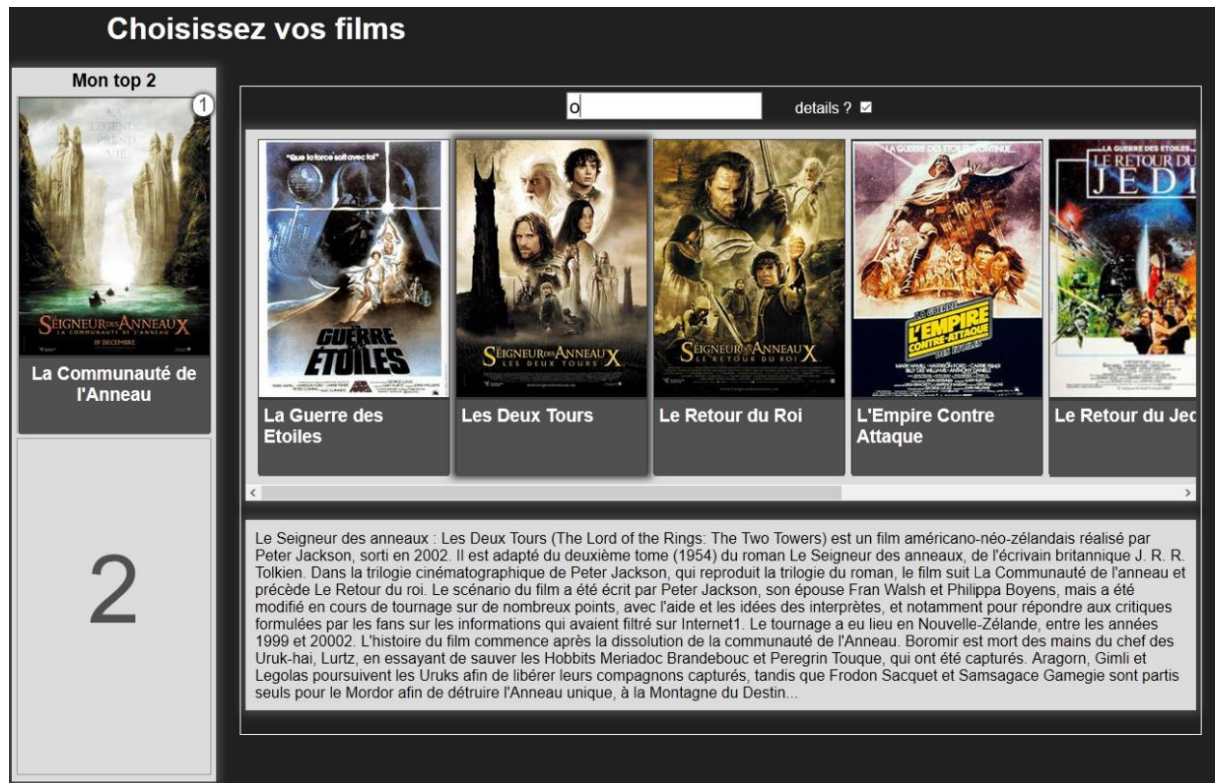


Catalogue de Films

L'objectif est de réaliser une page qui présente un catalogue de films. L'utilisateur peut choisir parmi ces films ses deux films préférés.

Comme l'illustre la figure ci-dessous, la page présente plusieurs zones. Sur la droite on trouve la zone du catalogue où sont affichés les films et sur la gauche la zone où sont affichés les deux films sélectionnés par l'utilisateur.



Le contenu du catalogue des films est construit automatiquement au chargement de la page par rapport à des données définies dans une liste. La structure des données dans la liste est définie un peu plus loin. Tant que l'on respecte le format de ces données, il doit être possible de changer les données dans la liste sans que cela ne nécessite de modifier le code qui permet le chargement du catalogue. Autrement dit, vous avez toute liberté de modifier ce catalogue, en rajoutant vos propres films si vous le souhaitez.

Règles de Gestion

- 1) L'utilisateur peut parcourir les différents films. Lorsqu'il survole à la souris l'un des films des informations sont affichées dans la partie inférieure de la page, si celle-ci est **active**.
- 2) En saisissant un texte dans la zone dédiée au-dessus des films, l'utilisateur a la possibilité de filtrer les films affichés : seuls ceux dont le titre contient le texte saisi sont affichés.

- 3) Une case à cocher permet de choisir si la partie inférieure, celle où apparaissent les informations sur le film survolé, est affichée (**active**) ou pas, selon que la case soit cochée ou non.
- 4) L'utilisateur peut choisir ses deux films préférés. Ceux-ci apparaissent alors dans la zone de gauche. Dans l'image ci-dessous un seul film a été sélectionné. La sélection d'un film se fait simplement en cliquant dessus. Le film vient alors de passer dans l'un des deux emplacements libres sur la gauche et est retiré du catalogue, sur la droite.
- 5) Si les deux emplacements sont déjà pris, un message d'alerte est affiché. Il est possible de retirer l'un des deux films choisis en cliquant dessus. Celui-ci est alors retiré de la sélection et replacé dans le catalogue, à la fin.

Le travail que vous devez réaliser consiste à écrire le code javascript permettant de mettre en œuvre le comportement décrit ci-dessus.

Vous devez utiliser les fichiers contenus dans l'archive fichiers-mini-projet-js2019.zip fournie. Cette archive contient :

- le fichier **index.html** dont vous ne modifierez pas le contenu,
- le fichier **films-static.html** qui est présenté plus loin,
- le fichier **style/style.css**, vous n'avez a priori pas besoin de modifier ce fichier (mais vous êtes libre de l'améliorer),
- le fichier **scripts/script.js**. Vous le complétez avec le code javascript que vous écrirez. Toutes les fonctions que vous écrirez dans ce fichier devront être commentées. C'est-à-dire que chacune commencera par un texte commentaire `/* ... */` qui décrira le rôle de la fonction, ses paramètres et son résultat, quand il y en a un.
- le dossier data qui contient le fichier **film.js**, celui-ci définit la liste (de type *Array*) contenant les données du catalogue des films. Le fichier de données est chargé dans le header du document html, comme le script à compléter.
- le dossier images qui contient les images correspondant aux affiches des films proposés.

Ce qui vous est fourni

- le fichier de données (data/film.js) définit une variable globale `filmData` qui contient la liste des films du catalogue. Vous pouvez compléter la liste, ou en créer une nouvelle, en y ajoutant vos propres films tant que vous respectez le format des données pour les films.
- les films sont représentés sous la forme d'une structure (un *objet javascript*). Chacune de ces données comporte trois champs : **title**, **image** et **text**. Pour tous les trois leur valeur est une chaîne de caractères. Si on dispose d'une variable représentant un tel objet film alors la notation pointée permet d'accéder au champ correspondant. Par exemple :

```
var someFilm = filmData[0];
```

```
// l'objet correspondant au premier film est stocké dans la variable someFilm
```

```
var theTitle = someFilm.title;
```

```
// récupération du titre de ce film (et stockage dans la variable theTitle)
```

```
var someFilm2 = filmData[2];
```

```
// l'objet correspondant au troisième film est stocké dans la variable someFilm2
```

```
var film2Src = someFilm2.image;
```

```
// récupération de la valeur du champ image de ce film et stockage dans la variable film2Src
```

```
var film2Txt = someFilm2.text;
```

```
// récupération de la valeur du champ text de ce film et stockage dans la variable film2Text
```

Le champ **title** représente naturellement le titre du film, le champ **image** identifie la source de l'image correspondant à l'affiche du film et le champ **text** contient le texte d'information affiché dans la zone dédiée lors du survol du film.

Ce que vous devez réaliser

- a) Etudiez le code html du document index.html afin de bien en comprendre la structure et de repérer les différents éléments qui constituent la page, et notamment leurs *id*.

Ce document utilise pour le moment une version vide du fichier scripts/script.js. Et donc les films ne s'affichent pas encore. C'est normal. C'est votre travail dans ce projet de compléter ce fichier de script.

- b) Dans l'archive, vous est fourni le fichier films-static.html. Celui-ci est une copie statique (donc sans code javascript actif) du code HTML correspondant au contenu de la page en action. Ce contenu est donc un exemple de résultat produit par les actions du code javascript lorsqu'il est actif.

Etudiez ce code html afin de comprendre ce que doit produire l'exécution de votre code javascript.

En particulier étudiez le code correspondant à chaque film représenté par un élément *div.film* dans *#films*. Vous analyserez en particulier les *id* qui apparaissent dans les différents éléments. Ils sont de la forme *i-film* et vous ferez le lien entre ce *i* et l'indice de l'article dans la variable *filmData*.

- c) Au chargement de la page, il faut construire le catalogue des films affichés dans *#films*. Pour chacun des films décrit par les données présentes dans *filmData* il faut donc créer l'élément *div.film* correspondant.
- En utilisant *createElement*, définissez une fonction *createFilm* qui prend en paramètre un entier *index* et dont le résultat est l'élément DOM *div.film* correspondant au film d'indice *index*. L'élément créé devra correspondre complètement à ce que l'on peut trouver dans la version statique, il faut par exemple donc intégrer l'image de l'affiche du film et son titre.

Et, en particulier, vous ferez également attention à ce que les `id` et `class` de l'élément créé correspondent à ce qui est obtenu dans la version statique.

- Faites le nécessaire pour qu'au chargement de la page, `#films` soit initialisé correctement avec tous les films de `filmData`.
- d) Choisissez un film, trouvez une image de son affiche et un petit texte le concernant. Ajoutez les données concernant ce film dans le tableau `filmData` en respectant la syntaxe. Vérifiez que le film est bien dans le catalogue au chargement de la page.
- e) Un événement `keyup` est émis à chaque fois que l'on relâche une touche dans un élément `input`.

Faites le nécessaire pour que seuls les films dont le `title` contient le texte présent dans `#filter` soient affichés dans le catalogue `#films`. Le filtrage doit être modifié au fur et à mesure de la saisie des caractères dans `#filter`. En d'autres termes, à chaque fois que vous saisissez une lettre, le filtre doit s'appliquer, et le contenu s'actualiser.

Une manière de procéder peut-être de modifier, en fonction de leur titre, la valeur de la propriété `style.display` des éléments représentant les films.

Etudiez la documentation de la méthode `includes` des objets javascript `String`.

- f) Faites le nécessaire pour gérer l'affichage ou le masquage de la zone `#details` selon que la case `#showDetails` soit cochée ou non.
- g) Faites le nécessaire pour que lors du survol d'un élément `div.film` la valeur du champ `text` de la donnée film correspondante soit affichée comme contenu de l'élément `#details`. Vous devriez pouvoir retrouver la donnée film concernée à partir de l'`id` de l'élément survolé.
- h) Gérez la fin de l'affichage dans `#details` à l'arrêt du survol du film.
- i) Il reste à s'occuper de la sélection des deux films préférés qui est déclenchée par un clic sur l'un des films de `#films`. Pour cela il faut d'abord identifier si l'un des éléments `#select1` ou `#select2` est libre. Si c'est le cas il faut insérer avant l'élément `span` qu'il contient, l'élément `div.film` qui a été cliqué.

Si aucun emplacement n'est libre il faut afficher un message comme demandé.

Enfin, vous devez faire en sorte que le prochain clic sur l'un des films préférés a pour effet

de déplacer l'élément *div.film* correspondant en fin de l'élément *#films*.

L'effet du survol de la souris sur l'élément film reste actif quand le film est sélectionné parmi les favoris.

Remarque : On rappelle que, dans la mesure où un nœud ne peut, évidemment, pas se trouver à deux endroits de l'arbre simultanément, il n'est pas nécessaire de gérer la suppression d'un nœud auprès de son parent quand on le déplace à un autre endroit de l'arbre. La suppression est en effet automatique du fait du déplacement.

Bonus : Vous pourrez rendre ce catalogue *responsive*. Idéalement, mobile first. Vous ferez toutes les adaptations nécessaires pour que ce catalogue soit aussi bien adapté aux petites résolutions qu'aux plus grandes.