TP ALGO 4 : LA BOUCLE BORNEE FOR

1. NOTION DE BOUCLE

Principe

La boucle for, ou boucle bornée, permet de répéter une instruction, ou un bloc d'instructions, un nombre connu de fois.

La syntaxe Python pour définir une boucle bornée est la suivante :

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

On répète les instructions (indentées) pour k allant de m à n-1.

L'instruction range peur prendre de 1 à 3 arguments :

for *k* in ... :	*k* prend successivement les valeurs entières
range(5)	de 0 à 4 donc 0, 1, 2, 3, 4 (attention : 5 est exclu)
range(2,8)	de 2 à 7 donc 2, 3, 4, 5, 6, 7
range(10,17,2)	de 10 à 16 mais de 2 en 2, donc 10, 12, 14, 16

Exemples à tester

- *Tester le programme ci-contre. Quelles sont les valeurs successives prises par k ? Qu'obtient-on à l'affichage ?*

```
for k in range(2,6):
    a = 3 * k
    print (a)
```

- *Même questions avec le programme ci-contre.*

```
for k in range(2,6):
    a = 3 * k
print (a)
```

- *Quel est le rôle de la fonction définie ci-contre ?*
  On pourra s'aider d'un tableau donnant les valeurs successives de k et de S dans le cas d'un appel de la fonction somme avec pour argument 6.
  Ecrire ce programme dans l'interpréteur et vérifier votre résultat. On pourra tester plusieurs exemples.

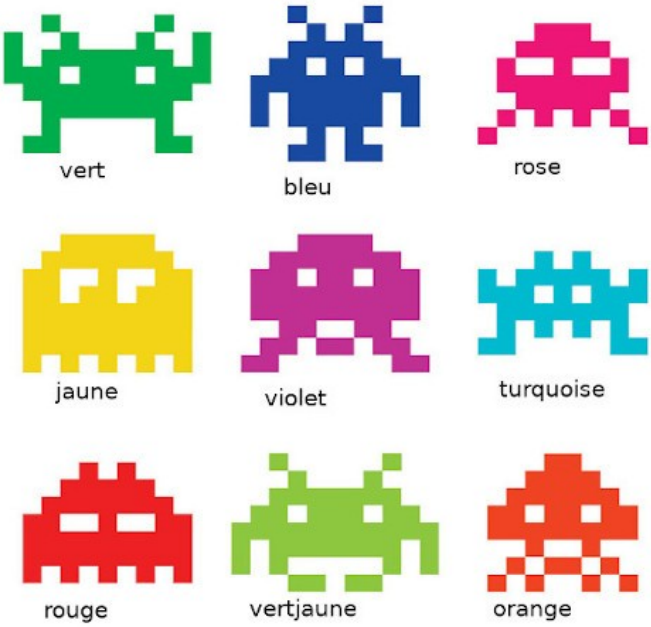
```
def somme(n):
    S=0
    for k in range(1, n+1):
        S = S + k
    return S
```

- *On peut également utiliser une boucle bornée en itérant par exemple sur les lettres d'un mot :*

```
mot = "bonjour"
for lettre in mot:
    print (lettre)
```

Application : Sprite *Space Invaders*

- *Choisir un sprite space invader et une couleur parmi ces modèles.*



- *A l'aide de turtle, créer une fonction dessiner_pixel(colonne, ligne) qui dessine un pixel carré coloré. On se basera sur une « grille » imaginaire appliquée au dessin.*
- *Commencer à dessiner les 3 premières lignes du sprite en commençant par la première ligne, puis la seconde, puis la troisième…*

Les instructions pour dessiner la deuxième ligne sont donc :

dessine_pixel(2,2)
dessine_pixel(3,2)
dessine_pixel(4,2)
dessine_pixel(5,2)
dessine_pixel(6,2)

Par exemple :

On remarque que seul le premier chiffre (qui représente la colonne) change d'une instruction à l'autre. Il varie en prenant successivement les valeurs 2, 3 , 4, 5 et 6. On va le remplacer par une variable k, on a alors l'instruction : *dessine_pixel(k,2)*

Il faut maintenant indiquer comment varie la variable k, elle varie en prenant successivement les valeurs 2, 3, 4, 5 et 6.

- Utiliser une boucle for pour écrire plus simplement les instructions **de la 2ème ligne**.
- Un s'inspirant des résultats précédents, dessiner **toutes les lignes** de votre space invader en utilisant des instructions for.

  (BONUS : Pour que le dessin soit plus rapide on pourra créer une des fonctions dessine_pixel_rapide).

Exercices d'application

Exercice 1 :

1. Quelles sont les valeurs prises par k dans chacun des cas suivants :

   a. for k in range(5)

   b. for k in range(10,25,3)

2. Compléter l'instruction pour que i prenne les valeurs suivantes :

a. 3, 4, 5, 6, 7 for k in range( )

b. 1, 3, 5, 7 for k in range( )

3. Quelles valeurs seront affichées après exécution des scripts suivants :

a.
```
for i in range(4) :
    print(2 * i)
```

b.
```
a = 1
for k in range(1, 4):
    a = k*a
print(a)
```

c.
```
a = 1
for k in range(1, 4):
    a = k*a
    print(a)
```

Exercice 2 :

1. Ecrire un script Python qui affiche les entiers pairs de 2 à 100.

2. Ecrire un script Python qui affiche la table de multiplication de 6.

3. Ecrire une fonction Python qui prend un entier n en paramètre et qui affiche la table de multiplication de n.

Exercice 3 : On veut écrire un script qui compte le nombre de « a » dans une phrase.

1. Copier le script ci-contre, en le complétant, dans l'éditeur Python :

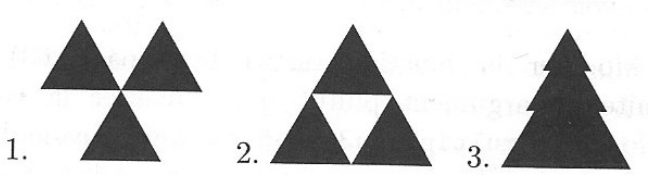
```
phrase="J'aime les maths"
nb_de_a=0
for lettre in phrase:
    if lettre ==     :
        nb_de_a =
print("Le nombre de 'a' est : ", nb_de_a)
```

2. Exécuter le script. Qu'affiche-t-il ?

3. Exécuter le script avec la phrase « Je vais à la plage ». Expliquer le résultat affiché.

Exercice 4 : Ecrire un programme qui demande à l'utilisateur un entier n puis calcule et affiche $1+2+3+\ldots+n$, ainsi que l'entier $\frac{n(n+1)}{2}$. Qu'observe-t-on ?

Exercice 5 : En utilisant la fonction permettant de dessiner un triangle, reproduire les figures suivantes :



Exercice 6 : Ecrire un programme demandant à l'utilisateur un entier n et traçant un damier de côté n.