COURS

Tle

La récursivité

Programme de terminale générale, enseignement de spécialité — BO spécial n°8 du 25 juillet 2019

Relue le 14 novembre 2026

Une poupée russe contient une poupée plus petite, qui en contient une autre, jusqu'à la plus petite. Une fonction récursive fonctionne de la même façon : elle s'appelle elle-même sur un problème plus petit, jusqu'à un cas simple qu'elle sait résoudre directement.

1 Principe

DÉFINITION

Fonction qui s'appelle elle-même dans sa définition.

DÉFINITION

Cas simple résolu sans appel récursif ; il garantit que la fonction s'arrête.

RÈGLE

Exemple : `def fact(n): return 1 if n == 0 else n * fact(n - 1).`

2 La pile d'appels

RÈGLE

Chaque appel est mis en attente sur la **pile d'appels** jusqu'à ce que l'appel suivant renvoie son résultat. Sans cas de base, ou si les appels ne s'en rapprochent pas, la pile déborde : Python lève une `RecursionError` (limite de l'ordre de 1 000 appels par défaut).

3 Quand l'utiliser

RÈGLE

La récursivité est naturelle pour les structures elles-mêmes récursives (arbres, listes chaînées) et pour les algorithmes **diviser pour régner** (tri fusion, exponentiation rapide). Certains calculs récursifs naïfs recalculent les mêmes valeurs (Fibonacci) : la **mémoïsation** ou la programmation dynamique les accélèrent.

EXEMPLE

Écris une fonction récursive somme(n) = 1 + 2 + … + n.

1. Cas de base : `if n == 0: return 0.`
2. Appel récursif : `return n + somme(n - 1).`
3. Vérifier : somme(3) = 3 + 2 + 1 + 0 = 6.

Une fonction récursive combine un cas de base et un appel sur un problème plus petit.

LE PIÈGE QUI COÛTE LE PLUS DE POINTS

Oublier le cas de base. — La fonction s'appelle sans fin et la pile d'appels déborde (RecursionError).

Le réflexe : Écrire le cas de base en premier.

ÉCRIRE UNE FONCTION RÉCURSIVE

1. Identifier le cas de base et sa valeur.
2. Exprimer le problème à l'aide du même problème plus petit.
3. Vérifier que chaque appel se rapproche du cas de base.
4. Tester à la main sur une petite valeur.

À VÉRIFIER

- Ai-je défini le vocabulaire ?
- Mon code est-il testé sur un exemple ?
- Ai-je traité les cas limites ?
- Ai-je justifié la complexité ou le résultat ?

À RETENIR

- **Fonction récursive** ; **cas de base** obligatoire.
- **Pile d'appels** ; `RecursionError`.
- Usages : arbres, diviser pour régner ; **mémoïsation** contre les recalculs.