1re

Programme de spécialité mathématiques de première générale — arrêté du 26 février 2026, BO n° 14 du 2 avril 2026 — MENE2602917A

Relue le 14 novembre 2026

COURS

Algorithmique — listes en Python

Les termes d'une suite, les résultats d'une simulation, les notes d'une classe : en mathématiques, on manipule souvent des **séries de valeurs**. En Python, on les range dans une **liste**, que l'on peut créer, parcourir et transformer en quelques lignes.

1 Créer une liste

DÉFINITION

Une liste est une suite ordonnée de valeurs entre crochets : `L = [3, 7, 1]`. On accède à un élément par son **indice**, qui commence à **0** : `L[0]` vaut 3, `L[2]` vaut 1. `len(L)` donne le nombre d'éléments.

RÈGLE

On construit une liste de trois façons : **en extension** (`[2, 4, 6]`), **par ajouts successifs** (`L = []` puis `L.append(x)` dans une boucle), ou **en compréhension** (`[k**2 for k in range(5)]` donne `[0, 1, 4, 9, 16]`).

`range(n)` produit les entiers de 0 à $n-1$: `range(5)` s'arrête à 4. C'est la source la plus fréquente d'erreurs de décalage.

2 Parcourir une liste

RÈGLE

On parcourt les **éléments** avec `for x in L:` ou les **indices** avec `for i in range(len(L)):`. Cela permet de calculer une somme, une moyenne, de chercher un maximum ou de compter les éléments qui vérifient une condition.

EXEMPLE

Écrire une fonction qui renvoie la liste des n premiers termes de la suite définie par $u_0 = 2$ et $u_{n+1} = 3u_n - 1$.

1. On part de `L = [2]`.
2. À chaque étape, on ajoute `3 * L[-1] - 1` (`L[-1]` est le dernier élément).
3. On répète $n-1$ fois.

`def termes(n):` / `L = [2]` / `for k in range(n - 1):` / `L.append(3 * L[-1] - 1)` / `return L` — `termes(4)` renvoie `[2, 5, 14, 41]`.

3 Écrire un algorithme sur une liste

ÉCRIRE UN ALGORITHME SUR UNE LISTE

1. Préciser ce que l'on veut obtenir (une valeur, une nouvelle liste).
2. Initialiser la variable résultat (`s = 0`, `m = L[0]`, `R = []`).
3. Parcourir la liste et mettre à jour le résultat à chaque élément.
4. Renvoyer le résultat et tester sur un petit exemple calculé à la main.

À RETENIR

- Liste : `L = [3, 7, 1]`, indices à partir de **0**, `L[-1]` = dernier, `len(L)` = longueur.
- Construire : extension, `append` dans une boucle, compréhension `[f(k) for k in range(n)]`.
- Parcourir avec `for x in L` ; initialiser, mettre à jour, renvoyer, tester.