Module : CMS
Semestre : 5
Année Universitaire : 2024/2025

UNIVERSITE IBN ZOHR
FACULTE DES SCIENCES
AGADIR

Solution du TP N°2 : HTML & CSS & JS

Objectif : Cette activité vise à :

- Pratiquer l'intégration de CSS dans un fichier HTML.
- Renforcer la compréhension des concepts du DOM (Document Object Model) en JavaScript.
- Mettre en œuvre des fonctions JavaScript pour manipuler des éléments du DOM.
- Maîtriser l'utilisation de fetch pour récupérer des données depuis une API REST.

Contexte

Vous allez créer un site e-commerce pour une boutique de café en ligne. Les données des produits seront récupérées à partir de l'API REST suivante :

https://fake-coffee-api.vercel.app/

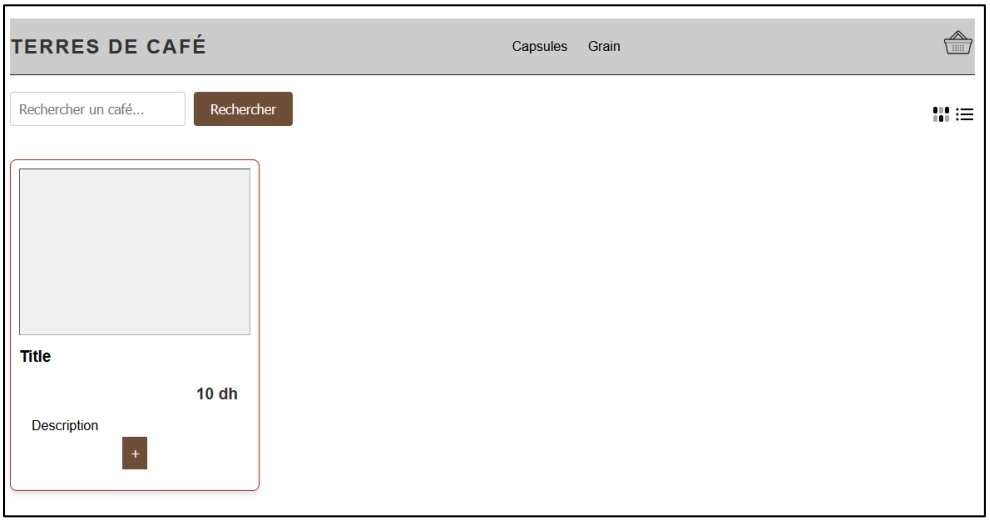
Ressources

Les ressources du module, y compris les connaissances sur le DOM et JavaScript. Téléchargez les fichiers de ressources fournis, comprenant un fichier HTML ,un fichier CSS et un fichier js. Assurez-vous de les conserver dans un seul dossier

---

Tâches à réaliser :

1) Dans le fichier CSS fourni, complétez les règles CSS pour reproduire le design présenté dans l'image suivante :



2) **Function getProducts()** : Complétez le code de la fonction JavaScript `getProducts()` dans le fichier “script.js”. Cette fonction doit permettre de récupérer les données des produits depuis l'API et les stocker dans une variable globale pour une utilisation ultérieure dans l'application. Utiliser la méthode `fetch()` pour envoyer une requête `GET` à l'URL de l'API : `'https://fake-coffee-api.vercel.app/api'`

3) Implémentez les fonctions suivantes dans le fichier script.js pour gérer l'affichage des produits :

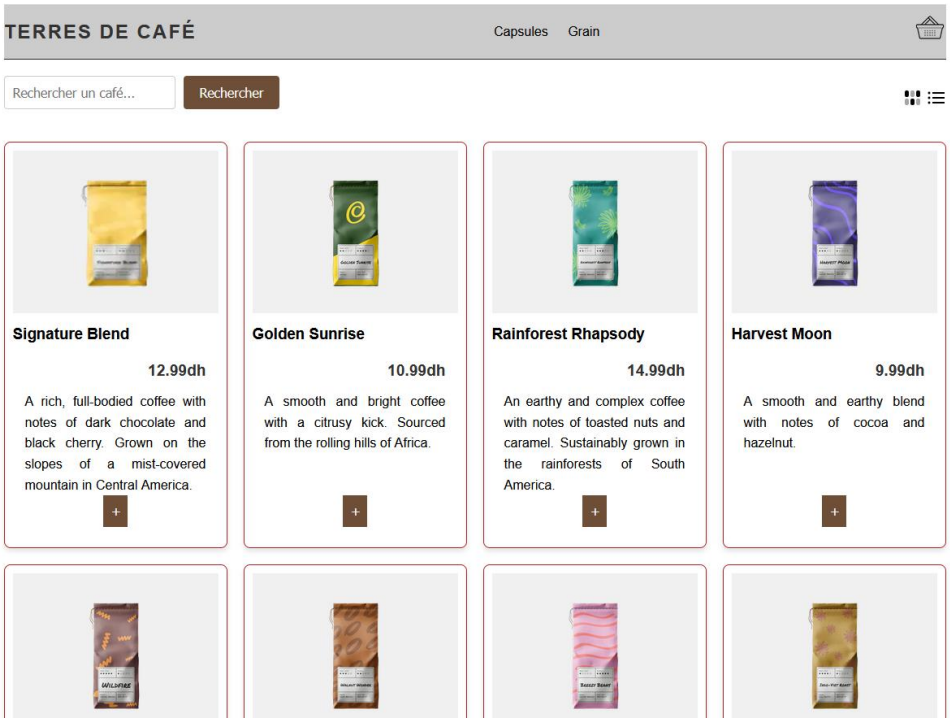
Function createProductCard(product) : Crée et retourne un élément div représentant une carte de produit.

- Paramètre : un objet 'product' contenant les détails du produit.
- Doit inclure : image, nom, prix, description et bouton "+".
- Utilisez ${product.image_url} , ${product.name} , ${product.price}, ${product.description}

Function displayProducts(products) : Affiche tous les produits sur la page.

- Paramètre : un tableau 'products' contenant les objets de produits retourner par fetch.
- Sélectionnez le conteneur des produits dans le DOM
- Vide le contenu existant du conteneur.
- Utilisez createProductCard() pour chaque produit du tableau
- Ajoutez chaque carte créée au conteneur.

Après l'exécution de votre code, le résultat affiché dans le navigateur doit être visuellement identique à l'image suivante :



4) Avant d'implémenter la fonctionnalité de basculement des modes d'affichage (grille / liste) :

- Donner le code js pour sélectionner les éléments DOM nécessaires :
  - Le conteneur des produits
  - L'icône pour la vue en grille
  - L'icône pour la vue en liste
- Ajoutez des écouteurs d'événements aux icônes pour les fonctions setGridView et setListView (que vous implémenterez par la suite)
- Implémentez les fonctions suivantes :

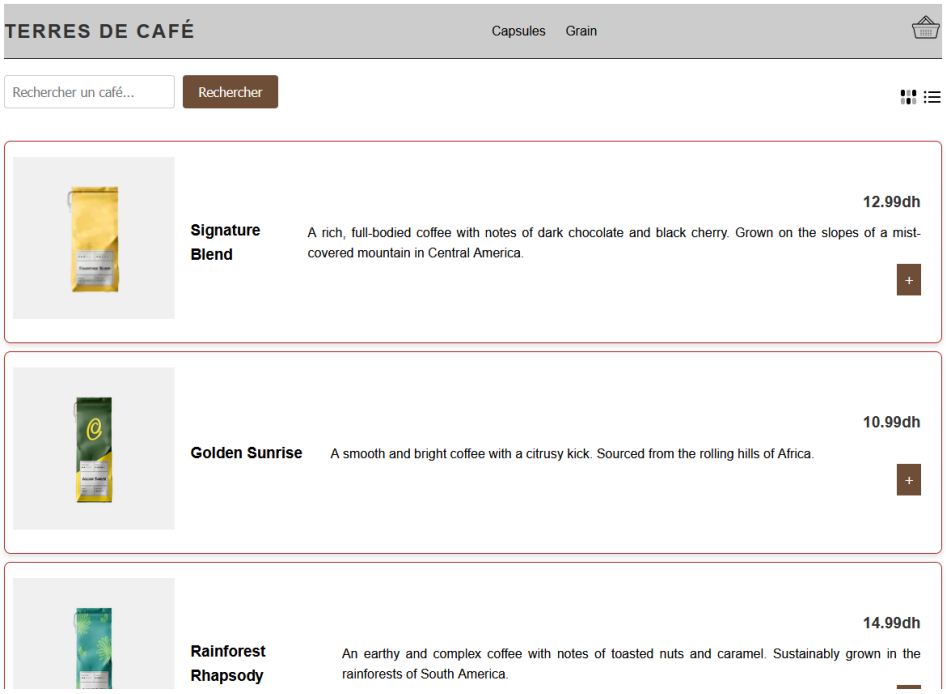
Function setGridView() :

- Modifiez le style du conteneur pour afficher les produits en grille.
- Ajustez les styles des cartes de produits pour convenir à une vue en grille.
- Utilise Flexbox ou CSS Grid pour disposer les produits côte à côte

Function setListView() :

- Modifiez le style du conteneur pour afficher les produits en liste.
- Ajustez les styles des cartes de produits pour convenir à une vue en liste.

Après l'implémentation de la fonction setListView(), lorsque l'utilisateur clique sur l'icône de liste, l'affichage des produits doit changer pour correspondre à l'image suivante :



5) Implémentez une fonctionnalité de filtrage qui permet aux utilisateurs de rechercher des produits spécifiques parmi les données retournées par l'API dans la fonction getProducts() .

Étapes à suivre :

- Assurez-vous que les données récupérées par fetch dans getProducts() sont stockées dans une variable globale accessible, par exemple 'products'.
- Créez une fonction filterProducts() qui :
  - Récupère la valeur saisie dans la barre de recherche.
  - Filtre le tableau 'products' (contenant les données de l'API) en fonction de cette valeur.
  - Affiche les produits filtrés en utilisant la fonction displayProducts().
- Ajoutez un écouteur d'événements sur le champ de recherche, en utilisant l'événement 'input' pour déclencher le filtrage en temps réel pendant la saisie.

Après avoir implémenté la fonctionnalité de filtrage, voici le comportement et l'affichage que vous devriez obtenir :