

## TYPES CONSTRUIITS - EXERCICES

### 1. TUPLES ET LISTES (QCM):

#### a. Parmi les affirmations suivantes, lesquelles sont vraies ?

- Une liste peut contenir plusieurs éléments.
- On peut ajouter des éléments à un tuple.
- On peut modifier les éléments d'une liste.
- Un tuple peut contenir à la fois des nombres et des chaînes de caractères.

#### b. Si *liste* est la liste [1, 3, 5], quelles sont les opérations valides ?

- `liste.append(4)`
- `liste[0]`
- `liste[0] = 4`
- `liste[4] = 7`
- `liste = [1, 3, 10, 7, 3]`

#### c. Si *triplet* est le tuple (1, 3, 5), quelles sont les opérations valides ?

- `triplet.append(4)`
- `triplet[0]`
- `triplet[0] = 4`

### 2. CALCUL D'UN MOYENNE:

fichier 5-moyenne.py

A la fin du semestre, il est temps de calculer les moyennes de français avant l'édition des bulletins. Pronote exporte les notes de Marta et Tiago sous la forme d'une liste:

```
marta= [10, 15, 11, 18, 7, 12, 13, 16, 18]
tiago= [8, 11, 12, 14, 10, 9, 15, 12, 11,13,18]
```

Téléchargez et ouvrez le fichier python `5-moyenne.py`. Le professeur a coché la case "*arrondir la moyenne au dixième*".

- a. Compléter la fonction `moyenne(liste)` qui permet de calculer la moyenne d'un élève arrondie au dixième.

```
# calcul de la somme des notes de liste à l'aide d'une boucle for
# calcul et retour de la valeur moyenne arrondie
# appel de la fonction dans le corps du programme pour Marta et Tiago
```

- b. Le professeur décide de cocher la case "*enlever la plus mauvaise note*". A l'aide de la fonction `indexMini(liste)`, affichez la moyenne corrigée de Marta et Tiago.

### 3. DICTIONNAIRES (QCM):

```
dico = {"a": True, "b": False, "c": False}
```

#### a. Quelle est la valeur de `dico[1]` ?

- "a"
- True
- "b"
- False
- Rien car l'expression n'est pas valide.

#### b. Quelle est la valeur de `dico["a"]` ?

- True
- False
- Rien car l'expression n'est pas valide.

#### c. Quelle instruction permet de modifier le dictionnaire de façon à ce que sa nouvelle valeur soit `{"a": True, "b": False, "c": False, "e": True}`?

- `dico["e"] = True`
- `dico.append("e")`
- `dico.append("e", True)`
- Ce n'est pas possible car un dictionnaire n'est pas modifiable.

#### d. Quels sont les affichages possibles lors de l'exécution du code suivant ?

```
for i in dico.keys():
    print(i, end=" ")
```

- a b c
- (a, True) (b, False) (c, False)
- True False False

#### e. Quels sont les affichages possibles lors de l'exécution du code suivant ?

```
for i in dico.items():
    print(i, end=" ")
```

- a b c
- (a, True) (b, False) (c, False)
- True False False

### 4. COMPREHENSIONS ET STRUCTURES IMBRIQUEES (QCM):

#### a. Si *liste* désigne la liste [1, [2, 3], [4, 5], 6, 7], que vaut `len(liste)` ?

- 1
- 3
- 5
- 7

#### b. Que vaut `[2 * n for n in range(5)]` ?

- [0, 2, 4, 6, 8]
- [0, 2, 4, 6, 8, 10]
- [0, 2, 4]
- Autre chose

#### c. Supposons que *liste* = [-5, 2, 3, -7, 42, 7]. Que vaut `[n for n in liste if n > 0]` ?

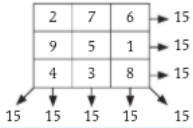
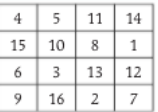
- [-5, 2, 3, -7, 42, 7]
- [2, 3, 42, 7]
- [False, True, True, False, True, True]
- Autre chose

## 5. CARRE MAGIQUE

Un carré d'ordre  $n$  est un tableau carré contenant  $n^2$  entiers strictement positifs. On dit qu'un carré d'ordre  $n$  est magique si :

- il contient tous les nombres entiers de 1 à  $n^2$ ;
- les sommes des nombres de chaque rangée, les sommes des nombres de chaque colonne et les sommes des nombres de chaque diagonale principale sont égales.

On modélise un carré par une liste de listes de nombres. Exemples :

Carré d'ordre $n$	Modélisation proposée
	<pre>carre3 = [     [2, 7, 6],     [9, 5, 1],     [4, 3, 8] ]</pre>
	<pre>carre4 = [     [4, 5, 11, 14],     [15, 10, 8, 1],     [6, 3, 13, 12],     [9, 16, 2, 7] ]</pre>

1. a) Quelle est la valeur de `len(carre4)` ?  
b) Quelle est la valeur de `carre3[1]` ?  
c) Quelle est la valeur de `carre3[0][2]` ?  
d) Quelle instruction permet de récupérer la valeur 3 de `carre4` ?

2.

- a) On propose le code suivant:

```
def somme_ligne(carre,n):
    """
    carre est un tableau carré de nombres
    n est un nombre entier
    """
    somme=0
    for i in carre[n]:
        somme+=i
    return somme
```

Que vaut `somme_ligne(carre4, 2)` ? À quoi sert cette fonction ?

- b) Écrire le code d'une fonction qui prend un carré en paramètre et qui vérifie que les sommes des nombres de chaque ligne sont égales.
- c) Proposer le code d'une fonction qui prend un carré en paramètre, ainsi que le numéro d'une colonne, et qui renvoie la somme des nombres de cette colonne.

## 6. LA RECHERCHE DU MINIMUM

Ecrivez votre propre fonction qui permet de trouver la valeur minimum d'un liste de nombre et son index.