

Feuille_info 4 : Slicing et listes en compréhension.

- 1) On considère la liste : $L = [10, 20, 30, 40, 50, 60, 70, 80]$
 Evaluer les expressions suivantes :
- $L[2:6]$
 - $L[:5:2]$
 - $L[::-1]$
 - $L[1:-1]$
- 2) On considère la liste de listes : $M = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$
 Que renvoient les expressions suivantes ?
- $M[0][1:]$
 - $[\text{row}[0] \text{ for row in } M]$
 - $[x \text{ for row in } M \text{ for } x \text{ in row if } x \% 2 == 0]$
- 3) Écrire, en une seule ligne avec une liste en compréhension, une expression qui construit :
- La liste des carrés des entiers de 0 à 9.
 - La liste des couples (i, j) avec $0 \leq i < j < 4$.
 - La liste de listes représentant la table de multiplication de 1 à 9 (*chaque sous-liste correspond à une ligne de la table*).
- 4) On affecte à la variable L la liste $[2, 0, 3, 6, 5, 8, 9, 0, 3, 4, 6, 2, 4]$.
 Ecrire dans ce tableau la liste associée à L au cours du programme suivant.
On effectue les instructions les unes après les autres, à chaque ligne on transforme la liste de la ligne précédente.

Instructions	L
	$[2, 0, 3, 6, 5, 8, 9, 0, 3, 4, 6, 2, 4]$
$L[0] = 0$	
$L += [0]$	
$L[2:4] = []$	
$L[4:4] = [215, 12]$	
$L[5:5] = []$	
$L[0:4:2] = [13, 12]$	(*)

- 5) On affecte à la variable L la liste de listes $[[2, 0, 3], [6, 5, 8], [9, 0, 3], [4, 6, 2]]$.
 Indiquer dans le tableau suivant ce qu'affiche Python :

Instructions	Affichage
<code>print(L)</code>	$[[2, 0, 3], [6, 5, 8], [9, 0, 3], [4, 6, 2]]$
<code>print(L[1])</code>	
<code>print(L[1][2])</code>	
<code>print(L[1][0:2])</code>	
<code>print(L[0:2])</code>	
<code>print(L[0:2][0:2])</code>	(*)
<code>print([x[0:2] for x in L[0:2]])</code>	(*)

6) On considère les deux programmes suivants :

Programme 1

```
L = [0, 2, 0, 3, 6, 0, 8, 9, 0, 3, 0, 6, 2, 4]
n = len(L)
for k in range(n-1,-1,-1):
    # k va de n-1 à 0 par pas de -1
    if L[k] == 0:
        L[k:k+1] = []
print(L)
```

Programme 2

```
L = [0, 2, 0, 3, 6, 0, 8, 9, 0, 3, 0, 6, 2, 4]
n = len(L)
for k in range(n):
    # k va de 0 à n-1 par pas de 1
    if L[k] == 0:
        L[k:k+1] = []
print(L)
```

a) Lequel de ces deux programmes affichent le message d'erreur :

```
if L[k] == 0:

IndexError: list index out of range
```

Expliquer pourquoi.

b) Qu'affiche l'autre programme ?

7) **Listes en compréhension.**

a) On considère la ligne de code suivante :

```
L = [ [ i-j for j in range(3) ] for i in range(3) ]
```

Décrire avec précision ce qui est associé à la variable L à la fin de ce code.

b) On considère la ligne de code suivante :

```
L = [ i-j for j in range(3) for i in range(3) ]
```

Décrire avec précision ce qui est associé à la variable L à la fin de ce code.

c) On considère la fonction suivante :

```
def produit(A, B):
    """ A, B étant deux listes de listes de nombres, représentant
        des matrices respectivement de taille nxp et pxr.
        Cette fonction renvoie le produit AxB sous la forme d'une liste de listes de nombres"""
    n, p, r = len(A), len(B), len(B[0])
    C = [ [ sum([A[i][k]*B[k][j] for k in range(p)]) for j in range(r) ]
          for i in range(n) ]

    return C
```

Réécrire cette fonction sans aucune liste en compréhension.

Il est inutile de recopier la spécification.

8) **Une fonction mystère.**

```
def mystere(n, p):
    """ ... """
    L = [ [] ]
    for i in range(p-1):
        L = [ x + [k] for x in L
              for k in range(n+1) if sum(x+[k]) <= n ]
    return [ x + [n-sum(x)] for x in L ]
```

Remarque : les différentes variables utilisées dans cette fonction conservent le même type.

a) Quel est le type des variables L, x, et k ?

b) A l'exécution de cette fonction combien de tours fait la boucle for i ?

c) Dans cette question, on passe pour arguments $n = 3$ et $p = 3$. (On exécute `mystere(3, 3)`)
Quel est l'objet associé à L à la fin de chaque tour de boucle ?

d) Pour x une liste d'entiers et n un entier quelconques,
quelle est la somme des valeurs de la liste `x+[n-sum(x)]` ?

e) Que fait cette fonction ?