

TP9 : Types construits

Exercice 1

A la fin du trimestre, le professeur de NSI calcule les moyennes avant l'édition des bulletins.

Le logiciel *Pronote* exporte les notes de Clara et Jean sous la forme d'une liste :

```
Clara = [10, 15, 11, 18, 7, 12, 13, 16, 18]
```

```
Jean = [8, 11, 12, 14, 10, 9, 15, 12, 11, 13, 18]
```

Le professeur a coché la case "*arrondir la moyenne au dixième*".

1. Télécharger et ouvrir le fichier Python `moyenne.py`.
2. Compléter la fonction `moyenne(liste)` qui permet de calculer la moyenne d'un élève arrondie au dixième en suivant les étapes ci-dessous :
 - ☐ 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 Clara et Jean
3. Le professeur décide de cocher la case "*enlever la plus mauvaise note*".
À l'aide de la fonction `indexMini(liste)`, afficher la moyenne corrigée de Clara et Jean.

Exercice 2

Rappel : Construction de listes par compréhension : On peut construire un tableau en insérant une boucle « `for` » à l'intérieur de la construction du tableau.

Exemples :

Nombres pairs de 0 à 20 :

```
>>> [i for i in range(0,21,2)]  
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Carrés des multiples de 3 compris entre 0 et 21 :

```
>>> [j**2 for j in range(0,22,3)]  
[0, 9, 36, 81, 144, 225, 324, 441]
```

1. Construire en une seule instruction la liste `[-5, -1, 3, 7, 11, 15, 19, 23, 27, 31]`.
2. Construire en une seule instruction la liste `[0, 2, 8, 18, 32, 50, 72, 98, 128, 162]`.
3. Construire en une seule instruction une liste composée de 10 fois la ligne suivante :
`[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`.

4. Construire en une seule instruction un tableau pyramidal de 10 lignes, où la première liste est $[\emptyset]$, la deuxième $[\emptyset, 1]$, etc.

Exercice 3

Écrire une fonction `somme(tup)` avec un paramètre « *tuple de longueur variable* » qui calcule la somme des nombres contenus dans le tuple.

Exemple :

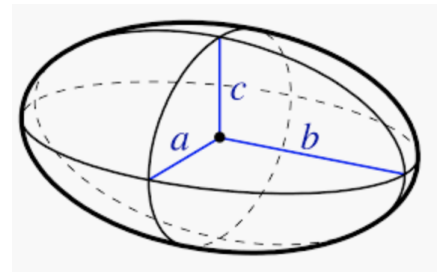
```
>>> somme ( (1, 15, 8.2, 7) )
31.2
```

Exercice 4

Écrire une fonction `VolMasseEllipsoide(a,b,c,mu)` qui retourne le volume et la masse d'un ellipsoïde grâce à un tuple.

Les paramètres sont les trois demi-axes et la masse volumique.

On rappelle que : $V = \frac{4}{3}\pi \cdot a \cdot b \cdot c$



(Exemples d'ellipsoïdes : certains astres, les ventricules du myocarde ...)

Exemple :

```
>>> VolMasseEllipsoide(10,8,6,0.19)
(2010.62, 382.02)
```

Exercice 5

Un jeu de 52 cartes est constitué de :

- 4 couleurs : *Pique, Carreau, Cœur* et *Trèfle*
- 13 valeurs : 2, 3, 4, 5, 6, 7, 8, 9, 10, *Valet, Dame, Roi* et *As*

1. Créer deux listes :

- a. La liste couleurs
- b. La liste valeurs constituée de tuples contenant la valeur en entier et la valeur en chaîne :

Exemple : (2, "Deux") (11, "Valet")

2. Écrire une fonction `creation_jeu()` qui crée et renvoie le jeu de cartes complet sous forme d'une **liste de 52 tuples** ayant la forme suivante :

`[("Pique", 1, "As"), ("Pique", 2, "Deux"), ..., ("Pique", 12, "Roi"), ..., etc.]`

3. On veut simuler une distribution de cartes pour 4 joueurs. Écrire une fonction `distribution()` qui mélange les cartes puis les séparent en **4 listes de 13 cartes**.

Par exemple, on obtiendra des listes de tuples ayant la forme suivante :

```
>>> distribution()
Le jeu du joueur 1 :
[('Pique', 14, 'As'), ('Coeur', 9, 'Neuf'), ('Trefle', 12, 'Dame'), ('Coeur', 14, 'As'), ('Carreau', 12, 'Dame'), ('Coeur', 12, 'Dame'), ('Pique', 6, 'Six'), ('Trefle', 5, 'Cinq'), ('Trefle', 2, 'Deux'), ('Pique', 3, 'Trois'), ('Trefle', 4, 'Quatre'), ('Carreau', 5, 'Cinq'), ('Coeur', 6, 'Six')]
#####
Le jeu du joueur 2 :
[('Trefle', 14, 'As'), ('Pique', 9, 'Neuf'), ('Carreau', 14, 'As'), ('Coeur', 10, 'Dix'), ('Coeur', 8, 'Huit'), ('Carreau', 2, 'Deux'), ('Carreau', 6, 'Six'), ('Carreau', 7, 'Sept'), ('Carreau', 8, 'Huit'), ('Pique', 11, 'Valet'), ('Trefle', 7, 'Sept'), ('Coeur', 7, 'Sept'), ('Coeur', 5, 'Cinq')]
#####
..., etc.
```

Conseil : pour mélanger aléatoirement les éléments d'une liste, on utilise la bibliothèque `random` et l'instruction `random.shuffle(liste)`.

Exercice 6

Écrire un programme Python qui permet de créer à partir d'un entier `n` saisi au clavier, un dictionnaire formé des entiers de 1 à `n` et de leurs carrés. Exemple pour `n = 7` le dictionnaire sera de la forme :

`{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49}`

Exercice 7

Écrire un programme en Python qui demande à l'utilisateur de saisir une chaîne de caractères, et de lui renvoyer un dictionnaire dont les clés sont les caractères de la chaîne saisie et les valeurs sont les nombres d'occurrences des caractères dans la chaîne.

Exemple : pour la chaîne `s = « langage »`, le programme renvoie le dictionnaire

`d = {'l':1, 'a':2, 'n':1, 'g':2, 'e':1}`

Exercice 8

Écrire une fonction en Python qui prends en paramètre une liste de nombres entiers et qui renvoie un dictionnaire dont les clés sont les entiers de la liste et dont les valeurs sont 'pair' ou 'impair' selon la parité du nombre.

Exercice 9

1. Créer une fonction `compte_mots_2_lettres()` qui prend comme argument une séquence sous la forme d'une chaîne de caractères et qui renvoie tous les mots de 2 lettres qui existent dans la séquence sous la forme d'un dictionnaire.

Exemple : pour la séquence ACCTAGCCCTA, le dictionnaire renvoyée serait :

```
{'AC': 1, 'CC': 3, 'CT': 2, 'TA': 2, 'AG': 1, 'GC': 1}
```

2. Créez une nouvelle fonction `compte_mots_3_lettres()` qui a un comportement similaire à `compte_mots_2_lettres()` mais avec des mots de 3 lettres.
3. Utilisez ces fonctions pour affichez les mots de 2 et 3 lettres et leurs occurrences trouvés dans la séquence d'ADN :

ACCTAGCCATGTAGAATCGCCTAGGCTTTAGCTAGCTCTAGCTAGCTG

Voici un exemple de sortie attendue :

```
1  Mots de 2 lettres
2  AC : 1
3  CC : 3
4  CT : 8
5  [...]
6  Mots de 3 lettres
7  ACC : 1
8  CCT : 2
9  CTA : 5
10 [...]

```

Exercice 10

Au jardin zoologique de Singapour, il y a 5 éléphants d'Asie, 23 cheetas, 13 girafes ...

On représente cet inventaire à l'aide d'un dictionnaire, de la façon suivante :

```
zoo_Singapour={ 'éléphant': ('Asie', 5), 'orangutan': ('Asie', 32), 'tigre blanc': ('Asie', 4), 'cheetah': ('Afrique', 23), 'girafe': ('Afrique', 7)}
```

On représente de la même façon le zoo de Paris :

```
zoo_Paris = { 'ours': ('Europe', 4), 'tigre': ('Asie', 7), 'girafe': ('Afrique', 11), 'hippopotame': ('Afrique', 3)}
```

On souhaite créer la fonction `plus_grand_nombre()` qui prend un zoo en paramètre et qui renvoie le nom de l'animal le plus représenté dans ce zoo.

Par exemple :

```
plus_grand_nombre(zoo_Singapour) == 'orangutan'
plus_grand_nombre(zoo_Paris) == 'girafe'
```

1.

a. Quel type de boucle peut-on envisager pour le code de cette fonction ?

- | | |
|---|--|
| <input type="checkbox"/> <code>for i in dico.keys()</code> | <input type="checkbox"/> <code>for i in dico.values()</code> |
| <input type="checkbox"/> <code>for (i,j) in dico.items()</code> | <input type="checkbox"/> Aucune boucle |

b. Écrire et tester le code de cette fonction.

2. On souhaite créer une fonction `nombre_total` qui prend un zoo en paramètre ainsi que le nom d'un continent, et qui renvoie le nombre d'animaux originaires de ce continent dans le zoo.

Par exemple :

```
nombre_total(zoo_Singapour, 'Afrique') == 30
nombre_total(zoo_Paris, 'Asie') == 7
```

a. Quel type de boucle peut-on envisager pour le code de cette fonction ?

- | | |
|---|--|
| <input type="checkbox"/> <code>for i in dico.keys()</code> | <input type="checkbox"/> <code>for i in dico.values()</code> |
| <input type="checkbox"/> <code>for (i,j) in dico.items()</code> | <input type="checkbox"/> Aucune boucle |

b. Écrire et tester le code de cette fonction.

3. On souhaite créer une fonction `nombre` qui prend un zoo en paramètre ainsi que le nom d'un animal, et qui renvoie le nombre de représentants de cet animal dans le zoo.

Par exemple :

```
nombre(zoo_Paris, 'orangutan') == 0
```

```
nombre(zoo_Singapour, 'cheeta') == 23
```

a. Quel type de boucle peut-on envisager pour le code de cette fonction ?

☐ `for i in dico.keys()`

☐ `for i in dico.values()`

☐ `for (i,j) in dico.items()`

☐ Aucune boucle.

b. Écrire et tester le code de cette fonction.