2.3 L'instruction *for* en Python

2.3.1 La boucle itérative for *(Pour)*

En Python, la boucle « for » ne fonctionne pas comme dans d'autres langages C ou Pascal.
Plutôt que de toujours itérer en suivant une progression arithmétique de nombres, c'est plus une **boucle de parcours** de listes ou de caractères qu'une boucle itérative.
Nous allons la travailler dans quelques exemples

Remarque 8 :

La syntaxe de la boucle `for` est donnée par :

```
for i in E:
    instructions
```

Exemple 7 :

```
words = ['cat', 'window', 'defenestrate']
for w in words:
        print(w)

cat
window
defenestrate
```

Exercice 5 :

Que fait ce petit programme ?

```
#-*-coding:Latin-1 -*

#Algorithme Mystère

sequence = "Bonjour les étudiants"

for lettre in sequence:
   if lettre in "AEIOUYaeéiyou":
      print (lettre)
   else:
      print ("*")
```

2.3.2 La fonction *range()*

La fonction *range*() permet à l'utilisateur de générer une série de nombres dans une plage donnée.
En fonction du nombre d'arguments que l'utilisateur transmet à la fonction, l'utilisateur peut décider où cette série de nombres commencera et se terminera ainsi que l'ampleur de la différence entre un nombre et le suivant.
range() prend principalement trois arguments :

1. **start** : c'est l'entier à partir duquel la séquence d'entiers doit être retournée ; c'est l'entier de départ
2. **stop** : entier avant lequel la séquence d'entiers doit être renvoyée. La plage d'entiers se termine à $stop - 1$.
3. **step** : c'est une valeur entière qui détermine l'incrément ou le pas entre chaque entier de la séquence.

Remarque 9 :

1. La fonction `range()` est utilisée pour générer une séquence de nombres.

2. `range()` est couramment utilisée pour la boucle, par conséquent, sa connaissance est un aspect clé lors du traitement de tout type de code `Python`. L'utilisation la plus courante de la fonction `range()` en Python est d'itérer le type de séquence *(Liste, chaîne, etc.)* avec les boucles `for` et `while`
3. **La syntaxe** est donnée par `range(start,stop,step)`, sachant que :
   (a) Il est possible de n'appeler qu'un seul argument `range(stop)` et nous avons comme entier de départ $n = 0$ (nous nous arrêtons à $stop - 1$; *(cf exemple 1 ci-dessous)*
   (b) Si nous avons 2 arguments `range(start,stop)` la liste part de l'entier start pour s'arrêter à $stop - 1$ *(exemple 2)*
   (c) Et pour 3 arguments `range(start,stop, step)` la liste part de l'entier start pour s'arrêter à $stop - 1$ en faisant des pas de longueur `step`*(exemple 3)*

Exemple 8 :

Des exemples

1.
```
for i in range(10):
        print(i)
```
   Et nous avons comme affichage :
   $$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9$$

2.
```
for i in range(4,10):
        print(i)
```
   Et nous avons comme affichage :
   $$4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9$$

3.
```
for i in range(2,10,3):
        print(i)
```
   Et nous avons comme affichage :
   $$2 \quad 5 \quad 8$$

4.
```
somme = 0
for i in range(1, 11):
    somme +=  i
print("La somme des 10 premiers naturels est", somme)
```
   Et nous trouvons **55**

5. Nous savons que $\lim\limits_{n\to+\infty}\sum\limits_{k=1}^{+\infty}\dfrac{1}{k^2} = \dfrac{\pi^2}{6}$, et, sans faire d'approximation de la limite, nous allons calculer $\sum\limits_{k=1}^{n}\dfrac{1}{k^2}$ jusque l'entier n choisi par l'opérateur.

```
#-*-coding:Latin-1 -*
#algo Somme_Inverse_Carres
  #Cet algorithme calcule la somme des inverses des carrés jusqu'à un ordre n, arbitrai

n=int(input("Jusqu'à quel entier souhaitez vous aller? "))

Somme = 0
for i in range(1,n+1): #Nous allons de 1 à (n+1)-1=n
    Somme += 1/(i**2)
print("Au rang ", n, "la somme est ",Somme)
```

Exercice 6 :

Ecrire un programme qui affiche tous les nombres pairs entre 0 et n, dans l'ordre croissant, le nombre n étant donné par l'opérateur.

Exercice 7 :

Ecrire un script qui calcule $p!$ pour un $p \in \mathbb{N}$ donné par l'utilisateur

Exercice 8 :

Ecrire un script qui calcule les n premiers termes de la suite :

$$\begin{cases} u_0 \in \mathbb{R}^* \\ u_{n+1} = \dfrac{1}{2}\left(u_n + \dfrac{a}{u_n}\right) \text{ avec } a \in \mathbb{R}^{*+} \end{cases}$$

On admettra, pour cet exercice que pour tout $n \in \mathbb{N}$, $u_n \neq 0$. *(Cette suite a été étudiée en L_0)*