

# suites arithmétiques

## Exercice 2 (Complément)

Soit  $(u_n)$  la suite arithmétique de raison 3 telle que  $u_1 = -5$ .

$(u_n)$  est la suite arithmétique définie par récurrence par :  $u_{n+1} = u_n + 3$  avec  $u_1 = -5$ , c'est-à-dire  $u_0 = -8$ .

Elle est par ailleurs définie d'une manière explicite par  $u_n = 3n - 8$ .

On considère les codes en Python ci-dessous :

### Mode éditeur

Code 1	Code 2	Code 3	Code 4
<pre>def u(n):     return 3*n - 8</pre>	<pre>def u(n)     U = -5     for i in range(1, n+1):         U = U + 3     return U</pre>	<pre>def u(n)     U = -5     for i in range(n):         U = U + 3     return U</pre>	<pre>def u(n)     U = -5     i = 0     while i &lt; n:         U = U + 3         i = i + 1     return U</pre>

Complétons pour chacun des codes 2 à 4 les tableaux ci-dessous :

Code 2

n	i	U
3		-5
	1	-2
	2	1
	3	4

Code 3

n	i	U
3		-5
	0	-2
	1	1
	2	4

Code 4

n	i	U	i < n ?
3	0	-5	Vrai
	1	-2	Vrai
	2	1	Vrai
	3	4	Faux

Les codes renvoient dans chaque cas la valeur 4 pour  $u_3$ . Or,  $u_3 = 1$ . Ils sont donc erronés.

Code 2

n	i	U
1		-5
	1	-2

Code 3

n	i	U
1		-5
	0	-2

Code 4

n	i	U	i < n ?
1	0	-5	Vrai
	1	-2	

De même, les codes renvoient la valeur -2 pour  $u_1$ . Or,  $u_1 = -5$ .

En modifiant et adaptant les codes ci-dessus, nous pouvons par exemple écrire les deux codes ci-dessous, lesquels renvoient la valeur du terme  $u_n$  pour  $n$  entré en argument de la fonction  $u$ .

Mode éditeur

Code fonctionnel 1
<pre>def u(n)   U = -8   i = 0   if n==0:     return U   else:     while i&lt;n:       U = U + 3       i = i + 1     return U</pre>

Test du code

n	i	U	i < n ?
3	0	-8	Vrai
	1	-5	Vrai
	2	-2	Vrai
	3	1	Faux

Pour  $n = 3$ , la valeur renvoyée par la fonction  $u$  est 1, c'est-à-dire  $u_3$ .

Mode éditeur

Code fonctionnel 4
<pre>def u(n):   U = -8   i = 0   if n==0:     return n   else:     for i in range(1,n+1):       U+=3     return U</pre>

Test du code

n	i	U
3	0	-8
	1	-5
	2	-2
	3	1

Pour  $n = 3$ , la valeur renvoyée par la fonction  $u$  est aussi égale à 1, c'est-à-dire  $u_3$ .

Conclusion

Ces deux codes fonctionnent parfaitement et permettent avec une boucle conditionnelle (while) et une boucle inconditionnelle (for i in range) d'obtenir la valeur du terme  $u_n$ .

Application en mode console

```
>>> U = [u(i) for i in range(11)] # Création d'une liste en compréhension à partir de u
>>>U
[-8, -5, -2, 1, 4, 7, 10, 13, 16, 19] ← liste des dix premiers termes de la suite u
```