

Travaux pratique
La programmation en Python
Révisions et consolidations

BTS SN-IR
2^{ème} année
Page 1 sur 3

Exercice1 : Trouver la valeur maximale dans une liste en Python

Écrire une fonction en Python appelée **trouver_max** qui prend en paramètre une liste d'entiers et retourne la valeur maximale présente dans la liste. Si la liste est vide, la fonction devrait retourner None.

Exemple d'utilisation :

```
ma_liste = [3, 8, 2, 10, 5]
valeur_maximale = trouver_max(ma_liste)

print("La valeur maximale dans la liste est :", valeur_maximale)
```

Contraintes :

- La fonction doit être définie comme suit : **def trouver_max(liste):**
- Si la liste est vide, la fonction doit retourner None.
- Utilisez une boucle pour itérer à travers les éléments de la liste et trouver la valeur maximale.
- Vous ne pouvez pas utiliser de fonctions prédéfinies de Python telles que max() pour résoudre cet exercice. La logique doit être implémentée manuellement.

Exercice 2: Calcul de la moyenne d'une liste

Écrire une fonction en Python appelée **calculer_moyenne** qui prend en paramètre une liste de nombres et retourne la moyenne de ces nombres. Si la liste est vide, la fonction devrait retourner None.

Exemple d'utilisation :

```
ma_autre_liste = [15, 7, 22, 45, 9, 12]
valeur_maximale_autre_liste = trouver_max(ma_autre_liste)

print("La valeur maximale dans la liste est :", valeur_maximale_autre_liste)
```

Contraintes :

- La fonction doit être définie comme suit : **def calculer_moyenne(liste):**
- Si la liste est vide, la fonction doit retourner None.
- Utilisez une boucle pour itérer à travers les éléments de la liste et calculer la somme de ces éléments.
- Ensuite, divisez la somme par le nombre d'éléments dans la liste pour obtenir la moyenne.
- Vous ne pouvez pas utiliser de fonctions prédéfinies de Python telles que sum() ou len() pour résoudre cet exercice. La logique doit être implémentée manuellement.

Exercice 3: Fusion de deux listes triées

Écrire une fonction en Python appelée **fusionner_listes** qui prend en paramètre deux listes triées par ordre croissant et retourne une nouvelle liste qui est la fusion des deux listes tout en maintenant l'ordre croissant. Si l'une des listes est vide, la fonction devrait retourner l'autre liste telle quelle.

Exemple d'utilisation :

```
liste1 = [1, 3, 5, 7]
liste2 = [2, 4, 6, 8, 10]
resultat = fusionner_listes(liste1, liste2)

print("Liste fusionnée :", resultat)
```

Contraintes :

La fonction doit être définie comme suit : **def fusionner_listes(liste1, liste2):**

Si l'une des listes est vide, la fonction doit retourner l'autre liste telle quelle.

Utilisez une approche itérative pour fusionner les deux listes tout en maintenant l'ordre croissant.

N'utilisez pas de fonctions prédéfinies de Python pour la fusion ou le tri des listes.

Exercice 4: Compter les occurrences d'éléments dans une liste

Écrire une fonction en Python appelée **compter_occurrences** qui prend en paramètre une liste et retourne un dictionnaire indiquant le nombre d'occurrences de chaque élément dans la liste.

Exemple d'utilisation :

```
ma_liste = [1, 2, 3, 1, 2, 4, 5, 1, 2, 3]
occurrences = compter_occurrences(ma_liste)

print("Occurrences des éléments dans la liste :", occurrences)
```

Le résultat attendu pour cet exemple est un dictionnaire indiquant le nombre d'occurrences de chaque élément dans la liste.

Contraintes :

- La fonction doit être définie comme suit : **def compter_occurrences(liste):**
- Utilisez un dictionnaire pour stocker le nombre d'occurrences de chaque élément.
- Parcourez la liste pour compter le nombre d'occurrences de chaque élément.
- Le dictionnaire résultant doit être structuré de la manière suivante : {élément: nombre d'occurrences}.

Exercice 5: Somme des carrés des nombres pairs

Écrivez un programme en Python qui demande à l'utilisateur de saisir une plage de nombres (début et fin), puis calcule et affiche la somme des carrés des nombres pairs dans cette plage.

Voici un exemple de déroulement attendu du programme :

```
Entrez le début de la plage : 2
Entrez la fin de la plage : 10
La somme des carrés des nombres pairs de 2 à 10 est : 220
```

Consignes :

Utilisez une boucle pour parcourir les nombres dans la plage spécifiée.
Utilisez une instruction conditionnelle pour vérifier si un nombre est pair.
Calculez le carré des nombres pairs et ajoutez-les à une variable de somme.
Affichez le résultat final.

Conseils :

Utilisez la fonction range pour générer une séquence de nombres dans la plage spécifiée.
Utilisez l'opérateur modulo (%) pour vérifier si un nombre est pair.

Exercice 6: Vérification de palindromes

Écrivez un programme en Python qui demande à l'utilisateur d'entrer un mot et détermine si le mot est un palindrome ou non. Un palindrome est un mot qui se lit de la même manière de gauche à droite et de droite à gauche.

Consignes :

1. Demandez à l'utilisateur d'entrer un mot en utilisant la fonction input.
2. Utilisez une boucle pour comparer les caractères du mot de gauche à droite et de droite à gauche simultanément.
3. Affichez un message indiquant si le mot est un palindrome ou non.

Exemple de déroulement attendu du programme :

```
Entrez un mot : radar
Le mot "radar" est un palindrome.
```

Conseils :

- Vous pouvez utiliser l'indexation négative pour accéder aux caractères de la fin vers le début d'une chaîne.
- Assurez-vous de gérer correctement les lettres majuscules et minuscules (par exemple, "Radar" devrait également être considéré comme un palindrome).

