

TP: boucle for

Pour chaque exercice vous testerez votre fonction, écrirez un jeu de test et remplirez la docstring.

Exercice 1

Écrire une fonction `somme_pairs(n)` qui calcule la somme des nombres pairs de 1 à n à l'aide d'un accumulateur.

Exercice 2

Écrire une fonction `somme_carres(n)` qui calcule la somme des carrés des entiers de 1 à n (ex: $1^2 + 2^2 + \dots + n^2$).

Exercice 3

Écrire une fonction `somme_multiples_3_ou_5(n)` qui calcule la somme des entiers de 1 à n qui sont multiples de 3 ou de 5.

Exercice 4

Écrire une fonction `nombre_multiple_3(n)` qui retourne le nombre d'entiers compris entre 1 et n qui sont des multiples de 3.

Exercice 5

Écrire une fonction `somme_chiffres(n)` qui calcule la somme des chiffres d'un nombre entier n (ex: $123 \rightarrow 1 + 2 + 3 = 6$) à l'aide d'un accumulateur

Exercice 6

Écrire une fonction `est_parfait(n)` qui vérifie si un nombre n est parfait (un nombre est parfait si la somme de ses diviseurs propres est égale à lui-même, ex: $6 = 1 + 2 + 3$).

Exercice 7

Écrire une fonction `calculer_pi(n)` qui approxime la valeur de π en utilisant la formule de Leibniz $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + \frac{(-1)^n}{2n+1}$ avec n itérations. Calculer cette somme puis retourne son quadruple.