

Corrigé – La boucle bornée

Exercice 1

1. (a) 5
(b) `for _ in range(10)`
2. (a) Un `i` remplace le `_` et est ensuite affiché. C'est une variable de boucle.
(b) 0, 1, 2, 3 et 4 (chacun sur une ligne)
3. (a) La fonction `range` prend deux paramètres 1 et 5.
(b) De 1 à 4
4. (a) La fonction `range` prend trois paramètres 5, 1 et -1.
(b) Le script affiche 5, 4, 3 puis 2 (chacun sur une ligne). On peut alors imaginer que le -1 permet d'aller « en arrière » avec un pas de 1.
5.

```
for i in range(1, 11):  
    print(i)
```
6.

```
for i in range(1, 10, 2):  
    print(i)
```


(a) 1, 3, 5, 7 puis 9 (chacun sur une ligne)
(b)

```
for i in range(3, 10, 3):  
    print(i)
```
7.

```
for i in range(41):  
    print(i)
```
8.

```
for i in range(40, -1, -1):  
    print(i)
```
9.

```
for i in range(40, 0, -5):  
    print(i)
```

Exercice 2

1. Il désigne la valeur avant laquelle on s'arrête : la dernière valeur prise par la variable est `stop-1`. La première est 0.
2. Ils désignent la valeur de départ (inclusive), et la valeur avant laquelle on s'arrête.
3. Ils désignent la valeur de départ, la valeur avant laquelle on s'arrête, et le pas.
4. Par défaut, la valeur de début est 0 et la valeur du pas est 1.

Exercice 3

Pseudo-code (langage naturel)	Python
Pour i dans séquence instructions	<code>for i in sequence:</code> instructions
Pour j variant de 0 à fin-1 instructions	<code>for j in range(fin):</code> instructions
Pour k variant de debut à fin-1 instructions	<code>for k in range(debut, fin):</code> instructions
Pour l variant de debut à fin-1 en variant de pas instructions	<code>for l in range(debut, fin, pas):</code> instructions

Exercice 4

1. 10 fois

2.

Itération	i	2*i
1	1	2
2	2	4
3	3	6
4	4	8
5	5	10
6	6	12
7	7	14
8	8	16
9	9	18
10	10	20

3. 2
4
6
8
10
12
14
16
18
20

4.

0
3
6
9
12
15
18
21
24
27

0
1
4
9
16

0
1
2
3
4
5
6
7
8
9

Exercice 5

```
1. | somme = 0
   | for c in range(5):
   |     somme = somme + c
```

somme vaut 10 dans les deux cas.

```
2. |         total = 0
   |         i = 0
   |         while i < 5:
   |             total = total + i**2
   |             i = i + 1
```

total vaut 30 dans les deux cas.