



420-SN1-RE

Programmation en sciences

Exercices

Programme / Département	200.B1 Sciences de la nature
Nom	Exercices pour le cours 11
Session	Hiver 2024
Professeur	Tommy Gagnon Joyal
Bureau / téléphone	À venir
Courriel	Utiliser MIO sur Omnivox / Sur Discord
Disponibilité	Lundi de 10h40 à 15h20 et Mardi de 10h40 à 12h30 (sur rendez-vous seulement)

Exercices	1
Exercice 1: Calcul du prix total d'une liste de courses	3
Exemple d'utilisation:	3
Exercice 2: Recherche de produits par prix	3
Exemple d'utilisation:	3
Exercice 3: Vérification de la présence d'un produit	3
Exemple d'utilisation:	3
Exercice 4: Panier d'achats interactif	4
Exercice 5: Gestion d'un magasin virtuel	4
Données (JSON)	4

Exercice 1: Comparaison de produits

Écrivez une fonction `comparer_produits(liste_produits, nom_produit_1, nom_produit_2)` qui prend en entrée une liste de produits au format JSON et les noms de deux produits, et compare leurs prix. La fonction doit renvoyer le nom du produit qui est le moins cher.

Exercice 2: Calcul du montant de la taxe

Écrivez une fonction `calculer_montant_taxe(liste_produits)` qui prend en entrée une liste de produits au format JSON et calcule le montant total des taxes à payer pour tous les produits.

Exercice 3: Recherche de produits par nom

Écrivez une fonction `rechercher_produit_par_nom(liste_produits, nom_produit)` qui prend en entrée une liste de produits au format JSON et le nom d'un produit, et renvoie toutes les informations sur ce produit s'il est présent dans la liste.

Exercice 4: Génération de facture

Écrivez une fonction `generer_facture(liste_produits, panier)` qui prend en entrée une liste de produits au format JSON et le panier actuel de l'utilisateur, et génère une facture détaillée indiquant chaque produit avec son prix unitaire, sa quantité et son prix total, ainsi que le montant total à payer.

Exercice 5: Gestion des stocks

Écrivez une fonction `mise_a_jour_stock(liste_produits, panier)` qui prend en entrée une liste de produits au format JSON et le panier actuel de l'utilisateur, et met à jour les stocks de chaque produit en soustrayant la quantité achetée du stock disponible.

Données (JSON)

```
liste_de_produits = [  
    {"nom": "Lait", "prix": 2.5, "quantite_stock": 100},  
    {"nom": "Pain", "prix": 1.0, "quantite_stock": 150},  
    {"nom": "Oeufs", "prix": 3.0, "quantite_stock": 80},  
    {"nom": "Fromage", "prix": 4.0, "quantite_stock": 120},  
    {"nom": "Jus d'orange", "prix": 2.0, "quantite_stock": 90},
```

```
{"nom": "Pâtes", "prix": 1.5, "quantite_stock": 200},  
{"nom": "Riz", "prix": 2.0, "quantite_stock": 180},  
{"nom": "Tomates", "prix": 1.8, "quantite_stock": 120},  
{"nom": "Pommes de terre", "prix": 1.2, "quantite_stock": 150},  
{"nom": "Poulet", "prix": 5.0, "quantite_stock": 100},  
{"nom": "Boeuf", "prix": 8.0, "quantite_stock": 80},  
{"nom": "Saumon", "prix": 7.0, "quantite_stock": 90},  
{"nom": "Bananes", "prix": 2.0, "quantite_stock": 100},  
{"nom": "Carottes", "prix": 1.2, "quantite_stock": 120},  
{"nom": "Pommes", "prix": 1.5, "quantite_stock": 130},  
{"nom": "Poires", "prix": 1.8, "quantite_stock": 110},  
{"nom": "Yaourts", "prix": 1.0, "quantite_stock": 200},  
{"nom": "Céréales", "prix": 3.0, "quantite_stock": 150},  
{"nom": "Biscuits", "prix": 2.5, "quantite_stock": 180},  
{"nom": "Chocolat", "prix": 3.5, "quantite_stock": 160},  
{"nom": "Café", "prix": 4.0, "quantite_stock": 140},  
{"nom": "Thé", "prix": 3.0, "quantite_stock": 170},  
{"nom": "Légumes surgelés", "prix": 2.8, "quantite_stock": 110},  
{"nom": "Pizzas", "prix": 6.0, "quantite_stock": 90},  
{"nom": "Glaces", "prix": 3.0, "quantite_stock": 120},  
{"nom": "Chips", "prix": 2.0, "quantite_stock": 200},  
{"nom": "Soda", "prix": 1.5, "quantite_stock": 180},  
{"nom": "Eau minérale", "prix": 1.0, "quantite_stock": 220}
```

]