

# PROJET Algo JS – MINI SITE E-COMMERCE

Vous ne devez utiliser aucune lib / aucun framework. Vous devez utiliser les structures vues en cours. Les tris abordés pendant les cours.

¶

## Objectif

Vous devez développer la logique JavaScript d'un mini site e-commerce. Ce projet vise à évaluer votre capacité à : • structurer des données • implémenter des algorithmes classiques • écrire un code JavaScript clair, lisible et fonctionnel

¶

## Structure du projet (obligatoire)

```
/project
├── index.html
├── style.css
└── script.js
```

¶

## Modèle de données – Produit

```
class Product {
  constructor(id, name, price, category) {
    this.id = id;
    this.name = name;
    this.price = price;
    this.category = category;
  }
}
```

¶

### Partie 1 – Catégories et sous-catégories

Les produits du site sont organisés par catégories hiérarchiques (exemple : Électronique → Ordinateurs → Portables).

Travail demandé • Créer une structure permettant de stocker les catégories • Ajouter une catégorie principale • Ajouter une sous-catégorie à une catégorie existante • Afficher les catégories : • de manière hiérarchique • par ordre alphabétique

¶ Le code doit permettre de parcourir logiquement les catégories.

¶

### Partie 2 – Gestion des produits

Les produits sont stockés dans un tableau JavaScript.

Travail demandé • Ajouter un produit • Afficher la liste des produits • Trier les produits : • par prix croissant • par nom (A → Z)

¶

### Partie 3 – Recherche d'un produit

Chaque produit possède un identifiant unique (id).

Travail demandé • Implémenter une fonction permettant de retrouver rapidement un produit à partir de son id • La fonction doit retourner : • le produit s'il existe • null sinon

¶

### Partie 4 – Panier d'achat

Le panier fonctionne selon le principe :

Dernier produit ajouté → premier retiré

Travail demandé • Ajouter un produit au panier • Retirer le dernier produit ajouté • Afficher le contenu du panier

¶

## Partie 5 – Gestion des commandes

Les commandes sont traitées dans l'ordre d'arrivée.

Travail demandé • Ajouter une commande à traiter • Traiter (retirer) la prochaine commande • Afficher les commandes en attente

☒

## Partie 6 – Historique des actions (BONUS)

Mettre en place un système permettant : • d'annuler la dernière action effectuée • de refaire une action annulée

☒

## Données de test obligatoires

Votre code doit fonctionner avec les appels suivants :

```
addCategory("Électronique");
addCategory("Vêtements");

addSubCategory("Électronique", "Ordinateurs");
addSubCategory("Vêtements", "Hommes");

addProduct(new Product(10, "PC Portable", 899, "Ordinateurs"));
addProduct(new Product(5, "T-shirt", 19, "Hommes"));
addProduct(new Product(20, "Smartphone", 699, "Électronique"));

searchProductById(10);

sortByPrice();
sortByName();

addToCart(10);
addToCart(5);
removeFromCart();

addOrder("Commande #1");
addOrder("Commande #2");
processOrder();
```

☒

## Règles d'évaluation

- Le projet doit fonctionner sans erreur JavaScript
- Le JavaScript est prioritaire sur le design
- Le code doit être :
  - lisible
  - correctement indenté
  - structuré en fonctions claires
  - Toute tentative de plagiat sera sanctionnée

☒

## Barème indicatif (20 points)

Partie Points Catégories & sous-catégories 3 Produits & tri 3 Recherche 3 Panier 3 Commandes 3 Bonus (ou autre feature) 5

☒

## Conseils

- Commencez par faire fonctionner la logique JavaScript avant l'interface
- Testez chaque fonctionnalité étape par étape
- Privilégiez un code simple et compréhensible
- Utilisez les structures de données vues en cours

¶

Bon courage. Ce projet évalue votre capacité à raisonner, structurer et coder proprement en JavaScript.

## Rendu

---

Code à déposer sur MyGES en zip. Veuillez nommer votre dossier de rendu avec votre **NOM Prénom** avant de le zipper svp. Pensez à ajouter un readme si nécessaire !

## ATTENTION

---

Comme je l'ai répété plusieurs fois pendant ce cours, chacun va avoir une vision différente de son algorithme. C'est pourquoi si je détecte des codes très similaires, la note sera divisée par 2 (ou pas 3 ou par 4, ...). Si le code est jugé fait par l'IA, et qu'il n'y a pas d'explication globale de la reflexion, ce sera jusqu'à -10 pts.