

## Les balises `img`, `map`, `area`, `figure`

On appelle *contenu embarqué* un fichier que la page HTML va afficher. Ces balises permettent notamment d'afficher des images ou de la vidéo, ainsi que de jouer du son.

Nous commencerons par examiner les balises (**qui ne sont pas nouvelles**) :

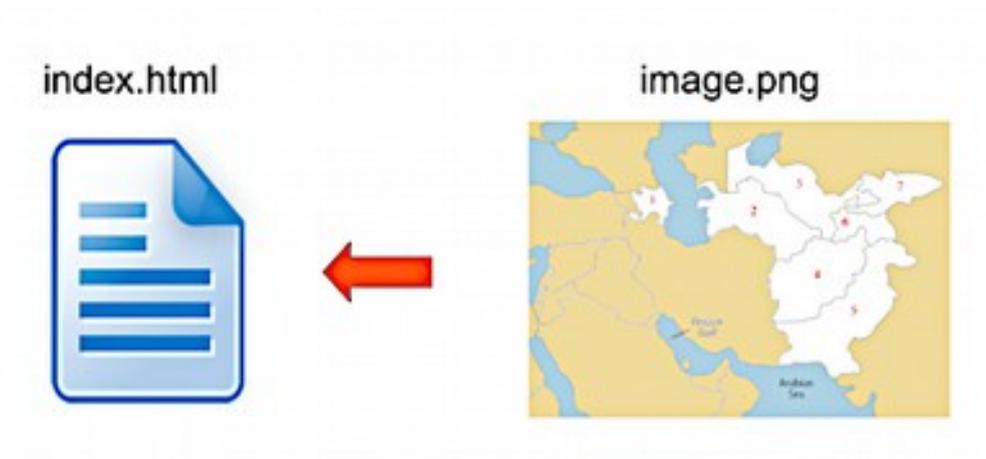
- `img` ;
- `map` ;
- `figure`.

Dans la partie suivante, nous introduirons les balises (**qui sont nouvelles**) :

- `audio` ;
- `video`.

Quelques précisions sur la balise `<img>` :

**La balise `<img>` permet d'afficher une image stockée dans un fichier externe :**



*Principe balise `img`*

**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 :**

Attribut	Valeur	Rôle
src	URL	Adresse du fichier
alt	Texte	Texte alternatif
width	Entier positif	Largeur de l'image en px
height	Entier positif	Hauteur de l'image en px
ismap	Opt. Ou ismap ou ""	Signifie un accès serveur pour la map
usemap	Chaîne	Nom de la map à utiliser

- **L'attribut src** : c'est évidemment l'attribut le plus important, il définit le chemin relatif du fichier à afficher, c'est-à-dire à partir de l'endroit où est localisé le fichier HTML.

Ensuite, à noter que l'ensemble des attributs suivants sont optionnels :

- **l'attribut alt** permet de définir un texte alternatif qui sera affiché par exemple si l'image est manquante, ou lu par un outil de synthèse vocale. Il est optionnel mais recommandé ;
- **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**.

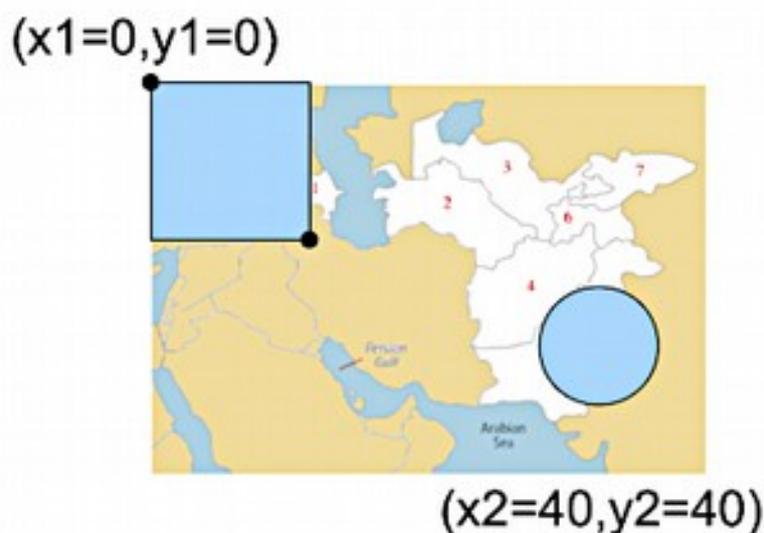
### Un mot sur la balise `<map>` :

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

Examinons l'exemple ci-dessous :

```

<map name="map1">
  <area shape="rect" coords="0,0,40,40" href="p1.html" alt="P.1">
  <area shape="circle" coords="150,150,20" href="p2.html" alt="P.2">
</map>
```



*Le code ci-dessus crée cette image et les zones cliquables correspondantes*

Dans la balise `img`, on a rajouté l'attribut `usemap` qui précise quelle map sera utilisée.

Ici, on utilisera la map appelée **map1** (attention à ne pas oublier le 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 œuvre l'exemple donné ici, avec une image de votre choix. Il vous faudra bien entendu créer les pages p1.html et p2.html, avec le contenu de votre choix.

### Sur la balise `<figure>`

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

```
<figure>
  
  <figcaption>Une photo de...</figcaption>
</figure>
```

### Les balises audio et video

Nous allons maintenant introduire les balises de contenu embarqué pour **l'audio** et la **vidéo**.

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 sons et ces vidéos.

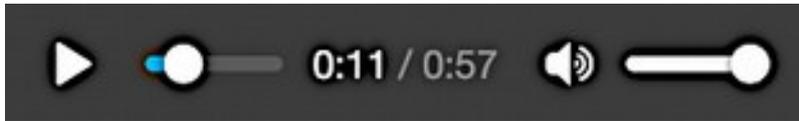
**Les balises HTML5 audio et video sont une réponse concrète à ce problème : une lecture sans plug-in de l'audio et de la vidéo dans une page web.** On n'entrera pas ici dans les détails techniques de cette norme, notamment les codecs utilisés, ni sur les rendus différents selon le navigateur.

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

**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 vont être capables d'en lire une sans problème.

Je vous propose de **mettre en œuvre l'exemple donné ci-dessous** avec un fichier audio de votre choix, ou celui proposé sur la page du module :

```
<audio controls>
<source src="Georgia.mp3" type="audio/mp3">
Votre navigateur ne supporte pas la balise audio
</audio>
```



*Le code ci-dessus fera apparaître ce contrôleur audio*

**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 vont être capables 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.

```
<video width="320" height="240" controls>
  <source src="couloir.mp4" type="video/mpeg">
  Votre navigateur ne supporte pas la balise video
</video>
```

Voici pour les balises permettant de gérer l'audio et la vidéo.

### **Exercice : audio et video**

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

Étudiez le code ci-dessous :

```
<script>
  var vid1=document.getElementById("maVideo");
</script>
<input type="button" onClick="vid1.play()" value="Play"/>
```

- dans un premier temps, on définit une variable JavaScript ciblant l'élément de vidéo. Si vous préférez, vous pouvez bien entendu utiliser jQuery pour cibler cet élément ;
- on ajoute un bouton qui applique la méthode `play()` à cet objet.

**Sur cette base, voici l'exercice :**

1. Créez un menu complet en ajoutant des boutons Pause, et Retour au début.
2. Affichez la durée écoulée, ainsi que la durée totale de la vidéo.
3. À la fin de la vidéo, une image devra s'afficher sous la vidéo dans un élément div initialement vide.

Pour mettre en oeuvre ces consignes, vous allez devoir chercher comment utiliser la méthode Pause et les propriétés *currentTime*, *duration* et *onEnded*.

Bon travail.