

Structure de données

Quelques rappels

En informatique on manipule essentiellement des données. Lorsqu'elles sont simples comme des nombres, des chaînes de caractères ou des booléens, on peut les stocker dans des variables qui seront typées en fonction de la nature de la donnée.

Par exemple, avec les affectations suivantes:

```
n=4
p=-5
r=0.7
phrase="hello"
test=True
```

On obtient en retour les types des variables : (Python est un langage où le typage est dynamique, c'est à dire que le type de la variable est automatiquement déterminé la première fois qu'elle est déclarée)

```
>>> type(n)
<class 'int'>
>>> type(p)
<class 'int'>
>>> type(r)
<class 'float'>
>>> type(phrase)
<class 'str'>
>>> type(test)
<class 'bool'>
```

Lorsque l'on a besoin de manipuler un grand nombre de données, qu'elles soient de même type ou pas, on utilise des structures de données comme les tuples, les listes, les tableaux ou les dictionnaires.

Les chaînes de caractères, les tuples et les listes sont des séquences, c'est à dire des suites ordonnées d'éléments indexés par une suite d'entiers

Avec :

- Chaînes de caractères et tuples sont non mutables (on ne peut les modifier)
- Les listes sont mutables
- Un tableau est une liste de liste
- Tuples et listes peuvent contenir des éléments de n'importe quel type, y compris d'autres tuples, listes ou encore des dictionnaires...
- Chacun de ces types dits construits possède un certain nombre de méthodes qui permettent d'agir sur l'objet (ajout, suppression, tris etc...)

Dans un dictionnaire les éléments sont indexés par des clés et sont modifiables.

Un exemple...

On voudrait stocker dans une seule variable, pour un élève les données suivantes:

- Nom : Térieur
- Prénom : Alain
- Date de naissance : 01/01/2000
- Programmation: 12
- Algorithmique : 10
- Projet : 15

Avec un tuple (les données ne seront pas modifiables):

```
eleve1=("Térieur", "Alain", "01/01/2000", 12, 10, 15)
```

Avec un tuple contenant des listes(modifiables):

```
eleve1=(["Térieur", "Alain", "01/01/2000"], [12, 10, 15])
```

Une liste, une liste de listes, une liste de tuples...

Un dictionnaire :

```
eleve1={"Nom": "Térieur", "Prenom": "Alain", "Date": "01/01/2000",  
"Programmation": 12, "Algorithmique": 10, "Projet": 15}
```

Si maintenant on souhaite, le faire pour une classe entière, on pourra stocker les données dans une variable: classe1=[eleve1, eleve2, ...](ou un tuple ou encore un dictionnaire...)

On peut créer une fonction qui calcule la moyenne d'un élève:

```
def moyenne(eleve):  
    '''  
    eleve : type dict  
    clés(type(valeur)) :  
    Nom(str), Prenom(str), date(str)  
    Programmation(float), Algorithmique(float)  
    Projet(float)  
    calcule et retourne la moyenne(float) des notes  
    '''  
    moy=(eleve["Programmation"]+eleve["Algorithmique"]+eleve["Projet"])/3  
    return moy
```

```
eleve1={"Nom": "Térieur", "Prenom": "Alain", "Date": "01/01/2000",  
"Programmation": 12, "Algorithmique": 10, "Projet": 15}  
print("moyenne de ", eleve1["Prenom"], eleve1["Nom"], " : ", moyenne(eleve1))
```

```
moyenne de Alain Térieur : 12.333333333333334
```

On pourra créer une variable contenant les moyennes de chaque élève, les moyennes de chaque matière et par exemple créer des classements suivant des critères...

La méthode...

Le cahier des charges:

- Stocker les données (nom, prénom, date de naissance, notes) pour les élèves d'une classe
- Calculer et stocker leur moyenne
- Calculer et stocker la moyenne générale de chaque matière
- ...

Choix de la structure de données:

- Pour un élève
- Pour la classe

Implémentation:(ne doit pas modifier le cahier des charges ni le choix de la structure de données)

- Enregistrement des données
- implémentation des fonctions

? EXERCICE 1 :

Implémenter le cahier des charges précédent pour une classe contenant les 3 élèves suivants:

- | | | |
|----------------------|----------------------|---------------------|
| • Nom : Térieur | • Nom : Onette | • Nom : Oma |
| • Prénom : Alain | • Prénom : Camille | • Prénom : Modeste |
| • Date : 01/01/2000 | • Date: 01/07/2004 | • Date: 01/11/2002 |
| • Programmation: 12 | • Programmation: 7 | • Programmation: 13 |
| • Algorithmique : 10 | • Algorithmique : 14 | • Algorithmique : 8 |
| • Projet : 15 | • Projet : 11 | • Projet : 17 |

Des données structurées...

Les données peuvent être déjà structurées dans des fichiers de types csv, json ou xml.

Par exemple, notre classe de 3 élèves pourrait être enregistrée sous la forme d'un fichier csv comme suit:

```
Nom,Prenom,Date,Programmation,Algorithmique,Projet
Térieur,Alain,01/01/00,12,10,15
Onette,Camille,01/07/04,7,14,11
Oma,Modeste,01/11/02,13,8,17
```

On peut alors récupérer ce fichier dans un programme, pour réaliser les calculs souhaités.

On obtiendra comme précédemment une liste de dictionnaires contenant les données des élèves, cependant les notes seront de type **str**...

```
import csv

reader = csv.DictReader(open('eleves.csv', 'r'))
classe = []
for row in reader:
    classe.append(dict(row))

print(classe)
```

```
[{'Nom': 'Térier', 'Prenom': 'Alain', 'Date': '01/01/00', 'Programmation': '12', 'Algorithmique': '10', 'Projet': '15'}, {'Nom': 'Onette', 'Prenom': 'Camille', 'Date': '01/07/04', 'Programmation': '7', 'Algorithmique': '14', 'Projet': '11'}, {'Nom': 'Oma', 'Prenom': 'Modeste', 'Date': '01/11/02', 'Programmation': '13', 'Algorithmique': '8', 'Projet': '17'}]
```

? EXERCICE 2 :

Avec Libre office (ou le bloc note..) créer le fichier eleves.csv , puis réaliser un programme qui affiche les résultats suivants:

```
moyenne de Alain Térier : 12.333333333333334
moyenne de Camille Onette : 10.666666666666666
moyenne de Modeste Oma : 12.666666666666666

moyenne de : programmation: 10.666666666666666
moyenne de : Algorithmique: 10.666666666666666
moyenne de : Projet: 14.333333333333334
```

Vous devriez pouvoir utiliser les fonctions écrites à l'exercice 1 en forçant les notes à être lues comme des nombres `float(...)`

i REMARQUE :

Comme on vient de le voir: Il n'y a pas unicité de la représentation concrète des données élèves, l'important ce sont les données pas leur représentation !