

Cours3_Listes

October 5, 2019

1 Les listes

2 Définitions et opérations sur les listes

Pour une présentation des listes et de leurs méthodes : voir aussi [Python_seconde](#)

2.0.1 Exercice 1

```
In [2]: from random import randint
        L = [randint(0, 20) for _ in range(35)]  #liste de 35 entiers aléatoires entre 0 et 20
        x = 10                                #le seuil
        print("Liste : ", L)
        print("Seuil : ", x)
        n = 0
        for e in L:
            if e > x:
                n = n + 1
        print("Nombre d'éléments de la liste L supérieurs au seuil : ", n)
```

Liste : [18, 10, 2, 20, 3, 15, 20, 12, 0, 8, 9, 11, 12, 4, 13, 8, 16, 17, 7, 1, 15, 4, 4, 19, 5

Seuil : 10

Nombre d'éléments de la liste L supérieurs au seuil : 18

2.1 Exercice 2

```
In [7]: from random import randint
        L = [randint(0, 20) for _ in range(35)]  #liste de 35 entiers aléatoires entre 0 et 20
        x = 10                                #le seuil
        print("Liste : ", L)
        print("Cible : ", x)
        occurrence = []
        for k in range(len(L)):
            if L[k] == x:
                occurrence.append(k)
        print("Liste des indices des éléments de la liste L égaux à la cible : ", occurrence)
```

Liste : [4, 9, 4, 11, 10, 20, 15, 11, 8, 14, 1, 4, 4, 20, 11, 7, 9, 19, 8, 11, 1, 5, 7, 9, 0, 1]
Cible : 10
Liste des indices des éléments de la liste L égaux à la cible : [4]

3 Listes par compréhension

3.0.1 Exercice 3

Quelle liste produit ce code ?

```
L = []  
for k in range(10):  
    L.append(k ** 2)
```

```
In [13]: #Corrigé de l'exercice 3  
        L = []  
        for k in range(10):  
            L.append(k ** 2)  
        print(L)
```

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

```
In [15]: #La même liste mais décrite par compréhension  
        L2 = [ k ** 2 for k in range(10)]  
        print(L2 == L)
```

True

3.1 Exercice 4 Liste de tous les carrés des entiers pairs entre 0 et 100

```
In [1]: [x ** 2 for x in range(0, 101, 2)]
```

```
Out[1]: [0,  
         4,  
         16,  
         36,  
         64,  
         100,  
         144,  
         196,  
         256,  
         324,  
         400,  
         484,  
         576,  
         676,
```

```
784,  
900,  
1024,  
1156,  
1296,  
1444,  
1600,  
1764,  
1936,  
2116,  
2304,  
2500,  
2704,  
2916,  
3136,  
3364,  
3600,  
3844,  
4096,  
4356,  
4624,  
4900,  
5184,  
5476,  
5776,  
6084,  
6400,  
6724,  
7056,  
7396,  
7744,  
8100,  
8464,  
8836,  
9216,  
9604,  
10000]
```

3.2 Exercice 4 Images positives des entiers entre 0 et 100 par la fonction cosinus

```
In [2]: from math import cos
```

```
In [3]: [cos(n) for n in range(0, 101) if cos(n) >= 0]
```

```
Out[3]: [1.0,  
0.5403023058681398,  
0.28366218546322625,  
0.960170286650366,
```

0.7539022543433046,
0.004425697988050785,
0.8438539587324921,
0.9074467814501962,
0.1367372182078336,
0.6603167082440802,
0.9887046181866692,
0.40808206181339196,
0.424179007336997,
0.9912028118634736,
0.6469193223286404,
0.15425144988758405,
0.9147423578045313,
0.8342233605065102,
0.7654140519453434,
0.9550736440472949,
0.26664293235993725,
0.5551133015206257,
0.9998433086476912,
0.5253219888177297,
0.3005925437436371,
0.9649660284921133,
0.7421541968137826,
0.022126756261955736,
0.8532201077225842,
0.8998668269691937,
0.11918013544881928,
0.6735071623235862,
0.9858965815825497,
0.39185723042955,
0.4401430224960407,
0.9933903797222716,
0.6333192030862999,
0.17171734183077755,
0.9217512697247493,
0.8243313311075577,
0.7766859820216312,
0.9496776978825432,
0.24954011797333814,
0.569750334265312,
0.9993732836951247,
0.5101770449416689,
0.31742870151970165,
0.9694593666699876,
0.7301735609948197,
0.0398208803931389,
0.8623188722876839]

3.3 Exercice 4 Tables de multiplications

In [20]: *# Peuplement par boucles imbriquées*

```
tables = []
for i in range(1, 11):
    nouvelle = []
    for j in range(1, 11):
        nouvelle.append(i * j)
    tables.append(nouvelle)
print(tables)
```

[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10], [2, 4, 6, 8, 10, 12, 14, 16, 18, 20], [3, 6, 9, 12, 15, 18, 21, 24, 27, 30], [4, 8, 12, 16, 20, 24, 28, 32, 36, 40], [5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [6, 12, 18, 24, 30, 36, 42, 48, 54, 60], [7, 14, 21, 28, 35, 42, 49, 56, 63, 70], [8, 16, 24, 32, 40, 48, 56, 64, 72, 80], [9, 18, 27, 36, 45, 54, 63, 72, 81, 90], [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]]

In [19]: *# Peuplement par listes en compréhension*

```
tables2 = [ [i * j for j in range(1, 11)] for i in range(1, 11) ]
print(tables2 == tables)
```

True

3.4 Exercice 5 Permuter les éléments d'une liste

Écrire un code pour permuter les éléments d'indices a et b de la liste L.

L = [2 , 3 , 5 , 7 , 11 , 13 , 17]

a = 2

b = 5

In [4]: *#Corrigé*

```
L = [2 , 3 , 5 , 7 , 11 , 13 , 17 ]
a = 2
b = 5
L[a], L[b] = L[b], L[a]
```

3.5 Exercice 6

Compléter l'instruction évaluée dans la console ci-dessous pour que la liste L3 contienne alternativement les éléments des listes L1 et L2 de même taille.

In [4]: L1 = [2, 3, 5, 7]

In [5]: L2 = [17, 24, 35, 81]

In [6]: L3 = [L1[i // 2] if i % 2 == 0 else L2[i//2] for i in range(2 * len(L1))]

In [7]: L3

Out[7]: [2, 17, 3, 24, 5, 35, 7, 81]

3.6 Exercice 7

```
In [8]: #Pour l'utilisation de Python Tutor dans le notebook
        from metakernel import register_ipython_magics #nécessite d'installer metakernel avec pip
        register_ipython_magics()
```

```
In [10]: %%tutor
```

```
M = [ [0, 0, 0] for i in range(3) ]
N = M
P = [e for e in M ]
Q = [ e[:] for e in M ]
M[2][1] = 3
```

```
<IPython.lib.display.IFrame at 0x7fbcd0be6ba8>
```