1 Les différents types de données

1.1 Tableau (list)

Un tableau contient des éléments de même type (entiers, booléens…) repérés par leur indice (position) dans le tableau. Les indices commencent à zéro.

```
tab = [3, 5, 9, 8]
# lire un élément
tab[2] # renvoie l'entier 9

# écrire un élément
tab[2] = 10 # le 9 est remplacé par 10
```

En Python, les tableaux se nomment des **`list`**.

1.2 Dictionnaire

Un dictionnaire contient des éléments repérés par une clé. Une clé est un élément non mutable : entier, chaîne de caractères, tuple.

```
dico = {"prems": 18, "deuz": 13, "troiz": 9}
# lire un élément
dico["deuz"] # renvoie l'entier 13

# écrire un élément
tab["deuz"] = 10 # le 13 est remplacé par 10
```

1.3 Tuple

Un tuple est un n-uplet non mutable.

```
tup = (3, 5, 9, 8)
# lire un élément
tup[2] # renvoie l'entier 9

# Il n'est pas possible de modifier le contenu d'un tuple.
```

1.4 Construction par compréhension

En Python il est possible de construire une structure de données de manière rapide et efficace.

```
tab = [0 for i in range(5)]
# tab = [0, 0, 0, 0, 0]

dico = {i: 0 for i in range(5)}
# dico = {0:0, 1:0, 2:0, 3:0, 4:0}

tup = (0 for i in range(5))
# tup = (0, 0, 0, 0, 0)
```

Code 1 – Construction par compréhension

Christophe Viroulaud

2 Applications

2.1 Construction par compréhension

Exercice 1 : Construire par compréhension le tableau `[1, 1, 1, 1, 1]`.

Exercice 2 : Construire par compréhension le tableau `[0, 1, 2, 3, 4]`.

Exercice 3 : Construire par compréhension le tuple `[4, 3, 2, 1, 0]`.

Exercice 4 : Construire par compréhension le tuple `[0, 2, 4, 6, 8]`.

Exercice 5 : Construire par compréhension le dictionnaire `[0:1, 1:1, 2:1, 3:1, 4:1]`.

Exercice 6 : Construire par compréhension le dictionnaire `["A":0, "B":1, "C":2, "D":3, "E":4]`. La fonction native **`chr()`** renvoie le caractère correspondant au code ASCII donné.

```
chr(65) # renvoie A
chr(66) # renvoie B
```

Exercice 7 : Construire par compréhension un tableau de dix entiers aléatoires compris entre 0 et 100. Il sera nécessaire d'utiliser la bibliothèque **`random`**.

2.2 Utilisation de structure de données

Exercice 8 :

1. Construire par compréhension un tableau de dix entiers aléatoires compris entre 0 et 100.
2. Écrire la fonction **`maxi(tab: list)`** $\rightarrow$ **`int`** qui renvoie le plus grand élément du tableau.

Exercice 9 :

1. Construire par compréhension un tuple de dix entiers aléatoires compris entre 0 et 1000.
2. Écrire la fonction **`somme(tup: tuple)`** $\rightarrow$ **`int`** qui renvoie la somme de tous les entiers du tuple.

Exercice 10 :

1. Construire un tableau de 8 mots.
2. Demander deux indices i et j à l'utilisateur.
3. Échanger les mots aux indices i et j.

Exercice 11 : Un livre peut être caractérisé par son titre, son auteur, son éditeur, son prix.

1. Construire un dictionnaire qui contient les informations du livre : *Il était deux fois* de Franck Thilliez aux éditions *Poche* à 8,70€.
2. Construire un dictionnaire pour *Fahrenheit 451* de Ray Bradbury aux éditions *Folio* à 6,30€.
3. Construire un tableau contenant les deux dictionnaires. Ajouter au moins un autre livre.
4. Écrire une boucle qui parcourt le tableau et affiche l'auteur de chaque livre.

Exercice 12 : Écrire la fonction **`lettres(mot: str)`** $\rightarrow$ **`dict`** qui renvoie un dictionnaire contenant le nombre d'occurrences de chaque lettre de *mot*. Par exemple :

```
>>> lettres("bonjour")
>>> {"b": 1, "o": 2, "n": 1, "j": 1, "u": 1, "r": 1}
```


Christophe Viroulaud