

Webinaire JavaScript

Utilisez JavaScript pour réaliser un site dynamique

Description du webinaire (compétences abordées) : Dans cet exercice assez complet, vous allez mettre en œuvre vos compétences en JavaScript afin de réaliser un site dynamique. Vous appellerez une API pour récupérer les données, manipuler le DOM et valider des formulaires.

Exercice 1

 **Compétences visées :**

- **Utiliser des variables en JavaScript**
- **Utiliser des structures conditionnelles et itératives en JavaScript**
- **Utiliser des fonctions en JavaScript**
- **Faire des requêtes ajax avec fetch et les promises**
- **Utiliser les api navigateur (validation et storage)**
- **Manipuler le DOM avec JavaScript**

Projets et paths concernés :

- **DW - P5**
- DW - P6
- DW - P7
- DA Front-End - projets 4 à 7 et 10 à 14

Énoncé

Contexte

Vous venez d'obtenir un poste dans une entreprise éditrice d'un journal papier distribué gratuitement. Celle-ci souhaite créer un canal digital de communication pour attirer un plus grand public et a déjà commencé le développement des outils numériques. Mais l'entreprise a besoin de votre aide pour terminer le site.

Actuellement une application back-end a déjà été entièrement créée, elle propose plusieurs fonctionnalités telles que la récupération des derniers articles, l'ajout d'articles aux favoris, et la récupération des articles favoris.

Une maquette HTML/CSS du site est aussi déjà au point, avec toutes les pages disponibles aux utilisateurs. [Le code du projet](#) est disponible ici.

Votre objectif est de **faire communiquer cette maquette avec l'application back-end** afin d'en faire un projet front-end fonctionnel.

Besoin fonctionnel

- A l'arrivée sur le site, les utilisateurs doivent voir les derniers articles publiés par le site, avec le nom de l'article, sa date de publication, et son contenu (image + texte). Les articles doivent apparaître du plus récent au plus ancien.
- Sur chaque article est présente une étoile blanche. Si l'utilisateur clique dessus, elle devient jaune et l'article est ajouté dans le localStorage pour être conservé par l'utilisateur. Il doit être possible de re-cliquer sur l'étoile pour retirer l'article des favoris.
- Dans la barre de navigation du site, deux boutons sont présents :
 - Le premier, en forme d'étoile, permet de se diriger vers la page des articles enregistrés en favoris par l'utilisateur. Cette page doit afficher de la même façon que l'accueil les articles, mais cette fois-ci, ceux enregistrés en favoris par l'utilisateur.
 - Le second en forme de courrier, redirige vers une page de contact, où l'utilisateur peut envoyer un message.

- Pour ce faire, il doit renseigner un formulaire avec son adresse mail et son message puis cliquer sur le bouton envoyer.
- Les champs doivent être vérifiés au moment du click sur le bouton, pour valider que l'adresse mail est au bon format, et que le message comporte au moins 5 caractères et au maximum 400.
- Cette fonctionnalité n'enverra aucune requête vers le back-end mais affichera un message de confirmation de l'envoi.

Conseils

- Quasi tous les fichiers HTML et CSS sont complets, vous n'avez pas besoin de les modifier pour réaliser cet exercice. Il y a deux exceptions pour les fichiers HTML :
 - Un template pour l'affichage des articles est fourni dans le dossier /view/__template . Vous pouvez l'utiliser directement, le modifier, le copier dans du JavaScript ou le recréer à partir du JavaScript, comme bon vous semble.
 - Les champs du formulaire de contact peuvent être modifiés pour ajouter des contraintes (pattern, required, minlength...).
- Des fichiers JavaScript ont été prévus pour réaliser le code avec quelques détails fournis: vous pouvez les utiliser ou les supprimer pour en créer d'autres.
- Dans le back-end, un fichier JSnews.json est présent, il peut être chargé sur Talend Api Tester. Vous pouvez ensuite tester le back-end, voir les différentes requêtes possibles, et le format de retour.

Etapes de réalisation

Vous ne savez pas comment démarrer l'exercice ? Voici une suggestion d'étapes à suivre pour mener à bien le projet.

Les étapes 1 et 2 sont recommandées en classes mais vous pouvez aussi réaliser des fonctions si vous n'êtes pas à l'aise avec la POO.

1. Avant de démarrer votre travail, commencez par bien lire l'énoncé, télécharger le code du projet et lire le README.
2. Créez une classe pour représenter un article dans le fichier article.js. Ajoutez-lui une propriété pour chaque donnée d'un article, et une méthode pour formater la date.
3. Créez une classe dans ArticleManager avec une propriété contenant la liste des articles, et une méthode pour trier ces articles par date décroissante.
4. Créez des fonctions dans favoritesManager pour gérer les favoris : ajouter, supprimer, récupérer les favoris et les enregistrer. Une fonction supplémentaire peut être utile pour retourner un tableau avec les id des favoris.
5. Dans la page home, requêtez le back-end pour récupérer les articles et les afficher sur la page HTML.
6. Ajoutez un événement click sur le bouton pour ajouter au favoris en appelant la fonction dans favoritesManager.
7. Dans la page favorites, requêtez le back-end pour récupérer les articles en favoris. Il faut envoyer les id favoris enregistrés dans le localStorage. Puis faites les afficher sur la page HTML.
8. Dans la page contact ajoutez un événement sur le click du bouton et traitez les champs du formulaire avec l'ApiValidation.

Pour aller plus loin

Si vous voulez aller plus loin avec cet exercice :

- Enregistrez la date d'ajout aux favoris, et trie les articles à afficher, du plus récent au plus anciennement ajoutés aux favoris.
- Au rechargement de la page, affichez en jaune les étoiles des articles déjà présents dans les favoris.
- Actuellement vous travaillez en local, mais le projet devra être

déployé sur différents environnements. Prévoyez un moyen simple de gérer ces environnements sans devoir changer de multiples lignes de codes dans différents fichiers.

Ressources utiles

En plus des cours OpenClassrooms, vous pouvez vous aider des ressources suivantes :

- [Les bases de JavaScript](#)
- [Utiliser l'apiValidation en JS](#)
- [Utiliser des query selector à la place des sélecteurs javascript basiques](#)
- [Utiliser l'interpolation de variables avec les backtick](#)
- [Utiliser les Promise](#)
- [Utiliser Ajax pour lancer une requête http](#)
- [Trier un tableau](#)

Correction

Un exemple de solution est disponible sur [ce repository](#).