

TP5

BOUCLES WHILE

À préparer : cours, exercices 1 et 2

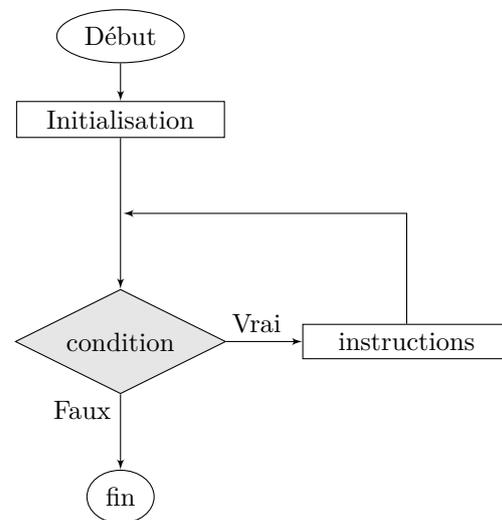
Notions utilisées dans ce TP

Ce TP utilise les parties suivantes du vade-mecum (en plus des TP précédents).

- III : bibliothèques math et random
- V : boucles while

On souhaite traduire un algorithme de ce type :

Initialisation
Tant que la condition est vraie,
 Réaliser les instructions
Fin



La syntaxe python pour écrire ce type d'instruction est la suivante :

```

initialisation
while condition :
    bloc_instructions
  
```

- Ne pas oublier l'indentation du bloc.
- La *condition* est un booléen (comme pour les if).
- Avec les boucles while, il y a un risque de créer une *boucle infinie*. Pour éviter cela, on respectera les règles suivantes.

La condition doit :

être initialisée : on doit pouvoir dire dès le départ si elle est vraie ou fausse. Les variables de la condition doivent donc être définies avant le while.

évoluer à chaque étape : au moins une des variables de la condition doit changer de valeur à chaque tour de boucle.

devenir fausse à un moment.

Exercice 1 Quelle est la valeur de la variable s à la suite de ces instructions ? On complètera le tableau. *Inutile de recopier ce programme sur l'ordinateur.*

```
s = 0
q = 21
i = 4
while q != 0 and i > 0:
    s = s + (q%2)*2**(4-i)
    q = q//2
    i = i-1
print(s)
```

s	q	i
0	21	4

Exercice 2 Exemple classique : "déterminer le premier n tel que ..."

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par $\begin{cases} u_0 = 0 \\ \forall n \in \mathbb{N}, u_{n+1} = u_n - 3n \end{cases}$

On peut montrer que $(u_n)_{n \in \mathbb{N}}$ est décroissante et tend vers $-\infty$.
Déterminer le plus petit entier $n \in \mathbb{N}$ tel que $u_n \leq -900$

```
u = 0
n = 0
while ..... :
    u = .....
    n = .....
print(...)
```

Compléter ce tableau pour vérifier que les premiers calculs sont bons et que la condition d'arrêt est cohérente.

u	n	Condition
$(u_0) : u = 0$	$n = 0$	vraie
$(u_1) : u =$		vraie
$(u_2) : u =$		vraie
...
Fin : $u = u_N$	$n = N$	Fausse, c'est-à-dire

réponse : $n = 25$.

Exercice 3 Transformer le programme suivant pour utiliser une boucle while.

```
for k in range(1,11) :
    print( k**2 )
```

Quelle version est la plus pertinente ici ?

Exercice 4 Soit $v_1 = 2$ et $v_{n+1} = \exp\left(\frac{v_n}{n+1}\right)$ pour $n \geq 1$.

1. Écrire une fonction `suite_v(n)` qui, étant donné n , renvoie la valeur de v_n .

2. Déterminer le plus grand $n \in \mathbb{N}$ tel que $v_n \geq 1,001$. On cherche le **dernier** n tel que ...

v	n	Condition
		vraie
		vraie
...
$v = v_{n-1}$	$n - 1$	vraie, donc
Fin : $v = v_n$	n	fausse, donc

3. Comment vérifier à l'aide de la fonction `suite_v(n)` ?

Exercice 5 Suite de Fibonacci. On souhaite modéliser une population de lapins. Soit L_n le nombre de couples de lapins au n -ième mois.

On suppose que $L_0 = 1$, $L_1 = 1$ et que pour $n \in \mathbb{N}$, $L_{n+2} = L_{n+1} + L_n$.

Déterminer au bout de combien de mois la population totale dépassera le million de lapins.

Exercice 6 Jeu du nombre mystère

Écrire un programme qui définit un nombre mystère aléatoire entre 1 et 100, puis demande à l'utilisateur de le deviner. Ce programme doit demander de saisir un nombre tant que le résultat n'a pas été trouvé et, à chaque étape, indiquera si le nombre mystère est plus grand ou plus petit que la proposition donnée.

