



L'essentiel

- En Python, pour répéter une instruction TANT QUE une condition est vérifiée, on utilise l'instruction **while** :

Algorithme :

```

a ← 1
b ← 2
TANT QUE a × b ≤ 100 FAIRE :
    | a ← 2 × a
    | b ← b + 1
FIN TANT QUE
AFFICHER (a,b).
```

Code Python :

```

a = 1
b = 2
while a*b < 100:
    a = 2*a
    b = b+1
print(a, b)
```

- L'ensemble des instructions à effectuer est indenté comme d'habitude.
- Ne confondez pas le **if** qui n'effectue les instructions qu'une fois et le **while** qui répète ces instructions. La structure "Si" n'est pas une boucle!
- Attention : dans une fonction contenant un **while**, dès qu'un **return** est rencontré, on quitte la boucle (et la fonction) sans exécuter la suite des instructions.
- Attention aux boucles infinies avec les **while**!

Exercices du jour

Exercice 1: Exécuter à la main, l'algorithme de début de cours, et prédire ce qui sera affiché. Vérifier ensuite en tapant le code Python correspondant.

Exercice 2: On veut créer le programme suivant : l'ordinateur choisit un nombre entier au hasard entre 1 et 100 puis demande à l'utilisateur de rentrer des nombres entiers jusqu'à deviner le nombre choisi par l'ordinateur. (*Dans un premier temps, on peut forcer le nombre choisi pour tester son programme et ne pas le laisser au hasard!*)

- Créer un script effectuant cette démarche.
- Modifier le script pour que le programme affiche, à chaque proposition de l'utilisateur, si le nombre donné est trop grand ou trop petit par rapport à celui cherché.
- Modifier le script pour que le programme n'accepte que 8 propositions au maximum.

Exercice 3: La suite de **Collatz**, ou suite de **Syracuse**, aurait été étudiée en premier lieu par Lothar Collatz, un mathématicien allemand au début des années 30. Il s'agit de construire une suite de nombres à partir d'un nombre donné n selon les règles suivantes :

- ▶ si n est pair, le nombre suivant est $\frac{n}{2}$;
- ▶ sinon, le nombre suivant est $3n + 1$.

La particularité de cette suite est que, quel que soit le nombre de départ, il semble que l'on finisse toujours par tomber sur 1, sans toutefois que l'on comprenne mathématiquement pourquoi. Actuellement, la conjecture a été vérifiée pour tous les nombres inférieurs à $5,7 \times 10^{17}$.

- Calculer à la main la suite de Collatz démarrant avec le nombre 5. Idem avec le nombre 13.
- Créer une fonction Python `collatz` qui prend en paramètre un nombre n et calcule le nombre suivant de la suite de Collatz.

3. En reprenant les exemples calculés à la main à la question 1, combien faut-il d'étapes pour arriver à 1 si on démarre avec 5? Avec 13?
4. Créer maintenant une fonction `etapes` qui prend en entrée un nombre `n` et calcule le nombre d'étapes nécessaires pour que la suite de Collatz arrive à 1 en démarrant avec `n`.
5. Tester la fonction avec 5, puis 13, puis 56572.

Exercices en autonomie

Exercice 4: On considère les deux fonctions ci-dessous :

```
def fonc1(x):
    """
    Entrée : x est un réel positif
    Sortie : Donne le plus grand entier n tel que
             1/n > x. Renvoie 0 si un tel n n'existe pas.
    """
    n = 1
    while (1/n < x) :
        n = n + 1
    if x >= 1:
        return 0
    else :
        return n
```

```
from random import *
def fonc2():
    """
    Entrée : Pas de paramètre
    Sortie : Donne le nombre de tirages uniformes
             d'un nombre aléatoire sur [0,1] qu'il a fallut
             pour obtenir un nombre plus petit que 0,9.
    """
    x=random()
    i=1
    while (x > 0.9) :
        i= i+ 1
    return i
```

1. Pour la première fonction, un élève qui a lancé la fonction n'a au bout d'une journée pas obtenu de résultat....
 - a. Donner un exemple d'appel possible de `fonc1` qui entraînerait ce problème.
 - b. Quelles sont les arguments qui posent problème?
 - c. Corriger la fonction.
2. Pour la seconde fonction :
 - a. Que taper dans la console pour appeler `fonc2`?
 - b. Expliquer pourquoi l'élève n'obtient pas de résultat et corriger le code.

Exercice 5: Une urne contient 3 boules numérotées 1 et 7 boules numérotées 0. On tire successivement et avec remise une boule de cette urne jusqu'à obtenir une boule numérotée 1.

1. Créer une fonction qui n'a pas de paramètre mais simule le tirage jusqu'à obtention d'une boule numérotée 1 et renvoie le nombre de tirage réalisés.
2. Tester cette fonction un grand nombre de fois et estimer le nombre de tirages moyens nécessaire à l'obtention d'une boule numérotée 1.

Aide pour les exercices

Solutions des exercices