

Corrigé – La boucle bornée

Exercice 1

- (a) 5
 - (b) `for _ in range(10)`
- (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)
- (a) La fonction `range` prend deux paramètres 1 et 5.
 - (b) De 1 à 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.
- ```
for i in range(1, 11):
 print(i)
```
- ```
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)
```
- ```
for i in range(41):
 print(i)
```
- ```
for i in range(40, -1, -1):  
    print(i)
```
- ```
for i in range(40, 4, -5):
 print(i)
```

## Exercice 2

- Il désigne la valeur avant laquelle on s'arrête : la dernière valeur prise par la variable est `stop-1`.
- Ils désignent la valeur de départ (incluse), et la valeur avant laquelle on s'arrête.
- Ils désignent la valeur de départ, la valeur avant laquelle on s'arrête, et le pas.
- 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<br>instructions                                 | <code>for i in sequence:</code><br>instructions               |
| Pour j variant de 0 à fin-1<br>instructions                          | <code>for j in range(fin):</code><br>instructions             |
| Pour k variant de debut à fin-1<br>instructions                      | <code>for k in range(debut, fin):</code><br>instructions      |
| Pour l variant de debut à fin-1 en<br>variant de pas<br>instructions | <code>for l in range(debut, fin, pas):</code><br>instructions |

Exercice 4

1. 10 fois

2.

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

3. 1  
4  
6  
8  
10  
12  
14  
16  
18  
20

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.

## Exercice 6

1. 16

```
2. def truc(n):
 x = 1
 for k in range(1, n):
 x = 2 * x
 print(x)
```

## Exercice 7

```
def somme(n):
 s = 0
 for k in range(1, n+1):
 s = s + k
 print(s)
```

## Exercice 8

$10000000000000 (= 10^{12})$  devrait être affiché après  $10^{12}$  tours de boucle mais cette opération demande trop de ressources et, bien que la boucle soit bornée, le programme ne s'arrête pas (du moins sur nos ordinateurs ordinaires).