

Travail pratique 1 - JavaScript - Syntaxe de base et affichage

Objectifs

Le but de ce travail pratique est de vous familiariser avec la syntaxe de base du JavaScript et de vous faire pratiquer l'affichage de contenu dans une page web.

Les trois travaux de la session sont liés. Vous allez travailler sur le même projet pour les trois travaux. Vous allez créer un site web d'un produit ou un service de votre choix. Vous allez ajouter de l'interactivité à votre site web à chaque travail.

Dans ce travail, vous allez afficher une liste de produits que vous pourrez trier et vous allez ajouter de l'interactivité pour afficher le détail d'un produit.

Vous devez respecter la maquette en fil de fer fournie mais vous pouvez remplacer par le contenu de votre choix (texte, images, etc.). Bien que le style ne soit pas évalué, votre CSS doit être valide et respecter les bonnes pratiques en terme de lisibilité et de hiérarchie.

Vous devez montrer l'avancement du projet à au moins une reprise. Je dois voir le code pour m'assurer que vous êtes sur la bonne voie.

Tâches à réaliser

Générer la navigation dynamiquement

- Vous devez créer un tableau contenant les liens de la navigation. Chaque lien doit contenir le texte à afficher et le nom de la page correspondante. Vous devez utiliser un objet pour chaque lien.
- Vous devez générer les onglets de la navigation dynamiquement à partir du tableau. Les liens doivent être fonctionnels et mener à la page correspondante.
- Au chargement de la page, le lien de la page courante doit être mis en évidence (ex: souligné, gras, couleur différente, etc.). L'objet Document contient l'information nécessaire pour faire cela. Faites une recherche et indiquez en commentaire la ressource que vous avez utilisée.

Mettez le code pour la navigation dans un fichier séparé que vous pourrez ajouter à chaque page. Ce n'est le plus optimal pour un site concret mais nous allons optimiser cela à la fin de la session.

La balise nav doit être donc être vide dans le HTML. Vous devez générer le contenu de la balise nav à partir du JavaScript.

Afficher la liste de produits

- Vous devez afficher une liste d'éléments HTML à partir d'un tableau d'objets contenant les données des produits. (Maximum 5 produits)
- Vous devez utiliser un objet pour chaque produit. Chaque objet doit contenir au moins 3 propriétés (ex: nom, prix, description, etc.). Ajoutez un id unique à chaque élément de la liste. Vous allez en avoir besoin pour la suite.
- Vous devez créer vous-même un tableau pour regrouper les objets. Vous pouvez utiliser des données fictives.
- Utilisez des images à partir de banques libres de droit comme [Unsplash](#) ou [Pexels](#).
- Renommez-vos images : `nom-du-produit.jpg` ou `nom-du-produit.webp` pour faciliter la programmation.
- Vous devez générer le lien vers les images en concaténant le chemin vers le dossier `assets/img/` avec le nom du produit et l'extension nécessaire.
- Lorsque vous générez le contenu de la liste, passez le id à l'élément HTML qui aura un événement de clic. Vous allez en avoir besoin pour la suite.

Afficher le détail d'un produit

- Au clic d'une vignette de la liste, vous devez afficher le détail dans la section aside.
- Utilisez une fonction pour réaliser cette tâche.
- L'objet cliqué doit être mis en évidence (ex: souligné, gras, couleur différente, etc.).

Trier la liste de produits

- Vous devez permettre de trier et filtrer les éléments de la liste en cliquant sur des boutons. Vous devez pouvoir filtrer d'au moins trois façons (ex: alphabétique a-z, alphabétique z-a, par prix croissant, etc.)
- Utilisez une seule fonction pour réaliser cette tâche.
- N'utilisez **pas** de champs de formulaire pour cet exercice comme les boîtes à cocher ou les menus déroulants

Remise

Vous devez remettre votre travail dans un dossier .zip, nommé avec votre nom et prénom, sur Teams dans la section **Devoirs**. Assurez-vous que le dossier contient tous les fichiers nécessaires pour afficher la page dans un navigateur. Vous devez inclure les fichiers HTML, CSS, JavaScript et les images.

Critères d'évaluation - Compte pour 20% de la note finale

Utilisation efficace des fonctions

- Les fonctions sont utilisées pour éviter la répétition de code
- Les fonctions sont efficacement utilisées

- Le code contient une fonction `init()` qui est appelée au chargement de la page et qui appelle les autres fonctions
- Le code contient une fonction pour chaque tâche
- Les paramètres et les valeurs de retour des fonctions sont efficacement utilisés

Utilisation efficace des chaînes de caractères

- Les chaînes de caractères sont utilisées pour afficher du contenu dans la page
- Au moins 5 des méthodes suivantes de chaînes de caractères sont utilisées: `toUpperCase`, `toLowerCase`, `charAt`, `indexOf`, `lastIndexOf`, `substring`, `slice`, `split`, `replace`, `includes`, `startsWith`, `endsWith`, `trim`, `concat`, `padStart`, `padEnd`
- Les méthodes de chaînes de caractères sont utilisées correctement et efficacement

Utilisation efficace des tableaux

- Les tableaux sont utilisés pour regrouper plusieurs éléments de même type
- Au moins 5 des méthodes suivantes de tableaux sont utilisées: `push`, `pop`, `splice`, `sort`, `reverse`, `indexOf`, `forEach`, `filter`, `join`, `split`
- Les méthodes de tableaux sont utilisées correctement
- Il est possible de filtrer et trier les éléments de la liste d'au moins 3 façons différentes

Utilisation efficace des objets

- Les objets sont utilisés pour regrouper des unités de données
- Les objets sont utilisés efficacement.
- Au moins 1 des méthodes d'objets suivantes est utilisée: `Object.keys`, `Object.values`, `Object.entries`

Structure et qualité de code

- Le code ne contient pas d'erreur
- Les `console.log()` de débogage sont retirés ou commentés
- Les variables sont nommées de façon explicite
- Les noms de fonctions sont explicites
- Le code est bien indenté
- Le code est facile à lire
- Le code est structuré de façon logique de façon à éviter la répétition
- Le code est structuré pour éviter les erreurs d'ordre d'exécution

Autonomie et attitude professionnelle

- Les éléments du devis sont présents
- Le travail est remis à temps
- Les références à du code externe sont bien documentées
- Le code n'est pas copié-collé de sources externes ou de générateurs de code ou par intelligence artificielle
- Le dossier de remise est bien nommé et complet.
- L'étudiant.e a montré l'avancement de son travail à son enseignant.e

