

## **Objectif**

Comprendre comment la gestion de la mémoire et les opérations de copie influencent le temps de calcul lors de l'utilisation des listes en Python.

## **Expérience : Mesurer le temps de calcul avec append et copier une liste**

Nous allons :

1. Créer une liste vide.
2. Ajouter des éléments à la liste avec append et mesurer le temps pris pour chaque ajout.
3. Comparer le temps pris pour agrandir la liste par ajout simple, et pour créer une copie de la liste avec copy().

## **Code de l'Expérience**

Vous pouvez utiliser le code suivant pour l'expérience. On utilise la bibliothèque time pour mesurer le temps de chaque opération.

```
import time

# Initialisation de la liste vide et des variables
taille_max = 10**6 # nombre d'éléments à ajouter
temps_append = []
temps_copy = []
liste = []

# Boucle pour ajouter des éléments et mesurer le temps
for i in range(taille_max):
    # Mesurer le temps pour un `append`
    debut_append = time.time()
    liste.append(i)
    fin_append = time.time()
    temps_append.append(fin_append - debut_append)

    # Mesurer le temps pour une copie complète
    debut_copy = time.time()
    copie_liste = liste.copy()
    fin_copy = time.time()
    temps_copy.append(fin_copy - debut_copy)

# Calcul de la moyenne des temps d'ajout et de copie
temps_moyen_append = sum(temps_append) / len(temps_append)
temps_moyen_copy = sum(temps_copy) / len(temps_copy)

print("Temps moyen pour un `append` :", temps_moyen_append)
print("Temps moyen pour une copie:", temps_moyen_copy)
```

## Explications et Observations à Attendre

1. **Opération append** : Au début, le temps pris pour chaque append est minime. Cependant, à mesure que la liste grandit, la mémoire doit parfois être réallouée pour agrandir la liste, ce qui prend plus de temps. Cette réallocation n'est pas constante et varie en fonction de la taille de la liste et de la mémoire disponible.
2. **Opération de copie** : La fonction `copy()` crée une copie entière de la liste, ce qui devient de plus en plus coûteux au fur et à mesure que la liste grandit. La copie est proportionnelle à la taille de la liste, donc pour les grandes listes, cette opération devient significativement plus lente.

## Analyse des Résultats

- **Temps d'append** : Le temps pour chaque ajout n'est pas constant et augmente légèrement lorsque des réallocations de mémoire se produisent. Cela peut être observé par des "pics" dans les temps d'exécution.
- **Temps de copie** : Plus la liste est grande, plus le temps de copie augmente de manière quasi linéaire, car chaque élément doit être dupliqué en mémoire.