

# P-uplets et tableaux : Exercices

## Exercice 1 (p-uplets)

On considère la suite d'instructions donnée ci-contre. Quelles sont les valeurs affectées aux variables a, b et c à la fin de cette séquence d'instructions ?

Dans l'ordre : Mme M. 19

```
tuple1 = (19, -2.2, 888)
tuple2 = ("Mlle", "Mme", "M.")
a, b, c = tuple1
c, a = a, c
d, e, f = tuple2
(b, a) = (f, e)
```

## Exercice 2 (p-uplets)

On rappelle que la fonction randint(a, b) du module random retourne un nombre entier aléatoire compris entre a et b (b compris).

- 1) Écrire le prototype d'une fonction lancer\_trois\_des permettant de retourner un 3-tuple de 3 nombres entiers simulant le lancer de 3 dés cubiques.

paramètres : aucun

valeur de retour : 3-tuple de nombres entiers

- 2) Compléter le code de cette fonction dans l'encadré ci-contre.
- 3) On appelle cette fonction avec l'instruction suivante :  
`mon_lancer = lancer_trois_des()`

```
import random

def lancer_trois_des():
    a = random.randint(1, 6)
    b = random.randint(1, 6)
    c = random.randint(1, 6)
    return a, b, c
```

Quel est le type de la variable `mon_lancer` ? un 3-tuple

Quelle instruction permet alors de calculer la somme des trois dés ?

- a) ~~`somme = mon_lancer[1] + mon_lancer[2] + mon_lancer[3]`~~
- b) `somme = mon_lancer[0] + mon_lancer[1] + mon_lancer[2]`
- c) ~~`somme = mon_lancer + mon_lancer + mon_lancer`~~

- 4) On appelle cette fonction avec l'instruction suivante :  
`de_1, de_2, de_3 = lancer_trois_des()`

Cette instruction génère-t-elle une erreur ? non

Si oui pourquoi ?

Si non, quelle instruction permet de calculer alors la somme des trois dés ?

`somme = de_1 + de_2 + de_3`

## Exercice 3 (p-uplets)

On considère les deux instructions ci-dessous en langage python. Lors de l'exécution elle provoque une erreur.

```
>>> debut_alphabet = ('a', 'b', 'c', 'd', 'd', 'f', 'g', 'h', 'i', 'j')
>>> debut_alphabet[4] = 'e'
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-7-131a7cf1d918> in <module>()
      1 tuple1 = ('a', 'b', 'c', 'd', 'd', 'f', 'g', 'h', 'i', 'j')
----> 2 tuple1[4] = 'e'

TypeError: 'tuple' object does not support item assignment
```

Pourquoi ?

**Car un tuple en python est immutable : on ne peut modifier les valeurs de ses éléments**

#### **Exercice 4 (tableaux)**

On considère la suite d'instructions données ci-contre.

Donner l'état des deux tableaux à la fin de la suite d'instructions.

```
[7, 55, 55, 7777]
[5, 7, 555, 7]
```

```
tab_x = [7, 77, 777, 7777]
tab_y = [5, 55, 555, 5555]
tab_x[2] = tab_y[1]
tab_y[3] = tab_x[0]
tab_x[1] = tab_x[2]
tab_y[1] = tab_y[3]
```

#### **Exercice 4-bis (Lire et modifier les éléments d'un tableau)**

On considère la suite d'instructions données ci-contre.

Donner les valeurs de chacune des variables a à f.

```
a=5
b='paf'
c='paf'
d='www'
e='www'
f='xxx'
```

```
tab = ['z', 'yy', 'xxx', 'www', 'paf']

a = len(tab)
b = tab[ len(tab)-1 ]
c = tab[-1]
d = tab[ len(tab)-2 ]
e = tab[-2]
f = tab[-3]
```

#### **Exercice 5 (tableaux construits par compréhension)**

Pour chacune des instructions ci-dessous, écrire le tableau qui a été créé.

```
tab_a = [ 2**x for x in range(11) ]
tab_b = [ 7 * x%2 for x in range(11) ]
tab_c = [ 7 * ((10**x) // 9) for x in range(1, 5) ]
tab_d = [ 'M. ' + car + ' ?' for car in 'XYZ' ]
```

```
[1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]
[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
[7, 77, 777, 7777]
['M. X ?', 'M. Y ?', 'M. Z ?']
```

#### **Exercice 6 (Itérer sur les éléments d'un tableau)**

On rappelle que la fonction `len()` permet de retourner la longueur d'un tableau mais aussi la longueur d'une chaîne de caractères.

```
>>> len("Coucou")
6
>>> len("Pardon ?")
8
```

1. Combien de noms vont être affichés à l'issue de la séquence d'instructions ci-dessous ? **5 (noms avec un "o")**

```
grands_noms = ["Lovelace", "Clarke", "Goldstine", "Hopper", "Recoque", "Hamilton"]
for nom in grands_noms:
    if "o" in nom:
        print(nom)
```

2. Quelle sera la valeur de `quantite_mystere` à la fin de la séquence d'instructions ci-dessous ? **44 (somme des nombres de caractères de chacun des noms)**

```
grands_noms = ["Lovelace", "Clarke", "Goldstine", "Hopper", "Recoque", "Hamilton"]
quantite_mystere = 0
for nom in grands_noms:
    quantite_mystere = quantite_mystere + len(nom)
```

### Exercice 7 (Utiliser des tableaux de tableaux pour représenter des matrices)

On représente la matrice donnée ci-contre grâce au tableau `ma_belle_matrice`.

```
ma_belle_matrice = [ ['a', 'b', 'c', 'd'],
                      ['e', 'f', 'g', 'h'],
                      ['i', 'j', 'k', 'l'],
                      ['m', 'n', 'o', 'p'],
                      ['q', 'r', 's', 't'] ]

a = ma_belle_matrice[3]
b = ma_belle_matrice[3][2]
c = ma_belle_matrice[1][2]
d = ma_belle_matrice[2][1]
```

- 1) Donner les valeurs de chacune des variables a, b, c et d

`['m', 'n', 'o', 'p']`

`o`  
`g`  
`j`

- 2) Proposer une suite d'instructions permettant de modifier `ma_belle_matrice` afin qu'elle corresponde à la matrice donnée ci-contre.

```
ma_belle_matrice[1] = ['X', 'X', 'X', 'X']
```

```
for ligne in range(5):
    ma_belle_matrice[ligne][2] = 'X'
```

ou alors :

```
ma_belle_matrice[1][0] = 'X'
ma_belle_matrice[1][1] = 'X'
ma_belle_matrice[1][2] = 'X'
ma_belle_matrice[1][3] = 'X'
```

```
ma_belle_matrice[0][2] = 'X'
ma_belle_matrice[1][2] = 'X'
ma_belle_matrice[2][2] = 'X'
ma_belle_matrice[3][2] = 'X'
ma_belle_matrice[4][2] = 'X'
```

'a'	'b'	'X'	'd'
'X'	'X'	'X'	'X'
'j'	'k'	'X'	'm'
'n'	'o'	'X'	'q'
'r'	's'	'X'	'u'

### Exercice 8 (Doubles boucles et coloriages)

### Cas numéro 1

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

Pour i=0, j va aller de euh .. nulle part.

Pour i= 1, j va aller de 0 à ... 0.

Pour i=2, j va aller de 0 à ... 1.

Pour i=3, j va aller de 0 à ... 2.

X							
X	X						
X	X	X					
X	X	X	X				
X	X	X	X	X			
X	X	X	X	X	X		
X	X	X	X	X	X	X	

### Cas numéro 2

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

	X	X	X	X	X	X	X
		X	X	X	X	X	X
			X	X	X	X	X
				X	X	X	X
					X	X	X
						X	X
							X

### Cas numéro 3

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

X	X	X	X	X	X	X	
X	X	X	X	X	X		
X	X	X	X	X			
X	X	X	X				
X	X	X					
X	X						
X							

### Cas numéro 4

```
for i in range(0, 8):
    for j in range(3, 5):
        M[i][j] = 'X'
```

			X	X			
			X	X			
			X	X			
			X	X			
			X	X			
			X	X			
			X	X			
			X	X			

### Cas numéro 5

```
for i in range(3, 5):
    for j in range(1, 7):
        M[i][j] = 'X'
```

	X	X	X	X	X	X	
	X	X	X	X	X	X	

### Cas numéro 6

```
for i in range(1, 8):
    for j in range(8-i, 8):
        M[i][j] = 'X'
```

							X
						X	X
					X	X	X
				X	X	X	X
			X	X	X	X	X
		X	X	X	X	X	X
	X	X	X	X	X	X	X