Catalogue de Films

Projet de Programmation Web[1]

–M1–

Projet Javascript

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

---

1 Dates et informations importantes

- Rendu du projet : Dimanche 24 Mai 2020 à minuit (via mail/ MyCourse).
- Le travail peut être réalisé individuellement (jusqu'à 9), en binôme (jusqu'à 11) ou en trinôme (jusqu'à la fin).
- Le respect des consignes sera pris en compte pour l'évaluation de ce travail, en plus de la qualité du travail rendu.

2 Versions

Ce document est susceptible d'être modifié, veillez à travailler avec la dernière version.

- Version actuelle : version du 12 avril 2020.
- Dernière version disponible sur : https://www.lamsade.dauphine.fr/ halili/Programmation_Web/Projets/Projet2.pdf

3 Description du projet

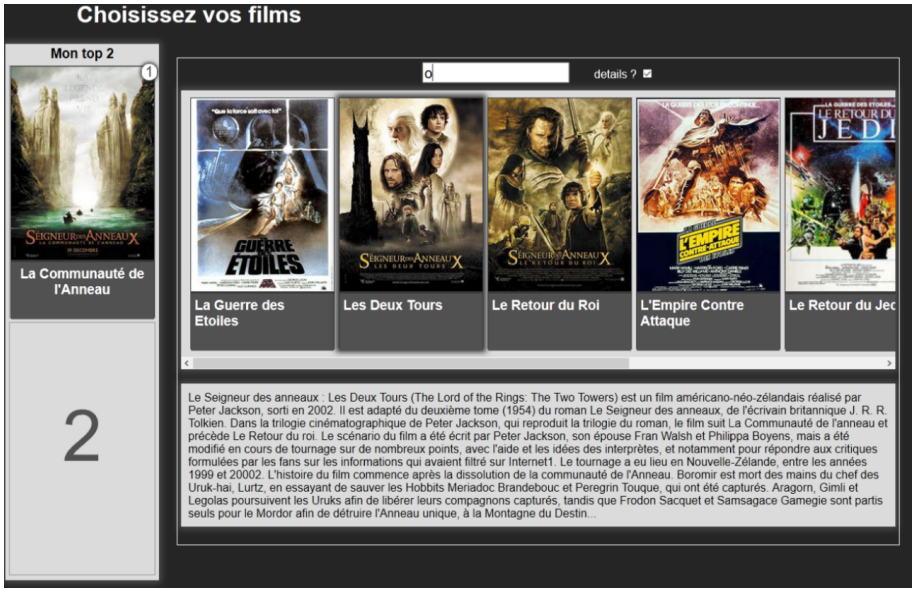
Comme l'illustre la figure ci-dessous, la page présente plusieurs zones. Sur la droite on trouve la zone du catalogue où sont affichés les films et sur la gauche la zone où sont affichés les deux films sélectionnés par l'utilisateur.

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.

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. 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. 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.

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 placer dans l'un des deux emplacements libres sur la gauche et est retiré du catalogue, sur la droite. 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.



4 Travail demandé

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-projet2-js2020.zip** fournie. Cette archive contient :

- le fichier **mini-projet-js-2020.html** que vous renommerez en index.html mais dont vous ne modifierez pas le contenu,
- le fichier **mini-projet-js-2020-static.html** qui est présenté plus loin,
- le fichier **style/style-projet2020.css**, vous n'avez a priori pas besoin de modifier ce fichier,
- le fichier **scripts/project2020.js**, que vous devez le compléter 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 est imposé et 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 comportent trois champs : title, image et text. 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.

  1. Etudiez le code html du document **mini-projet-js-2020.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/project2020.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.

2. Dans l'archive, vous est fourni le fichier **mini-projet-js-2020-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.

3. 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 et 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**.

4. 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.

5. Un événement keyup est émis à chaque fois que l'on relâche une touche dans un élément input.

6. 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.

   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. Faites le nécessaire pour gérer l'affichage ou le masquage de la zone #details selon que la case #showDetails soit cochée ou non.

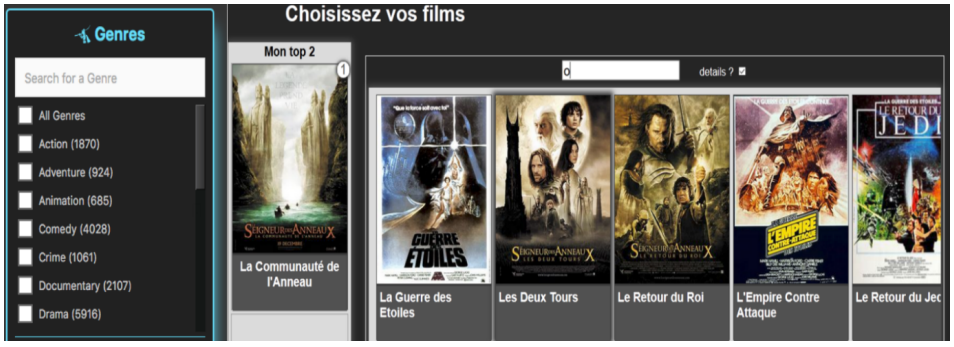
7. Faites le nécessaire pour que lors du survol d'un élément div.film la valeur du champ text de la donnée film correspondante soit affichée comme contenu de l'élément #details. Vous devriez pouvoir retrouver la donnée film concernée à partir de l'id de l'élément survolé.
8. Gérez la fin de l'affichage dans #details à l'arrêt du survol du film.
9. 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. 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.
10. Appliquez les modifications nécessaires pour permettre un filtrage de films par genre à partir d'une liste des cases à cocher réprésentant les genres tout à gauche du catalogue (ajout d'un attribut genre, etc..)

11. Ajoutez un champs de recherche au dessus de la liste permettant de filtrer la liste des genres affichés.



12. Ajoutez une page d'accueil (accueil.html) avec un menu horizontal contenant les liens aux pages (1)**Accueil** (accueil.html) (2)**Catalogue** (index.html), (3)**Nous contacter** (contact.html) et (4)**Mon Compte** (compte.html). D'autres pages de votre choix peuvent être introduites.
    - Privilégiez une courte présentation dans la page d'accueil.
    - La page **Mon Compte** doit présenter un formulaire de connexion avec les champs (adresse e-mail et mot de passe). Ce forumulaire doit être suivi par un lien vers un deuxième formulaire d'inscription (dans une nouvelle page (inscription.html)).
    - Tous les formulaires doivent munir des vérifications nécessaires pour avertir les utilisateurs d'un oubli ou d'une erreur lors du remplissage des formulaires.
    - Mettez à jour le rendu visuel de la page **index.html** (garder le même design que les autres pages; le header, le footer, etc.).
13. Rendez votre travail sous la forme d'une archive. Le nom de cette archive sera **Nom-Groupe-projetJS.zip**. Cette archive contiendra un répertoire dont le nom sera Nom-Groupe-Projet. Le contenu de ce répertoire sera organisé ainsi comme suivant:
    - un fichier **guide.txt**. Ce fichier mentionnera pour chacune des questions précédentes si elle a été traitée ou non et les éventuels problèmes de votre solution par rapport au cahier des charges demandé. Si vous avez ajouté des fonctionnalités, vous devez les détailler également dans ce fichier.
    - les fichiers mentionnés en introduction.

Tous vos fichiers doivent être codés en **UTF-8** et les noms des fichiers (y compris leurs extensions) doivent être en minuscules.