

Tout comme dans le premier TP, il est conseillé de tester le maximum d'exemples.

# 1 Boucle For

## 1.1 Description

Une des structures de répétition proposées en Python est la boucle `for`, aussi appelée « boucle bornée » ou « non conditionnelle ». On décide en amont du nombre d'itérations effectuées par la boucle.

```
for i in range(n):  
    Instructions
```

Ici, les instructions seront répétées  $n$  fois. Concrètement, `range(n)` crée une pseudo-liste de valeurs :  $\{0, 1, 2, \dots, n-1\}$  qui est parcourue par la variable  $i$ .

Plus généralement, on peut utiliser la syntaxe suivante, dans laquelle la variable `element` va prendre toutes les valeurs contenues dans la variable composée `objet`.

```
for element in objet:  
    Instructions
```

### ■ Exemple 1:

```
for car in "Bonjour":  
    print(4*car)
```

#### Remarque 1 (`range()`).

- `range(n)` renvoie une séquence d'entiers entre 0 et  $n-1$  compris (donc  $n$  valeurs)
- `range(d, f)` renvoie une séquence d'entiers entre  $d$  et  $f-1$  (donc  $f-d$  valeurs)
- `range(d, f, p)` renvoie une séquence d'entiers entre  $d$  et  $f-1$  avec un pas de  $p$ .

### ► Exercice 1

Tester les syntaxes suivantes :

1. 

```
for i in range(7):  
    print(i)
```

2. 

```
for i in range(2,7):  
    print(i)
```

3. 

```
for i in range(2,13,3):  
    print(i)
```

#### Remarque 2 (Interrompre une boucle).

L'instruction `break` permet d'interrompre une boucle.

## ► Exercice 2

Tester le programme suivant :

```
for i in range(1000):  
    print(i)  
    if i>5 :  
        break
```

## 1.2 Exercices d'application

### ► Exercice 3

Écrire un programme qui va afficher :

```
Ligne 1  
Ligne 2  
...  
Ligne 20
```

### ► Exercice 4

Écrire un programme qui va afficher le terme de rang 50 de la suite  $(u_n)$  définie par  $u_0 = 1$  et  $u_{n+1} = 0,8 * u_n + 5$ .

### ► Exercice 5

Écrire un programme qui va afficher la somme  $S = u_0 + u_1 + u_2 + \dots + u_{50}$  des termes de la suite définie dans l'exemple précédent.

### ► Exercice 6

Écrire un programme qui demande à l'utilisateur de rentrer un certain nombre (demandé à l'utilisateur) de valeurs consécutivement puis qui va afficher la somme des valeurs entrées puis la moyenne des valeurs entrées.

## 2 Boucle While

### 2.1 Description

```
while condition:  
    instructions
```

Ici, la condition, comme dans le cas d'une structure conditionnelle sera un True ou un False. Le bloc d'instructions qui suit est exécuté tant que la condition est vraie.

#### ■ Exemple 2:

```
while a>=b:  
    a=a-b  
    q=q+1
```

Dans cet exemple, la variable  $q$  est incrémentée à chaque répétition. Si on l'initialise en début de programme, elle aura comme valeur finale le nombre de répétitions de la boucle. On l'appelle **compteur**. On en utilise très fréquemment. On note parfois cette ligne :  $q+=1$  pour décrire l'incrément.

Dans l'exemple précédent, à quoi correspond la valeur  $q$  en fin de boucle ?

## 2.2 Exercices d'application

### ► Exercice 7

On définit une suite géométrique de raison  $q = 1,04$  et de premier terme 1. Rédiger un algorithme permettant de connaître le rang à partir duquel la valeur du terme dépasse le double du premier terme. Faire varier la valeur du premier terme. Que constate-t-on ?

### ► Exercice 8

Soit la suite  $(u_n)$  définie par  $u_0 = 1$  et  $u_{n+1} = 0,8 * u_n + 5$ .

Écrire un programme qui va afficher le premier rang à partir duquel  $u_n$  va dépasser 20.

### ► Exercice 9

Interpréter le programme suivant. La fonction `random()` permet de générer un nombre aléatoire entre 0 et 1. Donc la commande `floor(6*random() + 1)` génère un nombre entier aléatoire entre 1 et 6.

```
from random import random
de=1
q=0
while de != 6:
    de=floor(6*random() + 1)
    q=q+1
print(q)
```

### ► Exercice 10

Entrer le programme suivant :

```
a=int(input("Entrer la valeur de a : "))
b=int(input("Entrer la valeur de b : "))
while a>=b:
    a=a-b
    q=q+1
print(q,a)
```

À quoi correspondent les valeurs  $q$  et  $r$  par rapport à  $a$  et  $b$  ?

### ► Exercice 11

On définit la suite de Syracuse par le procédé suivant. Étant donnée une valeur initiale  $u_0$  fixée, pour

$$\text{tout } n \in \mathbb{N}, u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } n \text{ est pair} \\ 3u_n + 1 & \text{sinon} \end{cases}$$

Écrire un programme qui permet de déterminer le premier rang à partir duquel la suite atteint la valeur 1, en prenant des valeurs initiales entre 2 et 20.