

Ajouter des média

Nous naviguons sur Internet à la recherche de contenu intéressant et instructif, que nous trouvons habituellement sous forme de texte. Pour accompagner ce texte brut, HTML fournit des moyens d'intégrer des supports riches sous forme d'images, de pistes audio et de vidéos, mais aussi d'intégrer du contenu d'une autre page web sous la forme d'un `iframe`.

La possibilité d'inclure des images, des pistes audio, des vidéos et des `iframes` dans des sites Web existe depuis un certain temps. Le support du navigateur pour les images et les `iframes` a généralement été assez bon. Et tandis que la capacité d'ajouter des pistes audio et des vidéos sur un site Web existe depuis des années, le processus était assez lourd. Heureusement, ce processus s'est amélioré et est beaucoup plus facile avec l'intégration de ces fonctionnalités directement dans le HTML.

Aujourd'hui, nous pouvons librement utiliser des images, des images audio, vidéo et `iframe` sachant que ce contenu est pris en charge dans tous les principaux navigateurs.

Ajouter des images

Pour ajouter des images à une page, nous utilisons l'élément en ligne ``. L'élément `` est un élément autonome ou vide, ce qui signifie qu'il n'embarque aucun autre contenu et qu'il existe sous forme de balise unique. Pour que l'élément `` fonctionne, seul l'attribut `src`, et sa valeur, est requis pour spécifier la source de l'image. La valeur de l'attribut `src` est une URL, généralement relative au serveur sur lequel un site Web est hébergé.

Conjointement avec l'attribut `src`, l'attribut `alt` (texte alternatif), qui décrit le contenu d'une image, doit être utilisé. La valeur de l'attribut `alt` est reprise par les moteurs de recherche et les technologies d'assistance pour aider à transmettre le sens de l'image. Le texte spécifié par l'attribut `alt` sera affiché à la place de l'image si, pour une raison quelconque, l'image n'est pas disponible. L'attribut `title` doit aussi être présent pour afficher une infobulle au passage de la souris.

```
1 
```



Fig 1

Le texte de remplacement, "Un Koala dans les arbres !" s'affichera à la place d'une image manquante.

Formats d'image supporté

Les images sont disponibles dans différents formats de fichiers, et chaque navigateur peut prendre en charge (ou non) différents formats. Dans l'ensemble, les formats les plus communément pris en charge sont `gif`, `jpg` et `png`. Parmi ceux-ci, les formats les plus utilisés aujourd'hui sont le `jpg` et le `png`. Le format `jpg` fournit des images de qualité avec un nombre de couleurs élevé tout en conservant une taille de fichier décente, idéal pour des temps de chargement plus rapides. Le format `png` est idéal pour les images avec transparents ou de faible nombre de couleurs. Nous voyons le plus souvent les images `jpg` utilisées pour les photographies et les images `png` utilisées pour les icônes ou les motifs d'arrière-plan.

Dimensionner une images

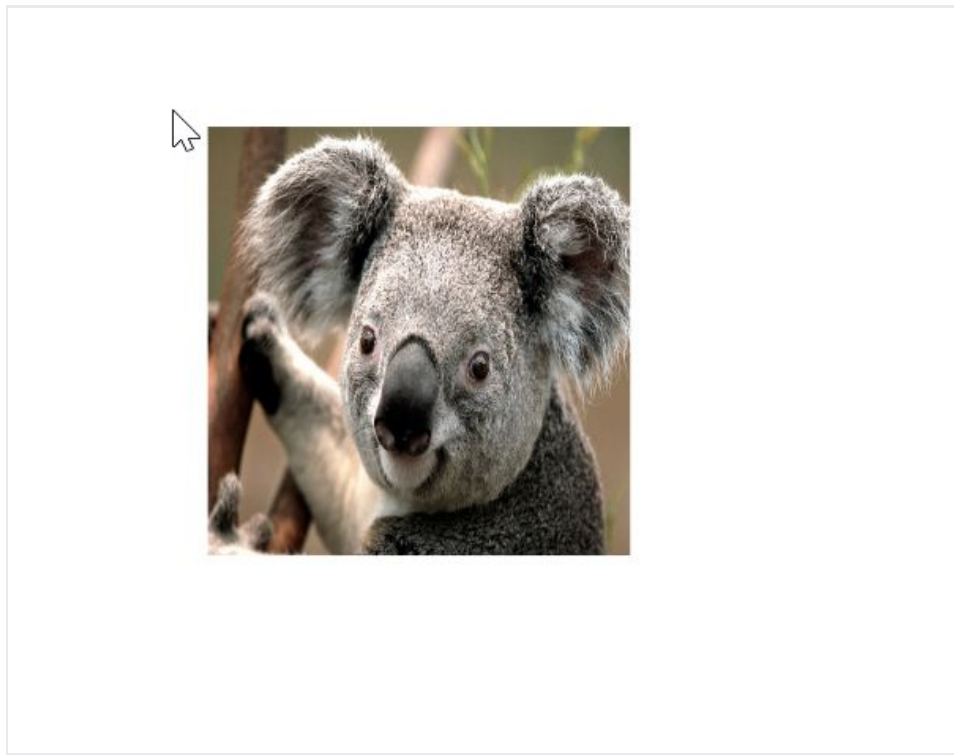
Il est important d'identifier la taille d'une image pour indiquer au navigateur la taille de l'image avant que la page ne soit chargée. Ainsi le navigateur peut réserver l'espace pour l'image et faire que la page se charge plus rapidement. Il y a plusieurs façons différentes de dimensionner les images pour qu'elles fonctionnent bien sur une page.

Une option consiste à utiliser les attributs `width` et `height` directement dans la balise `` en HTML.

En outre, les images peuvent être dimensionnées en utilisant les propriétés `width` et `height` en CSS. Lorsque les attributs HTML et les propriétés CSS sont utilisés, les attributs CSS auront priorités sur les attributs HTML.

Si vous ne spécifiez qu'une dimension, une largeur ou une hauteur, l'autre sera fonction de taille de la première. Par exemple, si nous voulons une image de 200 pixels de hauteur, nous pouvons définir la `height` à 200 pixels, et la largeur de l'image s'adaptera en conséquence. La définition d'une largeur et d'une hauteur fonctionnera également. En fixant les deux tailles, on risque de rompre le rapport de taille de l'image, ce qui aura pour effet de la déformer.

```
1 img {  
2   height: 300px;  
3   width: 300px;  
4 }
```



Lorsque vous utilisez les attributs de hauteur et de largeur directement en HTML, vous n'avez aucune information signifiante, même sur la taille d'origine d'une image et il peut être difficile de gérer de nombreuses images qui doivent toutes avoir la même taille. Dans ce cas, il est courant d'utiliser le CSS pour redimensionner les images.

Positionnement des images

Nous pouvons utiliser un certain nombre d'approches différentes pour positionner les images sur une page web. Par défaut, les images sont positionnées comme des éléments inline.

Toutefois, leurs positions peuvent être modifiées à l'aide de CSS, en particulier les propriétés `float`, `display` ainsi que les autres propriétés du modèle de boîte, y compris les `padding`, `border` et `margin`.

Positionnement inline des images

L'élément `` est par défaut un élément en ligne. L'ajout d'une image sans styles dans une page placera cette image sur la même ligne que le contenu qui l'entoure. En outre, la hauteur de la ligne dans laquelle une image apparaît sera modifiée pour correspondre à la hauteur de l'image, ce qui peut créer de grands espaces verticaux.

```

1 <p>
2   Le koala (Phascolarctos cinereus) aussi nommé Paresseux
3   australien est une espèce de marsupial arboricole
4   herbivore endémique d'Australie et le seul représentant
5   encore vivant de la famille des Phascolarctidés.
6   
8   Le koala se trouve dans les régions côtières de
9   l'Australie méridionale et orientale, d'Adélaïde à la
10  partie sud de la péninsule du cap York.
    </p>

```



Laisser les images dans leur positionnement par défaut n'est pas très courant. Le plus souvent, les images sont affichées en tant qu'élément de niveau bloc ou l'on modifie la propriété `display` à `float` sur l'un des côtés.

Positionnement de bloc

L'ajout de la propriété `display` et le réglage de sa valeur à `block` forcent l'image à devenir un élément de niveau bloc.

L'image apparaît alors sur sa propre ligne, ce qui permet de placer le contenu environnant au-dessus et au-dessous de l'image.

```

1 img {
2   display: block;
3 }

```

Le koala (*Phascolarctos cinereus*) aussi nommé Paresseux australien est une espèce de marsupial arboricole herbivore endémique d'Australie et le seul représentant encore vivant de la famille des Phascolarctidés.



Le koala se trouve dans les régions côtières de l'Australie méridionale et orientale, d'Adélaïde à la partie sud de la péninsule du cap York.

Faire flotter une image

Parfois, l'affichage d'une image comme `inline` ou `block`, ou peut-être même en `inline-block`, n'est pas idéal. Nous souhaitons peut-être que l'image apparaisse sur le côté gauche ou droit de l'élément contenant, et que le contenu puisse entourer ou longer l'image. Pour ce faire, nous utilisons la propriété `float` avec une valeur de `left` ou de `right`.

Comme nous l'avons déjà évoqué, rappelons que la propriété `float` était initialement destinée à positionner des images vers la gauche ou vers la droite d'un élément contenant.

Faire flotter une image est un début, cependant, tous les autres contenus s'aligneront directement contre elle. Pour assurer l'espacement autour d'une image, nous utiliserons la propriété `margin`. De plus, nous pouvons utiliser les propriétés de `padding`, de `border` et de `background` pour construire un cadre pour l'image.

```
1 img {  
2   background: #eaeaed;  
3   border: 1px solid #9799a7;  
4   float: right;  
5   margin: 8px 0 0 20px;  
6   padding: 4px;  
7 }
```

Le koala (*Phascolarctos cinereus*) aussi nommé Paresseux australien est une espèce de marsupial arboricole herbivore endémique d'Australie et le seul représentant encore vivant de la famille des Phascolarctidés. Le koala se trouve dans les régions côtières de l'Australie méridionale et orientale, d'Adélaïde à la partie sud de la péninsule du cap York.



Quand utiliser un élément d'image ou une image de fond ?

Il existe deux façons principales d'ajouter des images à une page Web. Une façon, telle que traitée ici, est d'utiliser l'élément `` dans le HTML. Une autre façon est d'utiliser la propriété `background` ou `background-image` dans le CSS pour assigner une image d'arrière-plan à un élément. L'une ou l'autre des deux options fonctionnent.

Cependant, elles ont chacune des cas d'utilisation spécifiques.

L'élément `` dans le HTML est l'option préférée lorsque l'image utilisée détient une valeur sémantique et que son contenu est pertinent pour le contenu de la page.

La propriété `background` ou `background-image` dans le CSS est l'option préférée lorsque l'image utilisée fait partie du design ou de l'interface utilisateur de la page.

En tant que tel, il n'est pas directement pertinent pour le contenu de la page.

L'élément `` est très populaire, et quand il a été ajouté à la spécification HTML, il a changé à jamais la façon dont les sites Web ont été construits.

En pratique

Maintenant que nous savons comment ajouter et positionner des images sur une page, nous allons jeter un oeil à notre site Web et voir où nous pouvons ajouter quelques images.

Commençons en ajoutant quelques images à notre page d'accueil. Précisément, nous allons ajouter une image dans chacune des sections qui mettent en avant quelques unes de nos pages.

Avant de sauter dans le code, cependant, nous allons créer un nouveau dossier nommé "img" au sein de notre dossier "assets". Puis, dans le dossier "img", nous allons créer un autre dossier nommé "home" spécifiquement pour les images de notre page d'accueil. Dans le dossier

"home" nous allons ajouter trois images: `quoi.jpg`, `quand.jpg` et `ou.jpg`.

Ensuite, dans notre fichier `index.html`, chaque section intro comporte un élément `<a>` enveloppant à la fois un élément `<h3>` et un élément `<h5>`. Déplacez l'élément `<h5>` au-dessus de l'élément `<a>` et remplacez-le par un élément ``. La valeur de l'attribut `src` pour chaque élément `` correspondra à la structure du dossier et au nom du fichier que nous avons mis en place, et la valeur de l'attribut `alt` décrira le contenu de chaque image ainsi que l'attribut `title`.

Le HTML pour notre première intro, pour la page Programme, ressemblera à ceci:

```
1 <section class="intro col-1-3">
2   <h5>Au programme...</h5>
3   <a href="programme.html">
4     
6     <h3>...des kilomètres de code</h3>
7   </a>
8   <p>Nous sommes heureux d'accueillir plus de vingt
9 orateurs qui présenteront les dernières technologies.
10 Préparez vous à un spectacle de qualité. </p>
11 </p>
12 </section>
```

Continuons ce modèle à la fois pour les présentations de la page de l'horaire -quand.jpg- et du lieu -ou.jpg-.

Maintenant que nous avons ajouté quelques images à notre page d'accueil, nous devons nettoyer un peu leurs styles et nous assurer qu'ils correspondent bien à la mise en page de notre page.

Puisque les images sont des éléments en ligne par défaut, modifions nos images dans les sections d'info pour les éléments au niveau des blocs. Définissons également leur largeur maximale à 100% pour s'assurer qu'ils ne dépassent pas la largeur de leurs colonnes respectives. La modification de cette valeur de largeur est importante car elle permet à nos images de s'adapter à la largeur des colonnes selon les besoins.

Enfin, redisons les angles des images légèrement et appliquons 22 pixels de marge inférieure aux images, fournissant un peu de respiration.

Une fois que nous ajoutons ces nouveaux styles à nos styles de page d'accueil existants (en utilisant la classe `intro` comme sélecteur de sélection pour les éléments ``), notre CSS ressemblera à ceci:

```
1 .intro img{
2   border-radius: 5px;
3   display: block;
4   margin-bottom: 22px;
5   max-width: 100%
6 }
```

Ensuite, ajoutons des images à la page Programme. Nous allons commencer par créer un dossier "programme" dans notre dossier "img" et placer les images de tous les intervenants.

Dans le fichier programme.html, ajoutons un élément `` dans chacun des éléments `<aside>` de l'information sur l'intervenant. Placez chaque élément `` dans l'élément `<div>` avec la classe "prof", juste au-dessus de l'élément ``.

La valeur de l'attribut `src` de chaque image correspondra au dossier "programme" que nous avons mis en place et au nom de l'intervenant. La valeur de l'attribut `alt` sera le nom de l'intervenant.

Le `<aside>` pour le premier conférencier, ressemblera à ceci:

```
1 <aside class="col-1-3">
2   <div class="prof">
3     
5     <ul>
6       <li>
7         <a href="https://twitter.com/prof1">@prof1</a>
8       </li>
9       <li>
10        <a href="http://www.prof1.com/">prof1.com</a>
11      </li>
12    </ul>
```

```
13 </div>
    </aside>
```

Comme nous l'avons fait avec les images sur notre page d'accueil, nous allons vouloir appliquer certains styles aux images sur la page "programme".

Commençons par appliquer la propriété `border-radius` avec une valeur de 50%, transformant nos images en cercles. À partir de là, fixons une hauteur fixe de 130 pixels à chaque image et les mettons à aligner verticalement au sommet de la ligne dans laquelle ils se trouvent.

Avec la hauteur et l'alignement vertical en place, appliquons des marges verticales aux images. En utilisant une marge négative de 66 pixels sur le haut des images, nous les tirons légèrement de l'élément `<aside>` et centrons verticalement avec le bord supérieur de l'élément `<div>` avec une valeur de classe "prof-info". Ensuite, l'application d'une marge de 22 pixels sur le bas de l'image fournit un espace entre l'image et l'élément `` dessous.

Lorsque nous ajoutons ces nouveaux styles à nos styles existants pour la page "programme" (en utilisant la classe "prof-info" comme sélecteur de sélection pour les éléments ``), notre CSS ressemblera à ceci:

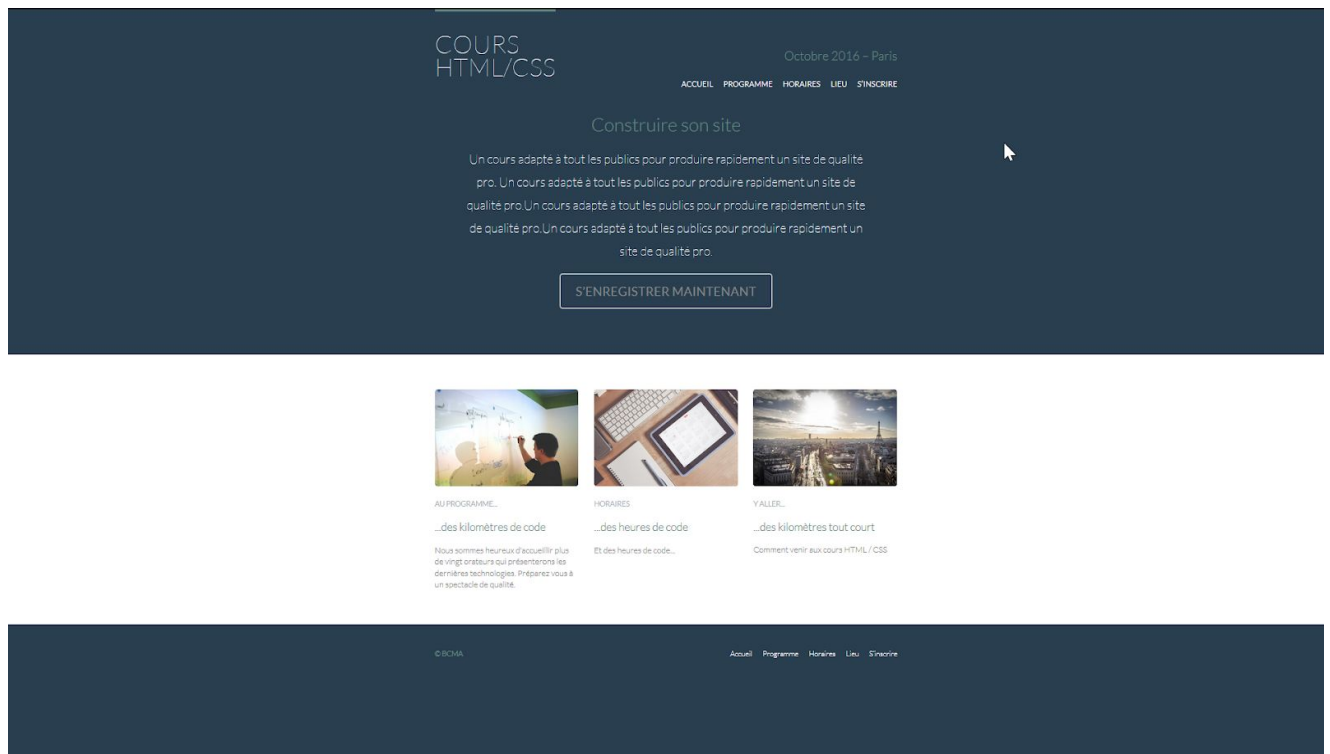
```
1 .prof-info img {
2   border-radius: 50%;
3   height: 130px;
4   margin: -66px 0 22px 0;
5   vertical-align: top;
6 }
```

Puisque nous utilisons une marge négative sur l'élément `` dans l'élément `<div>` avec une valeur de classe "profinfo", nous devons supprimer le padding sur le dessus de l'élément `<div>`.

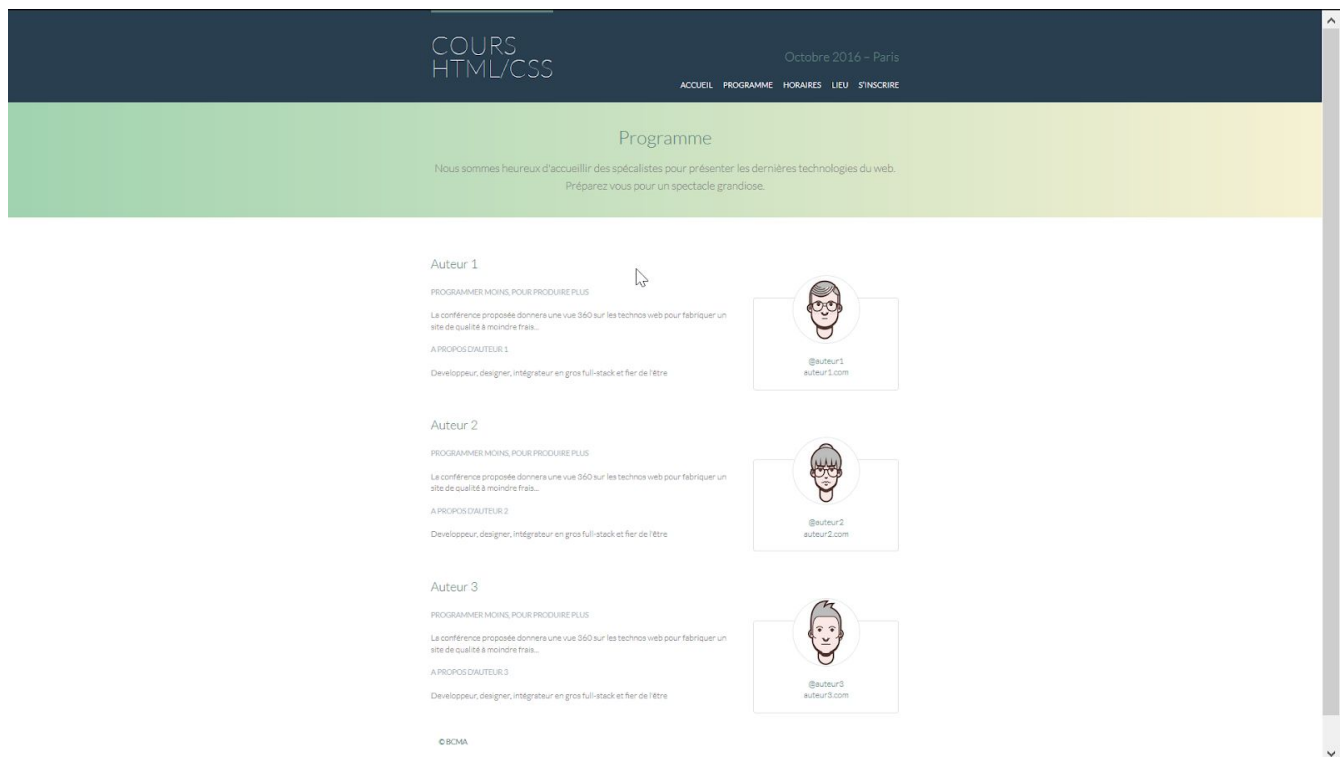
Auparavant, nous utilisions la propriété `padding` avec une valeur de 22px 0, plaçant ainsi 22 pixels de padding en haut et en bas et 0 pixels de padding à gauche et à droite de l'élément `<div>`. Nous allons changer cette propriété et sa valeur pour la propriété `padding-bottom`, car c'est le seul padding que nous devons identifier, et utiliser une valeur de 22 pixels.

Le nouvel ensemble de règles de classe "prof-info" ressemblera à ceci:

```
1 .prof-info{  
2   border: 1px solid #dfe2e5;  
3   border-radius: 5px;  
4   margin-top: 88px;  
5   padding-bottom: 22px;  
6   text-align: center;  
7 }
```



Notre page d'accueil après l'ajout des images à chaque section



Notre page Programme après l'ajout des images

Ajouter du son

HTML5 fournit un moyen rapide et facile d'ajouter des fichiers audio à un site Web par l'intermédiaire de l'élément `<audio>`. Comme avec l'élément ``, l'élément `<audio>` accepte une URL source spécifiée dans l'attribut `src`. Contrairement à l'élément ``, l'élément `<audio>` nécessite à la fois des balises d'ouverture et de fermeture.

```
1 <audio src="jazz.ogg"></audio>
```

Attributs audio

Plusieurs attributs peuvent accompagner l'attribut `src` sur l'élément `<audio>`. Les plus populaires sont `autoplay`, `controls`, `loop` et `preload`.

Les attributs `autoplay`, `controls` et `loop` sont tous des attributs booléens -vrai ou faux-. Comme attributs booléens, ils ne nécessitent pas une valeur indiquée. Au lieu de cela, quand chacun est présent sur l'élément `<audio>` sa valeur sera définie sur vrai, et l'élément `<audio>` se comportera en conséquence.

Par défaut, l'élément `<audio>` n'est pas affiché sur une page. Si l'attribut booléen `autoplay` est présent sur l'élément `<audio>`, rien n'apparaîtra sur la page, mais le fichier audio sera automatiquement lu lors du chargement.

```
1 <audio src="jazz.ogg" autoplay></audio>
```

Pour afficher l'élément `<audio>` sur une page, l'attribut booléen `control` est nécessaire. Quand il est appliqué à `<audio>`, l'attribut booléen `control` affiche les commandes audio par défaut d'un navigateur, y compris play et pause, rechercher, et la commande de volume. Noter que l'aspect du lecteur audio dépend du navigateur.

```
1 <audio src="jazz.ogg" controls></audio>
```



Lorsqu'il est présent sur l'élément `<audio>`, l'attribut booléen `loop` provoque la répétition continue d'un fichier audio, du début à la fin.

Enfin, l'attribut `preload` de l'élément `<audio>` permet d'identifier les informations éventuelles sur le fichier audio à charger avant la lecture du clip. Il accepte trois valeurs: `none`, `auto` et `metadata`. La valeur `none` ne préchargera aucune information sur le fichier audio, alors que la valeur `auto` préchargera toutes les informations sur le fichier audio. La valeur `metadata` se situe entre les valeurs `none` et `auto`, car elle préchargera toutes les informations de métadonnées disponibles sur le fichier audio, comme par exemple la longueur du clip.

Lorsque l'attribut `preload` n'est pas présent sur l'élément `<audio>`, toutes les informations sur un fichier audio sont chargées, comme si la valeur était définie à `auto`. Pour cette raison, l'utilisation de l'attribut `preload` avec la valeur `metadata` ou `none` est une bonne idée lorsqu'un fichier audio n'est pas essentiel à une page. Cela aidera à économiser la bande passante et permettra aux pages de se charger plus rapidement.

Solutions de sources multiples et alternatives

À l'heure actuelle, tous les navigateurs ne supportent pas les mêmes formats de fichiers audio, dont les trois plus populaires sont ogg, mp3 et

wav. Pour obtenir le meilleur support par les navigateurs, nous devons utiliser plusieurs sources audio qui seront inclus dans les balises d'ouverture et de fermeture d'un élément `<audio>`.

Pour commencer, nous supprimerons l'attribut `src` de l'élément `<audio>`. Au lieu de cela, nous utiliserons l'élément `<source>`, avec un attribut `src`, imbriqué dans l'élément `<audio>` pour définir une nouvelle source.

En utilisant un élément `<source>` et un attribut `src` pour chaque format de fichier, nous pouvons déclarer les fichiers audio de formats différents les uns après les autres. Nous utiliserons l'attribut `type` pour que le navigateur identifie les types audio disponibles. Lorsqu'un navigateur reconnaît un format de fichier audio, il charge ce fichier et ignore tous les autres.

Comme il a été introduit avec la norme HTML5, certains navigateurs peuvent ne pas prendre en charge l'élément `<audio>`. Dans ce cas, nous pouvons fournir un lien supplémentaire pour télécharger le fichier audio après tous les éléments `<source>` dans l'élément `<audio>`.

```
1 <audio controls>
2   <source src="jazz.ogg" type="audio/ogg">
3   <source src="jazz.mp3" type="audio/mpeg">
4   <source src="jazz.wav" type="audio/wav">
5   <a href="jazz.mp3">Télécharger</a> le fichier audio.
6 </audio>
```

Dans le code ci-dessus, l'élément `<audio>` inclut l'attribut booléen `controls` pour s'assurer que le lecteur audio soit affiché dans les navigateurs qui prennent en charge l'élément `<audio>`. L'élément `<audio>` n'inclut pas d'attribut `src` mais enveloppe trois éléments `<source>` différents. Chaque élément `<source>` comprend un attribut `src` qui fait référence à un format de fichier audio différent et un attribut `type` qui identifie le format du fichier audio. La dernière alternative, ajoutée si jamais un navigateur ne reconnaît aucun des formats de fichier audio, est le lien d'ancrage pour télécharger l'élément qui sera affiché.

En plus de l'élément `<audio>`, HTML5 a également introduit l'élément `<video>`, qui partage quelques similitudes avec l'élément `<audio>`.

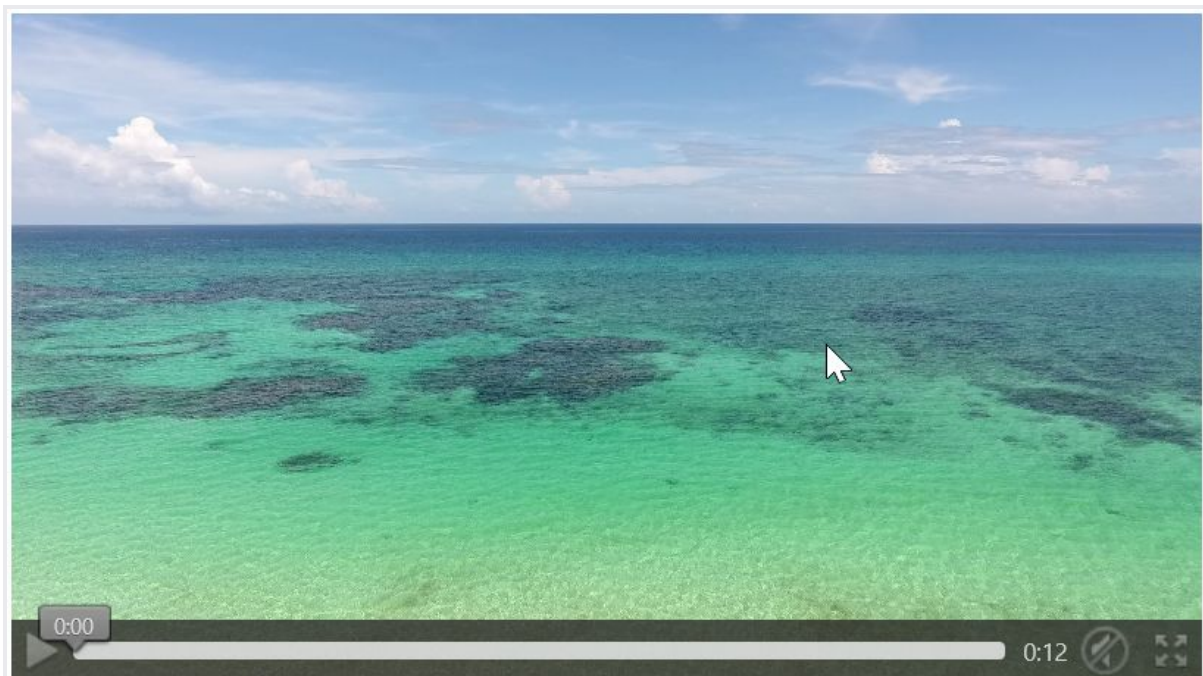
Ajouter une vidéo

L'ajout de vidéo en HTML5 est très similaire à l'ajout d'audio. Nous utilisons l'élément `<video>` à la place de l'élément `<audio>`. Tous les mêmes attributs (`src`, `autoplay`, `controls`, `loop`, et `preload`) et les alternatives s'appliquent aussi.

Avec l'élément `<audio>`, si l'attribut `controls` n'est pas spécifié, le clip audio n'est pas affiché. Avec les vidéos, si l'attribut `controls` n'est pas spécifié, la vidéo s'affiche. Toutefois, il est assez difficile à afficher à moins que l'attribut `autoplay` soit également appliqué. En général, la meilleure pratique ici est d'inclure l'attribut `controls` à moins qu'il y ait une bonne raison de ne pas permettre aux utilisateurs de démarrer, d'arrêter ou de rejouer la vidéo.

Puisque les vidéos prennent de l'espace sur la page, il est intéressant de préciser leurs dimensions, ce qui est le plus souvent fait avec des propriétés de largeur et de hauteur en CSS. Cela permet de s'assurer que la vidéo n'est pas trop grande et reste dans la page. En outre, la spécification d'une taille, comme avec les images, aide le navigateur à afficher les vidéos plus rapidement et lui permet d'allouer l'espace approprié nécessaire pour la vidéo sur la page.

```
1 <video src="horizon.mp4" controls></video>
```



Démo d'ajout de vidéo

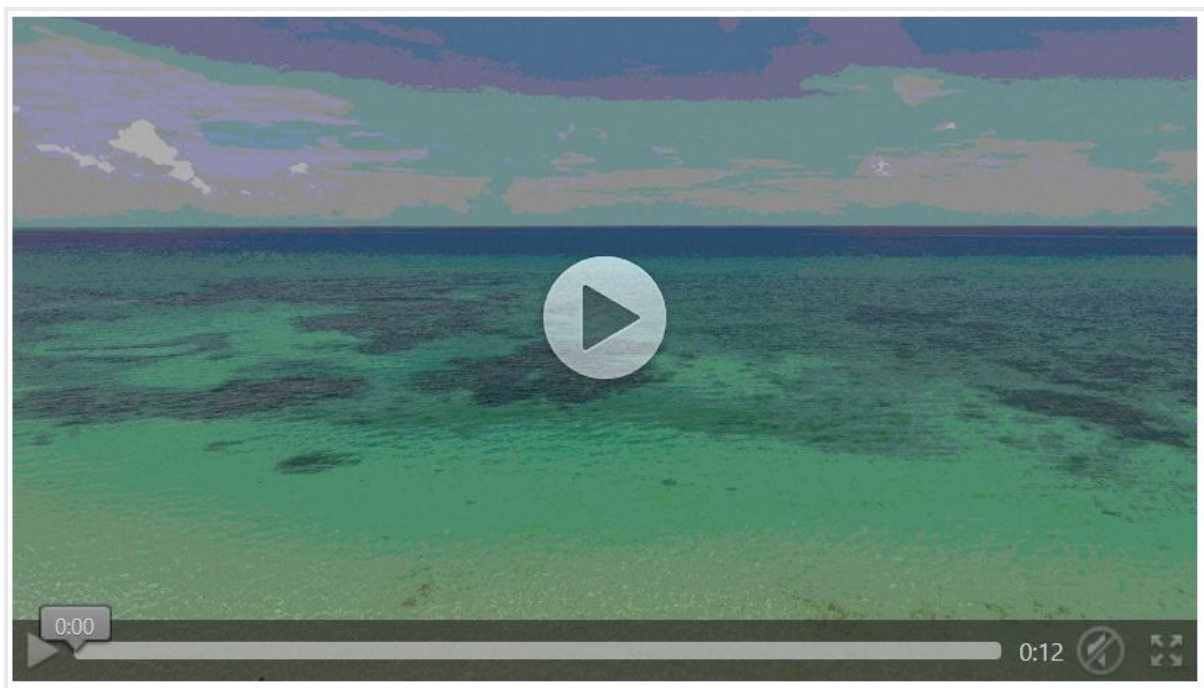
Personnalisation Audio & Video Controls

Par défaut, les éléments `<audio>` et `<video>` sont déterminés par chaque navigateur indépendamment. En fonction de la conception d'un site Web, si l'aspect du lecteur multimédia doivent respecter de façon obligatoire une charte il peut être nécessaire de prévoir un lecteur personnalisé mais il faudra alors un peu de JavaScript pour le faire fonctionner.

L'attribut poster

Un attribut supplémentaire disponible pour l'élément `<video>` est l'attribut `poster`. L'attribut `poster` permet de spécifier une image, sous la forme d'une URL, devant être affichée avant la lecture de la vidéo. L'exemple ci-dessous utilise une capture d'écran de la vidéo comme affiche pour la vidéo "Horizon".

```
1 <video src="horizon.mp4" controls  
2 poster="horizon.jpg"></video>
```



Vidéo alternatives

Comme avec l'élément `<audio>`, les retours vidéo sont également nécessaires. Le même format de balisage, avec plusieurs éléments `<source>` pour chaque type de fichier et un lien de téléchargement, peuvent s'appliquer également à l'élément `<video>`.

```
1 <video controls>
2   <source src="horizon.ogv" type="video/ogv">
3   <source src="horizon.mp4" type="video/mp4">
4   S'il vous plaît <a href="horizon.mp4">télécharger</a> la
5   vidéo.
6 </video>
```

Une autre option de secours qui pourrait être utilisée à la place d'une alternative de téléchargement est d'utiliser une vidéo intégrée YouTube, DailyMotion ou Vimeo. Ces sites d'hébergement vidéo nous permettent de télécharger nos vidéos, de fournir un lecteur vidéo standard et de nous permettre d'intégrer nos vidéos sur une page à l'aide d'un cadre en ligne.

HTML5 audio et formats de fichier vidéo

Le support du navigateur pour les éléments `<audio>` et `<video>` varie, de même que les formats de fichier requis avec ces éléments. Chaque navigateur a ses propres formats audio et vidéo préférés.

Il y a quelques outils qui aident à convertir un fichier [audio](#) ou [vidéo](#) dans différents formats, et une recherche rapide fournira une abondance d'options.

En pratique

Pour animer un peu notre page d'accueil nous allons ajouter une nouvelle section en dessous des présentations pour y mettre une superbe vidéo.

Sous notre élément `<div>` de classe "grid", ajoutons un nouvel élément `<div>` avec la valeur d'attribut de classe "grid". A la classe de grille qui centre notre contenu sur la page on va ajouter une classe "col-mid" pour donner une taille moitié moins grande à notre contenu.

Jusqu'à présent, la section principale de notre fichier "index.html" ressemble à ceci:

```
1 <section class="row">
2   <div class="grid">
3     ...
4   </div>
5   <div class="grid col-mid">
6     ...
7   </div>
8 </section>
```

Dans le nouvel élément `<div>` avec la classe "grid", nous ajoutons notre nouvel élément `<video>` qui aura les attributs `autoplay`, `controls` et `poster` et des éléments `<source>` pour les différents formats vidéos fournis. Notre HTML devrait maintenant ressembler à ceci:

```
1 <div class="grid col-mid">
2   <video autoplay controls poster="video/white-board.jpg">
3     <source src="video/white-board.ogv" type="video/ogg">
4     <source src="video/white-board.mp4" type="video/mp4">
5     <source src="video/white-board.webm" type="video/webm">
6     S'il vous plaît <a href="white-board.mp4">télécharger</a>
7     la vidéo.
8   </video>
9 </div>
```

Maintenant que nous devons ajouter notre classe "col-mid" dans notre fichier `style.css`. Nous allons la taille à la moitié de la taille de la classe "grid".

Ensuite, nous allons règle css pour que l'élément `<video>` prenne la totalité de la largeur définie par la classe "col-mid".

Le fichier `style.css` ressemble à ce qui suit:

```
1 /*
2  =====
3  Grid
4  =====
5  */
```

```
5 ...  
6  
7 .col-mid{  
8     width: 480px;  
9 }  
10  
11 .col-mid video{  
12     width: 100%;  
13 }
```

Ajouter des contenus par iframes

Une autre façon d'ajouter du contenu à une page consiste à intégrer une autre page HTML dans la page en cours. Cela se fait à l'aide d'un cadre en ligne ou d'un élément `<iframe>`. L'élément `<iframe>` accepte l'URL d'une autre page HTML dans la valeur d'attribut `src`. Cela va importer le contenu de la page HTML incorporée à afficher sur la page en cours. La valeur de l'attribut `src` peut être une URL relative à la page sur laquelle l'élément `<iframe>` apparaît ou une URL absolue pour une page entièrement externe.

De nombreuses pages utilisent l'élément `<iframe>` pour intégrer des médias dans une page à partir d'un site Web externe tel que Google Maps, YouTube et autres.

```
1 <iframe src="https://www.google.com/maps/embed"></ iframe>
```

L'élément `<iframe>` a quelques styles par défaut, y compris une bordure, une largeur et une hauteur. Ces styles peuvent être ajustés en utilisant les attributs HTML `frameborder`, `width` et `height` ou en utilisant les propriétés CSS de `border`, `width` et `height`.

Les pages référencées dans l'attribut `src` d'un élément `<iframe>` s'affichent selon leurs propres règles, car elles n'héritent pas de styles ou de comportements à partir de la page sur laquelle elles sont référencées. Tous les styles appliqués à une page incluant un élément `<iframe>` ne seront pas hérités par la page référencée dans l'élément `<iframe>`. En outre, les liens au sein de la page référencée dans l'élément `<iframe>` s'ouvrira à l'intérieur de ce cadre, laissant la page qui contient l'élément `<iframe>` inchangée.

En pratique

Les iframes fournissent un excellent moyen d'ajouter du contenu dynamique à une page. Nous allons donner faire cet exercice en ajoutant quelques cartes à notre page "Carte" .

Avant d'ajouter des cartes préparons d'abord notre page "Carte" pour une grille à deux colonnes. Sous la section principale de la page, nous ajouterons un élément `<section>` avec la valeur d'attribut class "row" pour identifier une nouvelle section de la page, et nous allons inclure des styles généraux, comme un fond blanc et un padding vertical .

Directement à l'intérieur de cet élément `<section>`, ajoutons un élément `<div>` avec la valeur d'attribut de classe "grid". La classe "grid" centre notre contenu sur la page et prépare les colonnes du tiers et des deux tiers à suivre.

Jusqu'à présent, la section principale de notre fichier "carte.html" ressemble à ceci:

```
1 <section class="row">
2   <div class="grid">
3     ...
4   </div>
5 </section>
```

Dans l'élément `<div>` avec la classe "grid", nous aurons deux nouvelles sections, l'une pour le lieu du premier cours et l'autre pour le lieu du second cours. Ajoutons deux nouveaux éléments `<section>` et donnons à chacun de ces éléments `<section>` une classe unique correspondant à son contenu. Nous allons utiliser ces classes pour ajouter des marges au bas de chaque section.

Notre HTML devrait maintenant ressembler à ceci:

```
1 <section class="row">
2   <div class="grid">
3     <section class="lieu-cours">
4       ...
5     </section>
6     <section class="lieu-cours-fin">
```

```

7      ...
8      </section>
9  </div>
10 </section>

```

Maintenant que nous avons quelques classes à travailler, nous allons créer une nouvelle section dans notre fichier style.css pour les styles pour la page "Lieu". Nous allons ajouter une marge de 66 pixels au bas de l'élément `<section>` avec la valeur d'attribut de classe "lieu-cours" pour insérer un espace entre elle et l'élément `<section>` ci-dessous.

Ensuite, nous allons ajouter une marge de 22 pixels au bas de l'élément `<section>` avec la classe "lieu-cours" pour fournir un espace entre elle et l'élément `<footer>` ci-dessous.

La nouvelle section du site dans le fichier style.css ressemble à ce qui suit:

```

1  /*
2   =====
3   Lieu
4   =====
5  */
6
7  .lieu-cours{
8      margin-bottom: 66px;
9  }
10 .lieu-cours-fin{
11     margin-bottom: 22px;

```

L'élément `<section>` avec la valeur d'attribut de classe "lieu-cours" a une marge inférieure inférieure à celle de l'élément `<section>` avec la valeur d'attribut de classe "lieu-cours" parce qu'elle se trouve à côté du padding du bas de l'élément `<section>` Avec l'attribut classe "row" l'ajout de cette marge et le padding ensemble nous donne la même valeur que la marge inférieure sur l'élément `<section>` avec la valeur d'attribut de classe "lieu-cours".

Il est maintenant temps de créer les deux colonnes dans chacun des nouveaux éléments `<section>`. Nous allons commencer par ajouter un élément `<div>` avec une valeur d'attribut de classe "col-1-3" pour établir

une colonne d'un tiers. Après cela, nous ajouterons un élément `<iframe>` avec une valeur d'attribut de classe "col-2-3" pour établir une colonne de deux tiers.

En gardant à l'esprit que les classes de colonne font à la fois les éléments `<div>` et `<iframe>` éléments en ligne, nous devons supprimer l'espace vide qui apparaîtra entre eux. Pour ce faire, nous ouvrirons un commentaire HTML directement après la balise `<div>` de fermeture, et nous fermerons le commentaire HTML immédiatement avant la balise `<iframe>` d'ouverture.

Dans l'ensemble, notre HTML pour les colonnes ressemblera à ceci:

```
1 <section class="row">
2   <div class="grid">
3     <section class="lieu-cours">
4       <div class="col-1-3"></div><!--
5       --><iframe class="col-2-3"></iframe>
6     </section>
7     <section class="lieu-cours-fin">
8       <div class="col-1-3"></div><!--
9       --><iframe class="col-2-3"></iframe>
10    </section>
11  </div>
12</section>
```

Dans chacun des éléments `<div>` avec une valeur d'attribut de classe "col-1-3", ajoutez le nom du lieu dans un élément `<h2>`, suivi de deux éléments `<p>`. Ajoutons un attribut `id` au titre `h2` qui identifiera uniquement chaque lieu, ici avec la valeur "carrel". Dans le premier `<p>` élément, nous allons inclure l'adresse du lieu, et dans le second élément `<p>`, nous allons inclure le site Web du lieu (dans un lien d'ancrage) et le numéro de téléphone.

À l'intérieur de chacun des paragraphes, utilisons l'élément retour à la ligne, `
`, pour placer des pauses dans l'adresse et entre le site Web et le numéro de téléphone.

Pour l'élément `<section>` avec la classe "lieu-cours", le HTML ressemble à ceci:

```

1 <section class="lieu-cours">
2   <div class="col-1-3">
3     <h2 id="carrel">Lycée Armand Carrel</h2>
4     <p>45 rue Armand Carrel<br> 75019 Paris</p>
5     <p> <a
6 href="https://www.ac-paris.fr/serail/jcms/p2_73320/accueil
7 " >Le site du lycée</a> <br> 01.44.52.57.00</p>
8   </div><!--
9   --><iframe class="col-2-3" src=""></iframe>
10 </section>
11

```

Le même schéma montré ici pour le cours Armand Carrel devrait également être appliqué celui de l'école élémentaire 30 rue Jeanne d'Arc 75013 Paris (en employant, bien sûr, l'adresse appropriée, le site Web, et le numéro de téléphone) avec un attribut id sur le titre avec la valeur "darc".

Nous devons rechercher ces adresses dans Google Maps. Une fois que nous avons localisé une adresse et créé une carte personnalisée, nous avons la possibilité d'intégrer cette carte dans notre page. En suivant les instructions de Google Maps il faut simplement partager l'adresse et récupérer dans l'onglet intégrer code HTML `<iframe>` fournis et l'intégrer dans notre page.

En copiant l'élément `<iframe>` de Google Maps, nous devons nous assurer que nous conservons l'attribut de classe "col-2-3" de notre élément `<iframe>`. Nous devons également veiller à ne pas nuire au commentaire HTML qui se ferme directement avant notre balise `<iframe>` d'ouverture. Nous supprimerons aussi tous les attributs autre que `src` qui sont par défaut dans la balise `<iframe>`.

Si vous regardez directement l'élément `<section>` avec la valeur de l'attribut de classe "lieu-cours", le code HTML ressemble à ceci:

```

1 <section class="lieu-cours">
2   <div class="col-1-3">
3     <h2 id="carrel">Lycée Armand Carrel</h2>
4     <p>45 rue Armand Carrel<br> 75019 Paris</p>
5     <p> <a
6 href="https://www.ac-paris.fr/serail/jcms/p2_73320/accueil

```

```

7  " >Le site du lycée</a> <br> 01.44.52.57.00</p>
8  </div><!--
9  --><iframe class="col-2-3"
10 src="https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d
11 10494.863564438023!2d2.3759748!3d48.8826924!3m2!1i1024!2i7
12 68!4f13.1!3m3!1m2!1s0x0%3A0x4fe205ac704a216b!2sLyc%C3%A9e+
13 Professionnel+Armand+Carrel!5e0!3m2!1sfr!2sfr!4v1482159789
14 509" ></iframe>
    </section>

```

Enfin, nous voulons nous assurer que les deux éléments `<iframe>` qui font référence à Google Maps partagent la même hauteur. Pour ce faire, nous créerons une nouvelle classe "lieu-loc" et nous l'appliquerons à chacun des éléments `<iframe>` à côté de la valeur d'attribut de classe "col-2-3" existante.

Le HTML pour l'élément `<section>` avec la valeur d'attribut de classe "lieu-cours" ressemble maintenant à ceci:

```

1  <section class="lieu-cours">
2    <div class="col-1-3">
3      <h2 id="carrel">Lycée Armand Carrel</h2>
4      <p>45 rue Armand Carrel<br> 75019 Paris</p>
5      <p> <a
6 href="https://www.ac-paris.fr/serail/jcms/p2_73320/accueil
7 ">Le site du lycée</a><br> 01.44.52.57.00</p>
8    </div><!--
9    --><iframe class="lieu-loc col-2-3" src
10 ="https://www.google.com/maps/embed?pb=!1m5!3m3!1m2!1s0x88
11 0e2ca55810a493%3A0x4700ddf60fcbfad6!2schicago+theatre!5e0!
12 3m2!1sen!2sus!4v1388701393606"></iframe>
13 </section>

```

Une fois la classe lieu-loc appliquée à chaque élément `<iframe>`, créons la règle de classe "lieu-loc" dans notre fichier style.css. Il inclut la propriété hauteur avec une valeur de 264 pixels.

Le jeu de règles de la classe "lieu-loc" ressemble à ceci:

```

1  .lieu-loc{ height: 264px; }

```


Nous avons maintenant une page de site, avec des cartes pour les différents endroits de notre cours.

COURS HTML/CSS

Octobre 2016 – Paris

[ACCUEIL](#) [PROGRAMME](#) [HORAIRE](#) [LIEU](#) [S'INSCRIRE](#)


Les Lieux

Les lieux de formation sont les suivants

Lycée Armand Carrel

45 rue Armand Carrel
75019 Paris


[Le site du lycée](#)
01.44.52.57.00

A Google Map showing the location of Lycée Professionnel Armand Carrel in Paris. The map is centered on the 10th arrondissement, near the Porte de la Chapelle. A red pin marks the school's location. Surrounding areas include Clignancourt, Sacré-Cœur, and various other schools like Lycée Jules Ferry and Lycée Jules Richard. The map includes street names, arrondissement numbers (9E ARR., 10E ARR.), and public transport icons.

Ecole Jeanne d'Arc

30 Place Jeanne d'Arc
75013 Paris

[Informations sur l'école](#)
01.45.83.49.12

A Google Map showing the location of 30 Place Jeanne d'Arc in Paris. The map is centered on the 13th arrondissement, near the Quartier de la Gare. A red pin marks the location. Surrounding areas include Bibliothèque Nationale de France, Decathlon Paris Rive Gauche, and various streets like Avenue de Choisy and Avenue Edouard. The map includes arrondissement numbers (13E ARR.) and public transport icons.

© BCMA

[Accueil](#) [Programme](#) [Horaire](#) [Lieu](#) [S'inscrire](#)

Identifier sémantiquement figures et légendes

Avec HTML5 est également venue l'introduction des éléments `<figure>` et `<figcaption>`. Ces éléments ont été créés pour marquer sémantiquement un contenu autonome ou des médias, généralement avec une légende. Avant HTML5 cela était souvent fait en utilisant une liste ordonnée. Alors qu'une liste ordonnée fonctionnait, le balisage n'était pas sémantiquement correct.

Figure

L'élément `<figure>` au niveau des blocs permet d'identifier et d'intégrer un contenu autonome, souvent des "médias" au sens large. Il peut entourer des images, des clips audio, des vidéos, des blocs de code, des diagrammes, des illustrations ou tout autre support autonome. Plus d'un élément de contenu autonome, comme plusieurs images ou vidéos, peuvent être contenu dans l'élément `<figure>` à la fois.

```
1 <figure>  
2     
3 </figure>
```



Figure et Légende

Pour ajouter une légende à l'élément `<figure>`, l'élément `<figcaption>` est utilisé. La `<figcaption>` peut apparaître en haut, en bas ou n'importe où dans l'élément `<figure>`. Cependant, il ne peut apparaître qu'une seule fois. Lorsqu'il est utilisé, l'élément `<figcaption>` servira de légende pour tout le contenu de l'élément `<figure>`.

De plus, l'élément `<figcaption>` peut remplacer l'attribut `alt` d'un élément `` si le contenu de l'élément `<figcaption>` fournit une description utile du contenu visuel de l'image.

```
1 <figure>
2   
3   <figcaption>Un Koala dans les arbres !</figcaption>
4 </figure>
```



Un koala dans les arbres

Démo légende de la balise figure

