

# Diaporama HTML5/CSS3/JavaScript



Page en cours d'écriture

## Besoin

Le besoin exprimé est de créer un diaporama en HTML5/CSS3/JavaScript avec les caractéristiques suivantes :

- Pour chaque diapo, un titre et une description s'affichent.
- Il peut y avoir un fichier audio opus associé à la diapo. Si il existe un fichier audio, le lire et passer à la diapo suivante à la fin de la lecture du fichier audio sinon passer à la diapo suivante au bout de 3 secondes.
- Il faudra également afficher un titre pour le diaporama, une liste de toutes les diapos disponibles avec leur titre.
- L'affichage du titre de la diapo en cours et de son descriptif ne s'affichera pas sur la diapo ainsi que le contrôleur audio.
- Les scripts et css complémentaires seront dans des fichiers séparés.
- La liste des diapo se trouve dans un fichier json.
- Il peut y avoir jusqu'à 200 photos.

Le diaporama doit avoir un en-tête avec le titre du diaporama, une image qui prend toute la largeur à 80%, la liste des diapositives sur le côté droit, le titre et la description de la diapositive en cours en dessous de l'image et de la liste. L'utilisateur doit pouvoir cliquer sur un élément de la liste des diapositives pour passer directement à cette diapositive.

## Analyse

Pour créer un diaporama en HTML5/CSS3/JavaScript, voici les grandes étapes :

- Créer une page HTML pour votre diaporama. Cette page contiendra un conteneur pour afficher les diapositives, un titre pour le diaporama, une liste de toutes les diapos avec leurs titres, et éventuellement un contrôleur audio.
- Écrire le CSS pour styliser votre page. Vous pouvez utiliser des frameworks tels que Bootstrap ou Materialize pour faciliter le processus.
- Écrire le code JavaScript pour récupérer les données des diapositives à partir d'un fichier JSON et les afficher dans le diaporama. Vous pouvez utiliser la méthode Fetch pour récupérer les données du fichier JSON.

- Ajouter un événement pour passer à la diapositive suivante après la fin de la lecture de l'audio, ou après 3 secondes si aucun fichier audio n'est associé à la diapositive.
- Pour masquer le titre de la diapo en cours et son descriptif sur la diapo, vous pouvez ajouter une classe CSS pour cacher ces éléments.
- Pour le contrôleur audio, vous pouvez utiliser la balise audio HTML5 et ajouter des événements pour démarrer et arrêter la lecture.

## Principe de fonctionnement

### Afficher et masquer les diapos

Dans le code CSS, chaque diapositive est définie comme un élément avec la classe `.slide`. Toutes les diapositives sont initialement masquées en utilisant la propriété CSS `display: none`. Ensuite, la diapositive en cours est affichée en utilisant la classe `.current`, qui a la propriété CSS `display: block`. De cette façon, une seule diapositive est affichée à la fois.

```
/* Diapositives */
.slide {
  ..
  display: none;
}

.slide.current {
  ..
  display: block;
}
```

Lorsque l'utilisateur clique sur un élément de la liste des diapositives, le script JavaScript met à jour la diapositive en cours en cachant la diapositive précédente et en affichant la nouvelle diapositive. Cela se fait en ajoutant et en supprimant la classe `.current` sur les éléments correspondants.

La propriété CSS `display` permet de définir le type d'affichage d'un élément. La valeur `block` est l'un des types d'affichage les plus courants en HTML. Elle permet de créer un élément en tant que bloc qui occupe toute la largeur de son conteneur et qui commence sur une nouvelle ligne.

Lorsque vous utilisez `display: block` sur un élément, cela signifie qu'il sera affiché sous la forme d'un bloc rectangulaire qui occupe toute la largeur de son conteneur. Cela peut être utile pour afficher des éléments tels que des paragraphes, des titres, des boutons, des images, des divs, etc.

Dans le cas du diaporama, l'utilisation de `display: block` permet d'afficher la diapositive en cours en tant que bloc rectangulaire à l'intérieur du conteneur de diaporama. Cela permet de faire en sorte que la diapositive remplisse tout l'espace disponible et soit parfaitement visible pour l'utilisateur.

```
// Récupérer la diapositive actuelle et la cacher
const currentSlide = document.querySelector('.slide.current');
if (currentSlide) {
```

```
currentSlide.classList.remove('current');
}

// Afficher la nouvelle diapositive
const newSlide = document.querySelectorAll('.slide')[slideIndex];
newSlide.classList.add('current');
```

## Modifier le code HTML avec JavaScript

Le code JavaScript est utilisé pour ajouter, supprimer, modifier ou réorganiser les éléments HTML et les propriétés de ces éléments, afin de créer des pages web interactives et dynamiques.

Le code HTML est modifié par JavaScript en utilisant le Document Object Model (DOM). Le DOM est une représentation hiérarchique des éléments HTML d'une page web, qui peut être manipulée en utilisant des scripts JavaScript.

Lorsque le script JavaScript est exécuté, il peut accéder aux éléments HTML existants sur la page web en utilisant des sélecteurs de document, tels que `getElementById`, `getElementsByClassName`, `querySelector`, etc. Il peut également créer de nouveaux éléments HTML en utilisant des méthodes telles que `createElement`, `appendChild`, `insertBefore`, etc.

Le script JavaScript peut ensuite modifier les propriétés des éléments HTML, telles que le contenu du texte, les attributs, les styles CSS, etc. Il peut également ajouter des écouteurs d'événements pour répondre aux interactions de l'utilisateur, tels que les clics de souris, les saisies clavier, les mouvements de la souris, etc.

Dans notre code JavaScript, nous utilisons `document.querySelector('.slide.current')` pour sélectionner l'élément HTML avec la classe `slide` et la classe `current` dans le document. Cette méthode retourne un seul élément HTML qui correspond au premier élément correspondant au sélecteur CSS spécifié.

Nous utilisons cette méthode pour récupérer la diapositive en cours et modifier sa propriété `display` pour la rendre visible, tout en masquant toutes les autres diapositives. Cela nous permet de créer l'effet de transition entre les diapositives dans le diaporama.

```
const slideshow = document.getElementById('slideshow');

...

// Créer une nouvelle diapositive
const newSlide = document.createElement('div');
newSlide.classList.add('slide');
newSlide.style.backgroundImage = `url(${slide.image})`;

// Ajouter le titre et la description de la diapositive
const title = document.createElement('h2');
title.textContent = slide.title;
newSlide.appendChild(title);

const description = document.createElement('p');
```

```
description.textContent = slide.description;  
newSlide.appendChild(description);
```

...

```
// Ajouter la nouvelle diapositive au diaporama  
slideshow.appendChild(newSlide);
```

et le code HTML :

```
<div id="slideshow">  
  <!-- Les diapositives seront ajoutées via JavaScript -->  
</div>
```

From:  
<https://www.abonnel.fr> - notes informatique & technologie

Permanent link:  
<https://www.abonnel.fr/informatique/langage/page-web/diaporama-css3-html5-javascript>

Last update: **2023/03/19 08:33**

