

Gestionnaire de stock

Création de dictionnaires

```
9  # Dictionnaire contenant les produits, leur quantité et leur prix
10 inventaire = {
11     "article1" : {"produit" : "tomate", "quantité" : 4, "prix" : 0.5},
12     "article2" : {"produit" : "coco", "quantité" : 6, "prix" : 0.8},
13     "article3" : {"produit" : "pomme", "quantité" : 3, "prix" : 0.7},
14     "article4" : {"produit" : "kiwi", "quantité" : 8, "prix" : 1.5}
15 }
16
17 # Dictionnaire pour définir les seuils personnalisés par produit (quantité au plus bas autorisée)
18 seuils = {
19     "tomate": 5,
20     "coco": 4,
21     "pomme": 3,
22     "kiwi": 6
23 }
24
```

Fonctions

```
25 # Fonction pour afficher l'inventaire du moment (produits et leur quantité)
26 def demande_inventaire() :
27     maintenant = datetime.now() # Variable pour faire apparaître le temps à un moment T
28     print("Stock au ", maintenant.strftime("%d/%m/%Y %H:%M:%S"), ":") # La méthode .strftime permet de définir les formats
29
30     for i in inventaire: # Boucle pour passer tous les articles de l'inventaire
31         print(inventaire[i]["produit"], ":", inventaire[i]["quantité"]) # Pour chaque article on montre le produit et la quantité
32
33 # Fonction pour vérifier si le stock est inférieur au seuil de chaque produit
34 def verification_seuil():
35     for article in inventaire.values(): # On met .values() parce qu'on veut parcourir les valeurs, et non pas les clés du dictionnaire (article1,
36         seuil = seuils.get(article["produit"], 5) # On utilise la méthode .get() pour aller chercher le seuil associé au produit dans le dictionnai
37         # Si le seuil n'est pas défini, il sera à 5 par défaut
38         if article["quantité"] < seuil:
39             print(f"Stock de {article['produit']} bas, achat nécessaire. (Seuil : {seuil})")
40
41 # Fonction pour obtenir la valeur totale du stock
42 def valeur_totale_stock():
43     total = 0 # Déclaration de la variable "total"
44     print("\nValeur du stock par produit :")
45
46     for article in inventaire.values(): # On met .values() parce qu'on veut parcourir les valeurs, et non pas les clés du dictionnaire (article1,
47         valeur_article = article["quantité"] * article["prix"] # Calcule la valeur totale par produit
48         print(f"- {article['produit']} : {article['quantité']} x {article['prix']} € = {valeur_article:.2f} €") # .2f est une instruction de format
49         total += valeur_article # Pour calculer le total, on stocke la valeur totale d'un produit, puis on y ajoute les autres
50
51     print(f"\nValeur totale du stock : {total:.2f} €\n") # .2f est une instruction de formatage pour mettre 2 chiffres après la virgule
52
```

Exploitation du système

```
94 # Exploitation du système de gestion du stock
95 while True:
96     tache = input("\nQue voulez-vous faire ? (inventaire, entrée, sortie, valeur) \n\nVotre choix : ").lower() # Input pour saisir l'action à faire
97
98     # Liste de conditions pour vérifier que les mots clefs sont saisis dans l'input + appel des fonctions correspondantes
99
100     if "inventaire" in tache:
101         demande_inventaire()
102         verification_seuil()
103
104     elif "entrée" in tache:
105         entree_stock()
106
107     elif "sortie" in tache:
108         sortie_stock()
109
110     elif "valeur" in tache:
111         valeur_totale_stock()
112
113     else: # Si aucun des mots clefs ne sont saisis dans l'input
114         print("Commande non reconnue. Veuillez réessayer. (inventaire, entrée, sortie, valeur)")
115
```

Résultat

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\mncar\OneDrive\Documents\QG\_IRIS\PORTFOLIO\V8> &
/Projet_algo_S2_A_Martins.py"
Que voulez-vous faire ? (inventaire, entrée, sortie, valeur)

Votre choix : entrée
Quel produit à ajouter ? coco
Quantité à ajouter ? 5
Vous avez ajouté 5 coco(s). Nouveau stock : 11
```

```
Que voulez-vous faire ? (inventaire, entrée, sortie, valeur)

Votre choix : xxx
Commande non reconnue. Veuillez réessayer. (inventaire, entrée, sortie, valeur)

Que voulez-vous faire ? (inventaire, entrée, sortie, valeur)

Votre choix : █
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\mncar\OneDrive\Documents\QG\_IRIS\PORTFOLIO\V8> &
/Projet_algo_S2_A_Martins.py"
Votre choix : inventaire
Stock au 21/01/2026 21:26:50 :
tomate : 4
coco : 11
pomme : 3
kiwi : 8
Stock de tomate bas, achat nécessaire. (Seuil : 5)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\mncar\OneDrive\Documents\QG\_IRIS\PORTFOLIO\V8> &
/Projet_algo_S2_A_Martins.py"
Que voulez-vous faire ? (inventaire, entrée, sortie, valeur)

Votre choix : valeur

Valeur du stock par produit :
- tomate : 4 x 0.5 € = 2.00 €
- coco : 11 x 0.8 € = 8.80 €
- pomme : 3 x 0.7 € = 2.10 €
- kiwi : 8 x 1.5 € = 12.00 €

Valeur totale du stock : 24.90 €
```