1 - Compléments sur les listes

Matrices

En mathématiques, une matrice est une liste de listes de même longueur. La dimension de la matrice est donnée par son nombre de lignes n et son nombre de colonnes p, sous la forme $n \times p$.
En informatique, on gardera ces notations. Mais on peut aller plus loin encore, en faisant une liste de listes de listes de... (listeception, pour ceux qui ont vu le film :-). On utilise cette structure de données rarement.
L'affichage des matrices est plus agréable ligne par ligne.

In [ ]:
```
mat3x2 = [[6,7],[8,9],[10,11]]
mat2x3 = [[6,7,8],[9,10,11]]
print("matrice de dimension 3x2")
for i in range(len(mat3x2)):
    print(mat3x2[i])
print()
print("matrice de dimension 2x3")
for i in range(len(mat2x3)):
    print(mat2x3[i])
```

L'accès à un élément se fait par `matrice[i][j]`. L'utilisation de la syntaxe `[a:b]` pour une sélection d'éléments ne donne pas les résultats que l'on pourrait attendre, elle est donc à éviter. Testez ci-dessous avec une matrice aléatoire de dimension 10×10.

In [ ]:
```
from random import *
matAlea = []
for i in range(10):
    matAlea.append([randint(1,100) for i in range(10)])
for i in range(len(matAlea)):
    print(matAlea[i])

print()
print(matAlea[3][6])
print(matAlea[3:6][0:2])
```

Construction de tableaux par compréhension

On dit qu'une liste est construite par compréhension lorsque l'on crée tous les éléments de la liste à l'aide d'une boucle for entre les crochets, comme dans le programme précédent `ligne = [randint(1,100) for i in range(10)]` **Exercices** : on reprend

In [ ]:
```
liste = [i for i in range(0,11)]
print(liste)
```

Exercice : créer avec cette méthode la liste des multiples de 3 compris entre -6 et 24, vérifier avec un affichage.

Exercice : créer en une seule ligne de code une liste de 20 nombres aléatoires compris entre -10 et 100.

Exercice : créer une liste qui contient le double des cubes des nombres impairs compris entre -3 et 5

2 - Les tuples

Un tuple, ou n-uplet, est un type données Python constitués d'éléments séparés par des virgules, et encadré par des parenthèses. Un tuple ressemble à une liste, mais un tuple n'est pas modifiable. Testez:

In [ ]:
```
eleve = ('Archibald','NSI','mathématiques','coloriage')
print(len(eleve))
print(eleve[0])
print(eleve[-1])
for element in eleve:
    print(element,end = " ; ")
print()
eleve = eleve + ('chess boxing',)    # on peut concaténer un autre tuple, ici avec un seul élément avec une virgule
print(eleve)
```

Testez les modifications :

In [ ]:
```
eleve[1] = 'Diablo 4'
print(eleve)
```

In [ ]:
```
eleve.insert("candy crush",2)
```

In [ ]:
```
eleve.remove('NSI')
```

Les tuples sont utilisés lorsque l'on souhaite que les données ne soient pas modifiables par erreur.

3 - Les dictionnaires

Un dictionnaire est un type de données Python, qui ressemble à une liste mais n'est pas ordonné. On ne peut donc pas accéder à un élément par index. A la place, on utilise une clé. L'analogie avec les dictionnaires usuel est évidente : la clé est le mot, la valeur associée à la clé est la définition du mot (dictionnaire français), ou sa traduction (dictionnaire français-anglais). Un dictionnaire s'écrit entre accolades sous la forme {*clé1 : valeur1* , *clé2 : valeur2* , ... ,*cléN : valeurN*}.

Création et modification d'un dictionnaire

Exemple de création:

In [ ]:
```
terreur = {}
terreur['film'] = 'Alien'
terreur['réalisateur'] = 'Ridley Scott'
terreur['actrice'] = 'Sigourney Weaver'
terreur['acteur'] = 'bisounours'
terreur['sortie'] = 1979
print(terreur)
print("longueur de terreur : ",len(terreur))
for cle in terreur:
    print("clé : ",cle," , valeur associée : ",terreur[cle])
```

Exemple de modification :

In [ ]:
```
del terreur['acteur']
print(terreur)
```

Contre-exemple pour la modification :
On reprend le dictionnaire initial et on essaie de supprimer 'bisounours' sans passer par sa clé.

In [ ]:
```
terreur = {'film': 'Alien', 'réalisateur': 'Ridley Scott', 'a
ctrice': 'Sigourney Weaver',\
           'acteur': 'bisounours', 'sortie': 1979}
del terreur['bisounours']
```

Quel est le type d'erreur renvoyé ? *Réponse* : Que doit-on écrire pour remplacer 'bisounours' par 'Ian Holm' (*note* : c'est l'acteur qui joue Bilbo dans le Seigneur des anneaux, il était jeune).

Quelques méthodes utiles:

In [ ]:
```
print(terreur.keys())
```

In [ ]:
```
print(terreur.values())
```

In [ ]:
```
elements = terreur.items()
print(elements)
print()
elements = tuple(elements)    # renvoie une liste de tuples
print(elements)
```

In [ ]:
```
print("année de sortie : ",terreur.get('sortie'))
print("décorateur : ",terreur.get('decorateur'))
print("comparaison avec terreur['decorateur'] : ")
print("décorateur : ",terreur['decorateur'])
```

On peut supprimer un élément associé à une clé, tout en récupérant la valeur associée, grâce à la méthode `.pop(_clé_)`

In [ ]:
```
acteur1 = terreur.pop('acteur')
print(terreur)
print(acteur1)
```

Contre-exemple pour l'accès la valeur de clé "film":

In [ ]:
```
print(terreur[0])
```

Quelle est l'erreur commise ci-dessus ?

Construction par compréhension

Exemple : Construction du dictionnaire des points de code ASCII des minuscules.
Question : avant d'exécuter le code suivant, précisez ce que sont les clés et les valeurs.

In [ ]:
```
asciiMin = {chr(n) : n for n in range(97,123)}
print(asciiMin)
```

Copie de dictionnaires

Comme pour les listes, la copie de dictionnaires est délicate.
Testez :

In [ ]:
```
terreur = {'film': 'Alien', 'réalisateur': 'Ridley Scott', 'a
ctrice': 'Sigourney Weaver',\
           'acteur': 'bisounours', 'sortie': 1979}
print("dictionnaire 'terreur', film de terreur : ",terreur['f
ilm'])
print()
scienceFiction = terreur
scienceFiction['film'] = 'Interstellar'
print("film de science fiction : ",scienceFiction['film'])
print("film de terreur : ",terreur['film'])
```

On utilise la méthode `.copy()` pour remédier à ce problème.

In [ ]:
```
terreur = {'film': 'Alien', 'réalisateur': 'Ridley Scott', 'a
ctrice': 'Sigourney Weaver',\
           'acteur': 'bisounours', 'sortie': 1979}
print("dictionnaire 'terreur', film de terreur : ",terreur['f
ilm'])
print()
scienceFiction = terreur.copy()
scienceFiction['film'] = 'Interstellar'
print("film de science fiction : ",scienceFiction['film'])
print("film de terreur : ",terreur['film'])
```

Remarque : nous verrons dans le notebook "types mutables" une explication de ce comportement.

Frederic Mandon (mailto:frederic.mandon@ac-montpellier.fr), Lycée Jean Jaurès - Saint Clément de Rivière - France (2015-2019)