

Plaquette 2 - Conditions, boucles while et seuils

Algorithmique et Python — Terminale spé maths

Exercices

Méthode

Une boucle *while* répète un bloc **tant qu'**une condition est vraie : on l'utilise quand le nombre de tours n'est **pas connu** à l'avance, typiquement pour chercher un **seuil** (« le premier n tel que... »). Trois points à vérifier : la condition est le **contraire** de l'objectif, les variables sont mises à jour **dans** la boucle, et la boucle **se termine** (justification mathématique : limite, monotonie).

Exercice 1 - Une fonction avec des conditions

On considère la fonction suivante.

```
def mention(note):  
    if note >= 16:  
        return "TB"  
    elif note >= 14:  
        return "B"  
    elif note >= 12:  
        return "AB"  
    elif note >= 10:  
        return "Passable"  
    else:  
        return "Refusé"
```

Que renvoient les appels avec 17, 14, 11,5 et 9 ? Pourquoi l'ordre des tests est-il important ?

Exercice 2 - Compter avec une condition

Écrire un script qui compte les entiers de 1 à 99 multiples de 3 ou de 5. Vérifier le résultat par un raisonnement.

Exercice 3 - Dépasser un million

Écrire un script qui détermine le plus petit entier n tel que $2^n > 10^6$, puis vérifier avec le logarithme.

Exercice 4 - Un seuil vers la limite

La suite (u_n) est définie par $u_0 = 50$ et $u_{n+1} = 0,9u_n + 1$; elle décroît vers 10.

- Écrire un script qui renvoie le premier rang tel que $u_n < 11$.
- Retrouver le résultat par le calcul, sachant que $u_n = 10 + 40 \times 0,9^n$.

Exercice 5 - Une somme qui dépasse 5

Écrire un script qui détermine le plus petit n tel que $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} > 5$.

Exercice 6 – Une boucle qui ne s’arrête pas

Un élève veut le premier rang où $u_n \geq 10$, avec $u_0 = 0$ et $u_{n+1} = 0,5u_n + 3$. Il écrit :

```
n = 0
u = 0
while u < 10:
    u = 0.5 * u + 3
    n = n + 1
print(n)
```

Le programme ne s’arrête jamais. Expliquer pourquoi.

Exercice 7 – Années bissextiles

Une année est bissextile si elle est divisible par 4, sauf les années séculaires (divisibles par 100) qui ne le sont que si elles sont divisibles par 400.

Écrire une fonction *bissextile(a)* renvoyant *True* ou *False*, et la tester sur 2024, 1900 et 2000.

Exercice 8 – Le temps de doublement

Une population augmente de 2 % par an. Écrire un script qui détermine au bout de combien d’années elle aura doublé, puis vérifier.

Exercice 9 – Nombres parfaits

Un entier $n \geq 2$ est **parfait** s’il est égal à la somme de ses diviseurs autres que lui-même (par exemple $6 = 1 + 2 + 3$).

Écrire un script qui affiche les nombres parfaits inférieurs à 1 000.

Exercice 10 – Approcher une limite à 0,001 près

La suite $u_n = 3 + \frac{2}{n}$ ($n \geq 1$) tend vers 3. Écrire un script qui détermine le premier n tel que $|u_n - 3| < 10^{-3}$, puis vérifier.

Exercice 11 – L’algorithme d’Euclide

On considère la fonction suivante.

```
def pgcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a
```

- Faire tourner la fonction à la main pour $a = 1\,071$ et $b = 462$.
- Pourquoi la boucle se termine-t-elle ?

Exercice 12 – Factorielle et milliard

Écrire un script qui détermine le plus petit entier n tel que $n! > 10^9$.

Exercice 13 – Deux conditions d’arrêt

Que fait ce script ? Que contiennent n et u à la fin ?

```
u = 1
n = 0
while u < 100 and n < 50:
    u = u + 0.5 * n
    n = n + 1
print(n, u)
```

Exercice 14 – Chercher un minimum par balayage

Écrire un script qui parcourt $[0 ; 5]$ avec un pas de 0,01 et renvoie l’abscisse et la valeur du minimum de $f(x) = x^2 - 4x$. Vérifier par le calcul.

Exercice 15 – Concentration d’un médicament

Un patient reçoit une dose toutes les 6 heures. Juste après la n -ième dose, la concentration vaut C_n (en mg/L), avec $C_1 = 10$ et $C_{n+1} = 0,7 C_n + 10$.

- Écrire un script qui détermine après quelle dose la concentration dépasse 30 mg/L.
- Vers quelle valeur la concentration tend-elle ?