

Les listes en Python : Guide complet

Les listes sont l'une des structures de données les plus utilisées en Python. Elles permettent de stocker une collection d'éléments, qui peuvent être de types différents (entiers, chaînes, flottants, etc.). Voici un aperçu détaillé de leurs caractéristiques et fonctionnalités :

1. Création d'une liste

Une liste est définie en plaçant des éléments entre crochets [], séparés par des virgules.

```
# Liste vide
ma_liste = []

# Liste avec des éléments
ma_liste = [1, 2, 3, "Python", 4.5]
```

2. Propriétés des listes

- **Ordonnées** : Les éléments conservent leur ordre d'insertion.
- **Modifiables** : Vous pouvez ajouter, supprimer ou modifier des éléments.
- **Hétérogènes** : Une liste peut contenir des types de données différents.

3. Opérations de base

Accès aux éléments

Les listes utilisent des indices (commençant à 0) pour accéder aux éléments.

```
ma_liste = [10, 20, 30, 40]
print(ma_liste[0]) # Affiche 10
print(ma_liste[-1]) # Affiche 40 (indice négatif pour accéder à l'envers)
```

Modification d'un élément

```
ma_liste[1] = 25
print(ma_liste) # [10, 25, 30, 40]
```

Ajout d'éléments

- **Avec** `append` : Ajoute un élément à la fin.
- **Avec** `insert` : Ajoute un élément à une position spécifique.

```
ma_liste.append(50)      # [10, 25, 30, 40, 50]
ma_liste.insert(2, 15)   # [10, 25, 15, 30, 40, 50]
```

Suppression d'éléments

- **Avec** `remove` : Supprime la première occurrence d'une valeur.
- **Avec** `pop` : Supprime un élément par son indice (par défaut, le dernier).
- **Avec** `del` : Supprime un élément ou une tranche.

```
ma_liste.remove(25)      # [10, 15, 30, 40, 50]
ma_liste.pop(2)          # [10, 15, 40, 50]
del ma_liste[1]           # [10, 40, 50]
```

4. Slicing (Tranches)

Vous pouvez extraire une partie d'une liste avec des indices.

```
ma_liste = [0, 1, 2, 3, 4, 5]
print(ma_liste[1:4])    # [1, 2, 3]
print(ma_liste[:3])      # [0, 1, 2]
print(ma_liste[::2])     # [0, 2, 4] (saut de 2)
```

5. Fonctions utiles

- `len` : Renvoie la taille de la liste.
- `sorted` : Renvoie une copie triée de la liste.
- `sum` : Calcule la somme des éléments (si numériques).
- `min` / `max` : Renvoie le plus petit / grand élément.

```
ma_liste = [3, 1, 4, 1, 5]
print(len(ma_liste))    # 5
```

```
print(sorted(ma_liste))    # [1, 1, 3, 4, 5]
print(sum(ma_liste))       # 14
print(min(ma_liste))       # 1
```

6. Boucles sur une liste

Vous pouvez parcourir une liste avec une boucle for.

```
for element in ma_liste:
    print(element)
```

7. Listes imbriquées

Les listes peuvent contenir d'autres listes.

```
matrice = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(matrice[1][2]) # 6
```

8. Compréhensions de liste

Une manière concise de créer des listes.

```
carres = [x**2 for x in range(5)]
print(carres) # [0, 1,
```