

# ECT1



## TP 06: La boucle for acte 1

---

### **Exercice 1 : Révision de range**

1- Quelles valeurs sont présent par k dans l'instruction `k in range(4)`

2- Quelles valeurs sont présent par k dans l'instruction `k in range(10,25,3)`

3- Quelles instruction avec range doit-on écrire pour que i prenne les valeurs suivantes 3, 4, 5, 6, 7, 8, 9

4- Quelles instruction avec range doit-on écrire pour que i prenne les valeurs suivantes 1, 3, 5, 7

### **Exercice 2 : Révision de for**

1- Sans taper le programme dire quel affichages obtient-on par exécution de chacun des programme suivants

```
# Programme 1
a=1
for k in range(1,4) :
    a=k*a
print(a)
```

```
# Programme 2
a=1
for k in range(1,4) :
    a=k*a
    print(a)
```

2- Compléter le script suivant afin qu'il affiche les cubes des entiers non nul de 1 à 12

```
# cube des entiers
for i=
    cube=
.....
```

**Exercice 3 : Comment calculer les termes d'une suite**

Dans les exercices sur les suites, nous avons étudié la suite  $(w_n)$  définie par  $w_0=4$  et la formule de récurrence  $w_{n+1}=2w_n-3$   
Calculer  $w_1$  puis  $w_2$

Si nous voulons calculer  $w_{12}$ , il faut répéter ces opérations 12 fois, c'est-à-dire : prendre le résultat précédent, le multiplier par . . . . ., puis . . . . .

Définissons une variable  $w$  par  $w=4$  c'est le premier terme  
A la première étape, on calcule  $2*w-3$ . Si l'on appelle de nouveau  $w$  le résultat, la deuxième étape revient encore à calculer  $2*w-3$ . Ainsi, on répète 12 fois l'opération  $2*w-3$  en nommant  $w$  le résultat à chaque fois.  
En Python cela donnerait (on ne demande pas de taper ce programme) :

```
u=4
for k in range(1,13) : # pour k allant de 1 à 12
    u=2*u-3 #calculer 2*u-3 et donner cette nouvelle valeur à u
print('u12 =', u)
```

**Exercice 4 : Calculer des termes de suites à la "main"**

1- Lire le programme suivant (NE PAS le taper dans Python), et compléter le tableau qui donne les valeurs successives de  $k$  et  $u$  au moment où le programme les affiche.

```
u=32
for k in range (1,5):
    u=u/2
    print('k=', k, ' u=', u)
```

valeur de $k$	1	2	....			
valeur de $u$						

- 2- Lire le programme suivant (NE PAS le taper dans Python), et compléter le tableau qui donne les valeurs successives de  $k$  et  $u$  au moment où le programme les affiche.

```
S=0
for k in range (1,6):
    S=S+2*k
print(S)
```

valeur de $k$						
valeur de $2k$						
valeur de $S$						

**Exercice 4 : Premières suites en Python**

**1- Cas d'une suite définie par son terme général**

Soit  $(u_n)$  définie par  $\forall n \in \mathbb{N}^*, u_n = \ln(n)$

Écrire un programme qui calcule et affiche le terme de rang  $n$

**a- Brouillon**

Étape 1: Est-ce une suite définie par son terme général ou par récurrence ?



**b- Python**

**2- Cas d'une suite définie par une relation de récurrence d'ordre 1**

Soit  $(u_n)$  définie par  $u_1=3$  et  $\forall n \in \mathbb{N}, u_{n+1}=7u_n-1$

Écrire un programme qui calcule et affiche le terme de rang  $n$

### a- Brouillon

Étape 1: Est-ce une suite définie par son terme général ou par récurrence ?

Étape 2: Écrire  $u_n$  en fonction de  $n$  puis remplacer  $n$  par la variable du for

### b- Python

La structure du programme est la suivante



#### Programme type pour afficher le terme de rang $n$ d'une suite récurrente d'ordre 1

```
n=int(input('n='))
u=          # valeur du 1er terme
for k in range(debut, n+1) :
    u=          #utiliser le brouillon et remplacer  $u_k$  et  $u_{k-1}$  par u
print(..) # avec indentation si on veut afficher tous les termes
sans sinon
```

### Exercice 5 : Suites en Python

Dans chacun des cas suivants, écrire un programme demandant  $n$  à l'utilisateur et affichant  $I_n$

a)  $I_1 = e$  et  $\forall n \in \mathbb{N}, I_{n+1} = nI_n + \ln(3)$

b)  $\forall n \in \mathbb{N}, I_n = 7^n + e^5$

**Exercice 5 : Suites et sommes en Python**

Dans chacun des cas suivants, écrire un programme demandant  $n$  à l'utilisateur et affichant  $u_n$  et

$S_n = \sum_{k=0 \text{ ou } 1}^n u_k$  (faire un affichage du type  $u_n = \dots$  et  $S_n = \dots$ ).

a)  $u_0 = 1$  et  $\forall n \in \mathbb{N}, u_{n+1} = \frac{1}{2}u_n + e$

La structure du programme avec des somme est la suivante



**Programme type pour afficher la somme d'une suite récurrente d'ordre 1**

```
n=int(input('n='))
u=          # valeur du 1er terme
S=u         # initialisation de la somme
for k in range(debut, n+1) :
    u=      #utiliser le brouillon et remplacer  $u_k$  et  $u_{k-1}$  par u
    S=S+u
print(S)
```

b)  $u_1 = 1$  et  $\forall n \in \mathbb{N}^*, u_{n+1} = 2nu_n + 3$