

MATHÉMATIQUES

Python : les boucles

1 La boucle bornée for

Exercice 1

Il est parfois utile dans un programme de répéter une ou plusieurs instructions un nombre fini de fois.

| | | | |
|--------------|--|--|---|
| Instructions | <pre>1 for i in range (3): 2 print(i)</pre> | <pre>1 for i in range (12,16): 2 print(i)</pre> | <pre>1 for i in range (5,15,3): 2 print(i)</pre> |
| Affichage | <p>La variable i prend les valeurs entières de 0 à 2, donc 3 valeurs.</p> <pre>>>> 0 1 2</pre> | <p>La variable i prend les valeurs entières de 12 à 15.</p> <pre>>>> 12 13 14 15</pre> | <p>La variable i prend les valeurs de 5 à 15 avec un pas de 3.</p> <pre>>>> 5 8 11 14</pre> |

1. Que fait ce script ?
- ```
1 s=0
2 for i in range (101):
3 s=s+i
4 print(s)
```
2. On considère l'algorithme en langage naturel suivant.

1.  $n \leftarrow 10$
2. Pour  $i$  allant de 1 à  $n$  faire
3.     Pour  $j$  allant de 1 à  $n$  faire
4.         Afficher la valeur de  $i \times j$
5.     Fin Pour
6. Fin Pour

- a. Traduire cet algorithme en langage Python.
  - b. Que fait ce script ?
3. Déterminer la valeur de la variable  $d$  après exécution du programme dans chacun des cas ci-dessous.

```
1 a=1;b=1;c=1;d=0
2 for k in range(2):
3 a=a+b
4 b=b+a
5 c=c+b
6 d=d+c
```

```
1 a=1;b=1;c=1;d=0
2 for k in range(2):
3 a=a+b
4 b=b+a
5 c=c+b
6 d=d+c
```

```
1 a=1;b=1;c=1;d=0
2 for k in range(2):
3 a=a+b
4 b=b+a
5 c=c+b
6 d=d+c
```

```
1 a=1;b=1;c=1;d=0
2 for k in range(2):
3 a=a+b
4 b=b+a
5 c=c+b
6 d=d+c
```

## 2 La boucle non bornée while

### Exercice 2

Il est parfois utile dans un programme de répéter une ou plusieurs instructions un nombre inconnu de fois. Lorsque le nombre de répétitions n'est pas connu à l'avance, on utilise une boucle non bornée qui est parcourue jusqu'à ce qu'une certaine condition ne soit plus vérifiée. Tant que cette condition est vérifiée, la boucle continue.

| Langage naturel                                   | Langage Python                             |
|---------------------------------------------------|--------------------------------------------|
| <b>Tant que condition faire</b><br>instruction(s) | <b>while condition :</b><br>instruction(s) |

1. Dans l'algorithme
- |                                                                                                                                                                                                                        |                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <ol style="list-style-type: none"> <li>1. <math>n \leftarrow \dots</math></li> <li>2. <b>Tant que</b> <math>m &gt; 9</math></li> <li>3.     <math>m \leftarrow m - 9</math></li> <li>4. <b>Fin Tant que</b></li> </ol> | la variable $m$ contient un nombre entier. |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|

- a. Si, en début d'algorithme, la variable  $m$  reçoit 5, quelle valeur contient-elle en fin d'algorithme ?
- b. Si, en début d'algorithme, la variable  $m$  reçoit 15, quelle valeur contient-elle en fin d'algorithme ?
- c. Dans cette question, la valeur que reçoit  $m$  en début d'algorithme est 30.  
Compléter (autant que nécessaire) le tableau suivant et en déduire la valeur de la variable  $m$  après l'exécution de cet algorithme.

| Etapes                      | $m$ | Condition vérifiée ? |
|-----------------------------|-----|----------------------|
| Avant la boucle             | 30  | Oui                  |
| 1er passage dans la boucle  |     |                      |
| 2ème passage dans la boucle |     |                      |
|                             |     |                      |
|                             |     |                      |

- d. La variable  $m$  peut-elle contenir 21 en fin d'algorithme ?
- e. Quelles valeurs peut recevoir la variable  $m$  en début d'algorithme pour que la boucle « Tant que » soit parcourue ?

2. On donne le script
- ```

1 n=0
2 somme=0
3 while somme<=10:
4     n=n+1
5     somme=somme+n
6 print(n)
    
```

Quelle est la valeur affichée ? Que se passe-t-il si on inverse les deux instructions de la boucle.

3. On donne le script
- ```

1 from random import randint
2 n=0
3 s=0
4 while s!=12:
5 de1=randint(1,6)
6 de2=randint(1,6)
7 s=de1+de2
8 n=n+1
9 print(n)

```

Que modélise-t-il ?

### Exercice 3

Un DAB propose de ne distribuer que des billets de 10 ou 20 €. Lors d'un retrait, le DAB est programmé pour distribuer le moins de billets possible.

- a. On a commencé à écrire un script qui calcule le nombre de billets de chaque sorte distribués par un DAB pour une somme  $n$  d'argent.  
Compléter ce script.

```
1 n=330
2 if n%10!=0:
3 print('Impossible')
4 else:
5 i=0
6 while n>=20:
7 n,i=.....,i+1
8 billets_20=i
9 billets_10=.....
```

- b. A quoi sert l'instruction de la ligne 2 ?  
c. Quelles sont les valeurs des variables  $\text{billets}_{20}$  et  $\text{billets}_{10}$  après exécution du script ?