

TP4

BOUCLES FOR

Préparer les exercices 1 et 2

On souhaite traduire un algorithme répétitif de ce type :

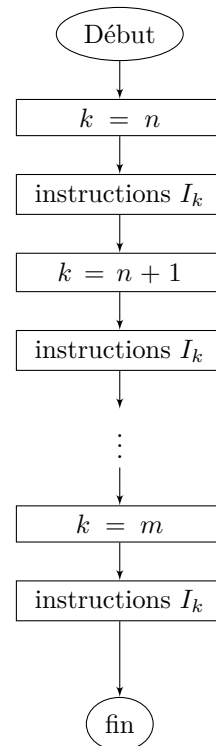
Pour k variant de n à m ,
 Réaliser les instructions I_k
Fin

La syntaxe python pour écrire ce type de répétitions est la suivante :

```
for k in range(n,m+1) :
    instructions Ik
```

et plus généralement

```
for element in sequence :
    bloc_instructions
```



- La variable de boucle (ici k) peut avoir un nom différemment. On veillera à rester cohérent.
- $\text{range}(n,m+1)$ est la liste des entiers $[n, n + 1, \dots, m]$.
- $\text{range}(n)$ est la liste des entiers $[0, 1, \dots, n - 1]$.

Exemple :

Afficher $1^2, 2^2, 3^2, \dots, 100^2$.

```
for i in range(1,101) :
    print(i**2)
```

Exercice 1 Pour tout entier naturel $k \leq 20$, calculer $y = 3k + 2$ puis afficher la valeur de y .

Exercice 2 Exercice classique : calculer le terme de rang n d'une suite récurrente

Soit $(u_n)_{n \in \mathbb{N}}$ la suite définie par $\begin{cases} u_0 = 0 \\ \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n + 4n} \end{cases}$

1. Compléter le tableau suivant donnant les calculs à faire pour obtenir u_n avec $n = 3$.

Étape	À faire	Instruction python
Initialisation	$u_0 = 0$	$u = 0$
$k = 0$	$u_1 =$	
$k = 1$		
$k = 2$		
Après la boucle		

On rappelle que la fonction racine carrée s'obtient avec `sqrt` de la bibliothèque `numpy`.

2. Écrire un programme python calculant et affichant u_n avec une boucle `for`. Ce programme et le tableau précédent doivent être cohérents.

```
import numpy as np

n = 10 # tester avec d'autres valeurs de n

u = .....

for ... in range(.....) :

.....

.....
```

Tester en calculant u_0, u_1, u_2 .

3. Que faut-il changer pour que le programme affiche tous les termes de la suite, de u_0 à u_n ?

Exercice 3 Soit $(v_n)_{n \in \mathbb{N}}$ la suite définie par

$$\begin{cases} v_0 = 5 \\ \forall n \in \mathbb{N}, v_{n+1} = v_n + \frac{2}{v_n} \end{cases}$$

Écrire une **fonction** python **Suite_v(n)** qui, étant donné un entier naturel n , renvoie la valeur de v_n .

Exercice 4 Les compteurs

1. Comment simuler le lancer d'une pièce qui tombe sur pile avec une probabilité de $\frac{1}{3}$ en utilisant `rd.randint(1,4)` ?

Le programme doit renvoyer 'P' (chaîne de caractères) avec probabilité $\frac{1}{3}$ et 'F' avec probabilité $\frac{2}{3}$.

```
import numpy.random as rd

def piece(): # cette fonction n'a pas de paramètre d'entrée
    r = rd.randint(1,4)

    if ..... :
        return 'P'
    else :
        return 'F'
```

2. Tester plusieurs fois en appelant `piece()` dans la console, jusqu'à obtenir au moins une fois chaque côté. Combien de lancers ont été nécessaires ?
3. Simuler 900 lancers de cette pièce et déterminer le nombre de 'P' obtenus. Ce nombre doit être proche de 300.

```
compteur = ...

for ... in ..... :

    if ..... == 'P' :

        .....
```

Exercice 5 Suite récurrente d'ordre 2

Soit $(w_n)_{n \in \mathbb{N}}$ la suite définie par

$$\begin{cases} w_0 = 1, w_1 = 1 \\ \forall n \in \mathbb{N}, w_{n+2} = w_n + w_{n+1} \end{cases}$$

Écrire une **fonction** python **Suite_w(n)** qui, étant donné un entier naturel n , renvoie la valeur de w_n .