

Exercices sur les listes (première NSI)

Les bases

Exercice 1 : application du cours

On exécute les instructions suivantes :

```
A = ["pomme", "poire", "banane"]
B = [1, 3, 8, 10, 25]
C = [[1, 2], [2, 3], [18, -1], [24, 12]]
D = [True, True, False, False, True, True]
```

Quel est le résultat de l'évaluation de :

- `len(A)`, `len(B)`, `len(C)` et `len(D)`.
- `A[1]`, `A[2]` et `A[-1]`.
- `B[2]`, `B[-1]` et `B[4]`.
- `C[0]`, `C[1]` et `C[-1]`.
- `D[0]`, `D[2]` et `D[-1]`.
- `2*A`.
- `3*B`.
- `3*D`.

Exercice 2 : construction par compréhension

On exécute les instructions suivantes :

```
A = [2*i+1 for i in range(5)]
B = [4*i-1 for i in range(5)]
C = [2*j+1 for j in range(5, 10)]
D = [2*j+1 for j in range(10, 20, 2)]
```

Que valent les variables A, B, C et D

Exercice 3 : construction par compréhension (bis)

On exécute les instructions suivantes :

```
A = [2*i+1 for i in range(5)]
B = [2*e+1 for e in A]
C = [e*e for e in B]
D = [2**e for e in B]
```

Que valent les variables A, B, C et D.

Exercice 4 : code par compréhension

Écrire le code permettant de construire par compréhension :

- la liste des carrés de tous les entiers compris entre 1 et 100.
- la liste `[2, 4, 6, 8, ..., 100]`.
- la liste des cubes de tous les entiers entre -50 et 50.

Exercice 5 : application d'une fonction

Étant donnée la fonction suivante :

```
def toto(x):  
    if x%2 == 0:  
        return 'p'  
    else:  
        return 'i'
```

Que donne l'évaluation de `[toto(x) for x in T]` avec T valant :

- `[5,8,17,22,23,40]`.
- `[10,13,27,40,12,20,29]`.

Exercice 6 : filtrage

On exécute les instructions suivantes :

```
L1 = [i for i in range(20) if i%2 == 0]  
L2 = [i for i in range(20) if i%3 == 1]  
L3 = [i for i in range(30) if i%3 == 0 or i%5 == 0]  
L4 = [i*5 for i in L3 if i%2 == 0]
```

Que valent les variables L1, L2, L3 et L4.

Exercice 7 : utilisation de méthodes

On exécute les instructions suivantes :

```
L = [2,9,7,12,7,3,15,18]  
L.append(1)  
L.remove(7)  
L.sort()  
L.reverse()
```

Que vaut la variable L.

Écriture de fonctions

Exercice 8 : duplicateSiTaille

Écrire la fonction `duplicateSiTaille` qui prend comme paramètre une liste et renvoie cette liste détriée si sa longueur est inférieure ou égale à 3, dédoublée sinon.

Exemple :

- `duplicateSiTaille(["chat","chien"])` renvoie `["chat","chien","chat","chien"]`.
- `duplicateSiTaille(["fraise","pomme","poire","oignon"])` renvoie `["fraise","pomme","poire","oignon","fraise","pomme","poire","oignon"]`.

Exercice 9 : multiplie

Écrire la fonction `multiplie` qui :

- prend comme paramètre une liste (de trois éléments).
- crée une nouvelle liste constituée de 1 fois le premier élément, 2 fois le second et 3 fois le troisième.
- renvoie cette nouvelle liste.

Parcours de listes

Exercice 10 : parcours des éléments

Qu'affiche le code suivant :

```
L = [2,8,7,4,10,15]
L1 = [0]*(max(L)+1)
L2 = [0]*(max(L)+1)
for i in L:
    if i%2 == 0:
        L1[i] = i
    else:
        L2[i] = i
print(L1)
print(L2)
```

Exercice 11 : plus_un

Écrire une fonction `plus_un` qui prend comme paramètre une liste de nombres quelconques et renvoie cette liste où chaque valeur est augmentée de 1.

Exemple :

`plus_un([6,12,18,5])` renvoie `[7,13,19,6]`.

Exercice 12 : reverseL

Écrire la fonction `reverseL` qui :

- prend comme paramètre une liste.
- crée une nouvelle liste constituée des éléments dans l'ordre inverse.
- renvoie cette nouvelle liste.

Tableaux en 2 dimensions

Exercice 13 : liste de listes

On exécute l'instruction suivante :

```
T = [[2,3,4,1],[1,2,3,3],[2,0,1,2]]
```

Quel est le résultat de l'évaluation de :

- `len(T)`.
- `len(T[10])`.
- `T[1][1]`, `T[2][2]`, `T[2][0]` et `T[3][2]`.

Exercice 14 : « mutabilité »

Que vaut la liste référencée par la variable `mm` après l'exécution du code suivant ?

```
m = [1,2,3]
mm = [m,m,m]
m[0] = 100
```

Exercice 15 : coloriage

Représenter sous forme d'un tableau la valeur de la liste `M` après l'exécution du code suivant :

- code 1 :

```
M = []
for i in range(8):
    M.append([0,0,0,0,0,0,0,0])
for i in range(8):
    for j in range(i):
        M[i][j] = 'X'
```

- code 2 :

```
M = []
for i in range(8):
    M.append([0,0,0,0,0,0,0,0])
for i in range(8):
    for j in range(i+1,8):
        M[i][j] = 'X'
```