ALGO 05 – LES BOUCLES "FOR"

I Avant-propos : range

La fonction **range** permet de créer rapidement des listes de valeurs **entières** consécutives ou régulièrement espacées.

	Explication
<code>range(fin)</code>	Renvoie la liste des entiers entre 0 et fin-1
<code>range(deb, fin)</code>	Renvoie la liste des entiers entre deb et fin-1
<code>range(deb, fin, pas)</code>	Renvoie la liste des entiers entre deb et fin-1 avec un pas

1. Définir la liste [3,4,5,6,7] avec range.

Entrée [1]: `range(3, 8)`

2. Définir la liste [0,1,2,3,...,100] avec range.

Entrée [2]: `range(101)` #Ou `range(0, 101)`

3. Définir la liste [4,6,8,10,12,14,16] avec range.

Entrée [3]: `range(4, 17, 2)`

4. Définir la liste [0,3,6,9,12] avec range.

Entrée [4]: `range(0, 13, 3)`

II Présentation des boucles for

Les **boucles for** permettent de **répéter** une liste d'instruction un certain nombre de fois.

Pour tout élément d'une séquence

Réaliser les instructions ...

Fin

`for element in sequence:`

`bloc d'instructions`

II.1 La boucle "for ... in Liste"

Expliquons sur un exemple.

Entrée [5]:

```
L = ["Lundi", "Mardi", "Mercredi"]
for e in L :
    print(e)
```

Valeurs de e successives	Instructions réalisées
"Lundi"	Affiche la chaîne de caractères "Lundi"
"Mardi"	Affiche la chaîne de caractères "Mardi"
"Mercredi"	Affiche la chaîne de caractères "Mercredi"

Out [5]: "Lundi"
"Mardi"
"Mercredi"

Entrée [6]: L = [3, True, "Info"]
for e in L :
 print(e)

Valeurs de e successives	Instructions réalisées
3	Affiche le flottant 3
True	Affiche le booléen True
"Info"	Affiche la chaîne de caractères "Info"

Out [6]: 3
True
"Info"

Entrée [7]: L = [1, 2, 3]
for e in L :
 print("Bonjour")

Valeurs de e successives	Instructions réalisées
1	Affiche la chaîne de caractères "Bonjour"
2	Affiche la chaîne de caractères "Bonjour"
3	Affiche la chaîne de caractères "Bonjour"

Out [7]: "Bonjour"
"Bonjour"
"Bonjour"

Entrée [8]: L = [6, 14, 42]
for x in L :
 x = x+1
 print(x)

Valeurs de x successives	Instructions réalisées
6	Change la valeur de x en 7 Affiche le flottant 6
14	Change la valeur de x en 15 Affiche le flottant 15
42	Change la valeur de x en 43 Affiche le flottant 43

Out [8]: 7
15
43

Entrée [9]:

```
L = [1, 2, 3]
for e in L:
    e = e**2+1
    print(e)
```

Out [9]:

```
2
5
10
```

II.2 La boucle "for ... in Chaîne de caractères"

Entrée [10]:

```
for e in "Lundi" :
    print(e)
```

Valeurs de e successives	Instructions réalisées
L	Affiche L
u	Affiche u
n	Affiche n
d	Affiche d
i	Affiche i

Out [10]:

```
L
u
n
d
i
```

II.3 Boucle "for ... in range(...)"

Dans la partie précédente, la variable parcourait une liste. Lorsque l'on veut utiliser une variable qui prend des valeurs entières, on peut utiliser la commande `range(deb, fin)`, qui désigne l'ensemble des entiers compris entre `deb` (inclus) et `fin` (exclu).

Par exemple,

Entrée [11]:

```
for k in range(1,4) :
    print(k)
```

Valeurs de k successives	Instructions réalisées
1	Affiche l'entier 1
2	Affiche l'entier 2
3	Affiche l'entier 3

Out [11]:

```
1
2
3
```

Entrée [12]:

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

Out [12]:
6
8
10
12

Entrée [13]:
`for k in range(3) :
 print("Il fait chaud !")`

Out [13]:
Il fait chaud !
Il fait chaud !
Il fait chaud !

Entrée [14]:
`L = ["Lundi", "Mardi", "Mercredi"]
for k in range(len(L)) :
 print(L[k])`

Out [14]:
Lundi
Mardi
Mercredi

III Lecture de programme

Pour tous les programmes suivantes, compléter le tableau et l’affichage du programme.

(i)

```
L=[1,3,5,7]
for i in L:
    print(i**2)
```

Valeurs de i successives	Instructions réalisées
1	Affiche l’entier 1
3	Affiche l’entier 3
5	Affiche l’entier 5
7	Affiche l’entier 7

Out [14]:
1
9
25
49

(ii)

```
for k in "Truc":
    print(k)
```

Valeurs de k successives	Instructions réalisées
T	Affiche T
r	Affiche r
u	Affiche u
c	Affiche c

Out [14]:
T
r
u
c

(iii)

```
for n in range(4):
    n=7*n
    print(n)
```

Valeurs de n	Instructions réalisées
0	Actualise la valeur de n en 0 Affiche 0
1	Actualise la valeur de n en 7 Affiche 7
2	Actualise la valeur de n en 14 Affiche 14
3	Actualise la valeur de n en 21 Affiche 21

```
Out [14]: 0
          7
          14
          21
```

(iv)

```
L=[1,3,7]
for e in L:
    print(e)
    print(2*e)
    print("Test")
```

Valeurs de e	Instructions réalisées
1	Affiche 1 Affiche 2 Affiche "Test"
3	Affiche 3 Affiche 6 Affiche "Test"
7	Affiche 7 Affiche 14 Affiche "Test"

```
Out [14]: 1
          2
          Test
          3
          6
          Test
          7
          14
          Test
```

(v)

```
for i in range(4):
    print("Info")
print("Maths")
```

Valeurs de i successives	Instructions réalisées
0	Affiche "Info"
1	Affiche "Info"
2	Affiche "Info"
3	Affiche "Info"
—	Affiche "Maths"

Out [14]:
 Info
 Info
 Info
 Info
 Maths

(vi)

```
compteur=0
for k in range(4):
    compteur=compteur+1
    print(compteur)
print("compteur=", compteur)
```

Valeurs de k successives	Valeurs de compteur
—	0
0	1
1	2
2	3
3	4

Out [14]:
 1
 2
 3
 4
 compteur = 4

(vii)

```
compteur=0
for k in range(1,4):
    compteur=compteur+k
    print(compteur)
print("compteur=", compteur)
```

Valeurs de k successives	Valeurs de compteur
—	0
1	1
2	3
3	6

Out [14]:
 1
 3
 6
 compteur = 6

(viii)

```
compteur=1
for e in [1,3,6,-1]:
    compteur=compteur+e
print(compteur)
```

Valeurs de e successives	Valeurs de compteur
—	1
1	2
3	5
6	11
-1	10

Out [14]: 10

(ix)

```
L=[1,3,6,-1]
compteur=0
for k in range(len(L)):
    compteur=compteur+k
print(compteur)
```

Valeurs de k successives	Valeurs de compteur
—	0
0	0
1	1
2	3
3	6

Out [14]: 6

(x)

```
L=["Info","Maths","Phys"]
for k in range(len(L)):
    print(k)
    print(L[k])
```

Out [14]: 0
Info
1
Maths
2
Physique

(xi)

```
L=[]
for i in range(2,7):
    L.append(3*i)
    print(len(L))
print(L)
```

Out [14]: 1
2
3
4
5
[6, 9, 12, 15, 18]

IV Exercices

Exercice 1 *Les questions de cet exercice sont indépendantes.*

1. À l'aide d'une boucle `for`, afficher tous les entiers de 90 à 100.

Entrée [15]:

```
for k in range(90,101):  
    print(k)
```

Out [15]:
90
91
92
93
94
95
96
97
98
99

2. À l'aide d'une boucle `for`, afficher le carré de tous les entiers de 1 à 10.

Entrée [16]:

```
for k in range(1,11):  
    print(k**2)
```

Out [16]:
1
4
9
16
25
36
49
64
81
100

3. Ecrire une fonction, appelée `tabledemultiplication`, qui prend en argument un entier n et qui renvoie une liste contenant le résultat de toutes les multiplications de la forme $n \times k$ pour $k \in \{0, \dots, 10\}$. On vérifiera que `tabledemultiplication(5)` renvoie bien la table de 5.

Entrée [17]:

```
def tabledemultiplication(n):  
    L=[]  
    for k in range(11):  
        L.append(n*k)  
    return(L)
```

Entrée [18]: `tabledemultiplication(5)`

Out [18]: `[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]`

Exercice 2 On place la somme de 1000 euros sur un compte d'épargne. Chaque année, les intérêts sur l'argent placé rapportent 10% (le capital est donc multiplié par 1.10 chaque année).

1. Écrire un code à l'aide d'**une seule variable** qui permet de calculer et d'afficher le capital pour les deux premières années.

```
Entrée [19]: S = 1000
S = S*1.10 #Capital la première année
print(S)
S = S*1.10 #Capital la deuxième année
print(S)
```

```
Out [19]: 1100.0
1210.0
```

2. Écrire un code à l'aide d'**une boucle for** qui permet de calculer et d'afficher le capital pour les deux premières années. Vérifier que les résultats concordent avec ceux de la question précédente.

```
Entrée [20]: S = 1000
for k in range(1,3):
    S = S*1.10
    print(S)
```

```
Out [20]: 1100.0
1210.0
```

3. Écrire un code à l'aide d'**une boucle for** qui permet de calculer et d'afficher le capital de la dixième année uniquement.

```
Entrée [21]: S = 1000
for k in range(1,11):
    S = S*1.10
    print(S)
```

```
Out [21]: 2593.742460100001
```

4. Écrire une fonction en Python, appelée `capital`, qui prend en entrée un entier `n` et qui renvoie le capital de la `n`-ième année. *On vérifiera que la valuation de la fonction en 2 donne 1210.0.*

```
Entrée [22]: def capital(n):
S = 1000
for k in range(1,n+1):
    S = S*1.10
return(S)
```

```
Entrée [23]: capital(2)
```

```
Out [23]: 1210.0
```

5. Écrire une fonction en Python, appelée `listecapital`, qui prend en entrée un entier `n` et qui renvoie une liste contenant le capital de la première année jusqu'à `n`-ième année. *On vérifiera que la valuation de la fonction en 2 donne [1100.0,1210.0].*

```
Entrée [24]: def listecapital(n):
              L=[]
              S = 1000
              for k in range(1,n+1):
                  S = S*1.10
                  L.append(S)
              return(L)
```

```
Entrée [25]: listecapital(2)
```

```
Out [25]: [1100.0,1210.0]
```

Exercice 3 *Les questions de cet exercice sont indépendantes.*

1. Ecrire une fonction, appelée `factorielle`, qui prend en argument un entier n et qui renvoie la valeur de $n!$. *On vérifiera que `factorielle(5)` renvoie 120.*

```
Entrée [26]: def factorielle(n):
              P=1
              for k in range(1,n+1):
                  P=P*k
              return(P)
```

```
Entrée [27]: factorielle(5)
```

```
Out [27]: 120
```

2. Ecrire une fonction, appelée `sommeentiers`, qui prend en argument un entier n et qui renvoie la valeur de la somme $1 + 2 + \dots + n$. *On vérifiera que `sommeentiers(6)` renvoie 21.*

```
Entrée [28]: def sommeentiers(n):
              S=0
              for k in range(1,n+1):
                  S = S + k
              return(S)
```

```
Entrée [29]: sommeentiers(6)
```

```
Out [29]: 21
```

3. Ecrire une fonction, appelée `compteurlettre`, qui prend en argument une lettre et un mot à l'utilisateur et qui renvoie le nombre d'occurrences de cette lettre dans le mot choisi. *On vérifiera que `compteurlettre(a, abracadabra)` renvoie 5.*

```
Entrée [30]: def compteurlettre(lettre, mot):
              compteur=0
              for e in mot:
                  if e == lettre:
                      compteur = compteur + 1
              return(compteur)
```

```
Entrée [31]: compteurlettre("a", "abracadabra")
```

```
Out [31]: 5
```