afficher ces images et ces vidéos. Le format flash (propriétaire) a fini par se faire une place de choix. Nous avons en tête les sites tels que YouTube, Dailymotion.

Les balises HTML5 `audio` et `video` sont une réponse concrète à ce problème : une lecture sans plug-in de l'audio et de la vidéo dans une page web. On n'entrera pas ici dans les détails techniques de cette norme, notamment les codecs utilisés et sur les rendus différents selon le navigateur.

Un des intérêts d'HTML5 est que les ressources audio et vidéo peuvent être désormais manipulées par JavaScript, comme vous allez le découvrir dans l'exercice qui suit.

s7 -----

L'intégration d'un fichier audio est très simple : il suffit de préciser la source. Plusieurs balises sources peuvent être définies les unes en dessous des autres de façon à s'assurer qu'une majorité de navigateurs va être capable d'en lire une sans problème.

Je vous propose de mettre en oeuvre l'exemple donné ici avec un fichier audio de votre choix ou celui proposé sur la page du module.

L'attribut `controls` est un booléen qui permet d'activer ou de désactiver les contrôles visuels. Vous pouvez tester une restitution sans menu en remplaçant `controls` par un `autoplay`.

s8 -----

L'intégration d'un fichier vidéo est très simple : il suffit ici encore de préciser la source. À nouveau, plusieurs balises sources peuvent être définies les unes en dessous des autres de façon à s'assurer qu'une majorité de navigateurs va être capable d'en lire une sans problème.

Je vous propose de mettre en oeuvre l'exemple donné ici avec un fichier vidéo de votre choix ou celui proposé sur la page du module.

s9 -----

Dans cet exercice je vous propose de créer des boutons de contrôle d'une vidéo en JavaScript.

Étudiez le code ci-dessous.

Dans un premier temps, on définit une variable JavaScript ciblant l'élément de vidéo.

Si vous préférez, vous pouvez bien entendu utiliser jQuery.

On ajoute un bouton qui applique la méthode play() à cet objet.

À vous de créer un menu complet en ajoutant des boutons pause et retour au début.

Je vous demande également d'afficher la durée écoulée ainsi que la durée totale de la vidéo. Une dernière chose : à la fin de la vidéo, une image doit s'afficher sous la vidéo dans un élément div initialement vide.

Vous allez devoir chercher comment utiliser la méthode Pause et les propriétés currentTime, duration et onEnded.

Bon travail.

s10 -----

Dans la prochaine leçon, nous allons introduire la balise canvas. Grâce à JavaScript, il va être possible de créer à l'intérieur de cette balise des dessins que l'on va pouvoir transformer et même animer dynamiquement.

À très bientôt.