

```
s = experience()
n = 1
L = [s] # moyenne de
while n < nExperiences:
    n = n+1
    s = s + experience()
    L.append(s/n) # s
plt.plot(list(range(1,
nExperiences+1)),
nExperiences)
```

# FICHE n°6 : PROGRAMMER DES BOUCLES

## Syntaxe :

Langage naturel	Python
<b>Tant que</b> Condition est vraie Instructions	<b>while</b> Condition: Instructions
<b>Fin Tant que</b>	

<b>Pour</b> i allant de 3 à 7 Instructions	<b>for</b> i in range(3,8): Instructions
<b>Fin Pour</b>	

En Python, **range(3,8)** désigne la séquence des entiers  $n$  vérifiant  $3 \leq n < 8$ .  
**range(5)** désigne la séquence des entiers 0, 1, ..., 4.

Sortir de la boucle	break
Générer un nombre entier aléatoire de l'intervalle [1;5]	randint(1,5)

## Rappels :

<b>Si</b> Condition <b>Alors</b> Instructions1	<b>if</b> condition: Instruction1
<b>Sinon</b> Instructions2	<b>else:</b> Instruction2
<b>Fin Si</b>	

## Exercice 1 :

- 1) Tester le programme ci-contre. Qu'affiche-t-il en sortie ?
- 2) Ecrire et tester un programme qui affiche tous les entiers inférieurs à 16.
- 3) Ecrire et tester un programme qui affiche tous les entiers compris entre 18 et 45.

```
for i in range(10):
    print(i)
```

## Exercice 2 :

- 1) Tester le programme ci-contre. Qu'affiche-t-il en sortie ?
- 2) Ecrire et tester un programme qui affiche tous les entiers pairs compris entre 18 et 45.
- 3) Ecrire et tester un programme qui affiche tous les entiers impairs compris entre 50 et 150.

```
n=0
while n<10:
    print(n)
    n=n+2
```

## Exercice 3 :

- 1) On donne le programme ci-contre. Recopier et compléter le tableau suivant par les premières valeurs prises par les variables S et i.

```
s=0
for i in range(101):
    s=s+i
    print(s)
```

i		1	2	3													
S	0	1	3														

- 2) Quel problème permet de résoudre cet algorithme.
- 3) a) En s'inspirant des programmes précédents, écrire et tester un programme permettant de calculer la somme des entiers de 34 à 145.  
b) Même question pour la somme des entiers de 67 à 456.

## Exercice 4 :

On place un capital de 500€ sur un compte rémunéré à 3% par an.  
L'algorithme ci-contre, écrit en langage naturel, permet de calculer le nombre d'années au bout desquelles le capital sera doublé.

Affecter à S la valeur 500  
Affecter à A la valeur 0  
**Tant que** S<1000  
Affecter à S la valeur 1,03xS  
Affecter à A la valeur A+1  
**Fin Tant que**  
Afficher A

- 1) Le programme ci-contre traduisant l'algorithme précédent comprend une erreur. Corriger le programme et le tester.
- 2) Modifier le programme précédent de telle sorte que le capital et le taux de rémunération soient saisis en entrée. Le tester dans un nouveau contexte à décrire.

```
s=500
a=0
while s<1000:
    s=1.03*s
    a=a+1
print(a)
```

## Exercice 5 :

On dépose 25€ dans une tirelire.  
L'algorithme suivant, écrit en langage naturel, permet de calculer le nombre de pièces de 1€ ou 2€ ajoutés de façon aléatoire dans la tirelire avant de dépasser 50€.

- 1) Compléter l'algorithme.
- 2) Pourquoi le programme affiche en sortie « D-1 » ?
- 3) Ecrire et tester un programme traduisant cet algorithme.

Affecter à S la valeur 25  
Affecter à D la valeur 0  
**Tant que** S<...  
Affecter à A la valeur aléatoire 1 ou 2  
Affecter à S la valeur ...  
Affecter à D la valeur D+1  
Afficher A  
**Fin Tant que**  
Afficher D-1

## Exercice 6 :

D'après "Document ressource pour la classe de seconde" – juin 2009

On demande à l'utilisateur de deviner en moins de six essais un nombre tiré au hasard entre 10 et 100.  
On lui indique à chaque fois si le nombre proposé est supérieur ou inférieur au nombre cherché.

- 1) L'algorithme qui suit, écrit en langage naturel, permet d'effectuer le jeu.
  - a) Que représentent les variables E, S et N ?
  - b) Ecrire et tester un programme traduisant cet algorithme.

2) Sans stratégie, il est difficile de gagner. En effet, selon le choix des valeurs, il sera ou non possible de déterminer à coup sûr la solution.

La méthode consiste, en choisissant à chaque fois la valeur située au milieu de l'intervalle en cours, à réduire de moitié l'amplitude de l'intervalle dans lequel se trouve le nombre.

Tester cette stratégie pour gagner à tous les coups à ce jeu et détailler un coup gagnant.

```
Affecter à S la valeur d'un nombre
aléatoire entier compris entre 10 et 100
Affecter à E la valeur 1
Tant que E<7
  Saisir N
  Si N>S
    Alors afficher "C'est moins"
  Sinon
    Si N<S
      Alors afficher "C'est plus"
    Sinon
      Afficher "C'est gagné"
      Sortir de la boucle
  Fin Si
Fin Si
Affecter à E la valeur E+1
Fin Tant que
Si E=7
  Alors afficher "C'est perdu"
Fin Si
```

### **Exercice 7 :**

Ecrire et tester un programme permettant de calculer la somme des entiers naturels pairs inférieurs ou égaux à 1000.

### **Exercice 8 :**

Ecrire et tester un programme permettant de calculer la puissance d'un nombre.

### **Exercice 9 :**

1) Ecrire un algorithme qui permet de déterminer la moyenne d'une série de nombres générés de façon aléatoire. On pourra utiliser une boucle **Pour**.

2) Ecrire et tester un programme traduisant cet algorithme.



Hors du cadre de la classe, aucune reproduction, même partielle, autres que celles prévues à l'article L 122-5 du code de la propriété intellectuelle, ne peut être faite de ce site sans l'autorisation expresse de l'auteur.

[www.maths-et-tiques.fr/index.php/mentions-legales](http://www.maths-et-tiques.fr/index.php/mentions-legales)