

Première NSI - données en table

5. Travaux dirigés

qkzk

Se tester

1. Manipulation de fichiers CSV

1. Un enregistrement est représenté en Python par :
 - a. une liste
 - b. un dictionnaire
 - c. un p-uplet
 - d. un ensemble
2. Dans un fichier CSV, les attributs sont séparés par :
 - a. des virgules
 - b. des points-virgules
 - c. des tabulations
 - d. des espaces
3. On dispose d'une table de données `Table` représentée par une liste de dictionnaires. En entrant `Table[0]` on obtient :
 - a. une ligne
 - b. une colonne
 - c. une cellule

2. Opérations sur les tables

On dispose de la table `T` représentant les notes d'élèves dans trois matières :

Nom	Maths	Anglais	Informatique
Joe	16	17	18
Zoé	19	15	17
Max	14	19	13

1. Pour sélectionner des colonnes selon un critère donné, laquelle des fonctions `select`, `projection` et `tri` utiliserait-on ?
2. Selon la définition ci-dessous, `select(T, "'17' in ligne.values()")` renvoie une table :

```
def select(table, critere):  
    def test(ligne):  
        return eval(critere)  
    return [ligne for ligne in table if test(ligne)]
```

 - a. vide
 - b. avec une ligne
 - c. avec deux lignes
 - d. avec trois lignes
3. Soit `U` la table suivante :

Nom	Âge	Courriel
Joe	16	joe@info.fr
Zoe	15	zoe@info.fr

Selon la définition ci-dessous `jointure(T, U, 'Nom')` renvoie une table ayant :

```
from copy import deepcopy
def jointure(table1, table2, cle1, cle2=None):
    if cle2 is None:
        cle2 = cle1
    new_table = []
    for line1 in table1:
        for line2 in table2:
            if line1[cle1] == line2[cle2]:
                new_line = deepcopy(line1)
                for cle in line2:
                    if cle != cle2:
                        new_line[cle] = line2[cle]
                new_table.append(new_line)
    return new_table
```

- a. 3 lignes
- b. 2 lignes
- c. 6 colonnes
- d. 5 colonnes
- e. 7 colonnes
- f. 4 colonnes

S'entraîner

3. Déterminer des fonctions basiques

- Déterminer une fonction qui calcule la cardinalité d'une table, c'est-à-dire son nombre de lignes.
- Déterminer une fonction qui renvoie la liste des attributs d'une table.

4.Reconnaître une fonction

Quel est le rôle de la fonction suivante :

```
def mystere(t, cs):
    t_p = []
    for l in t:
        new_l = {}
        for c in l:
            if c in cs:
                new_l[c] = l[c]
        t_p.append(new_l)
    return t_p
```

5. Tester la cohérence d'une table

- Déterminer une fonction `coherence_attributs(table)` qui teste si chaque ligne d'une table a le même ensemble d'attributs.
- Déterminer une fonction `existe_doublons(table, attribut_ref)` qui vérifie si un attribut de référence apparaît deux fois avec la même valeur dans une table.

6. Lier tableur, fichier CSV et dictionnaires

On dispose de la liste de dictionnaires suivante :

```
base_aliens = [
    {'NomAlien': 'Zorglub', 'Sexe': 'M', 'Planete': 'Trantor', 'NoCabine': '1'},
    {'NomAlien': 'Blorx', 'Sexe': 'M', 'Planete': 'Euterpe', 'NoCabine': '2'},
    {'NomAlien': 'Urxiz', 'Sexe': 'M', 'Planete': 'Aurora', 'NoCabine': '3'},
    {'NomAlien': 'Zbleurdite', 'Sexe': 'F', 'Planete': 'Trantor', 'NoCabine': '4'},
    {'NomAlien': 'Darnerane', 'Sexe': 'M', 'Planete': 'Trantor', 'NoCabine': '5'},
    {'NomAlien': 'Mulzo', 'Sexe': 'M', 'Planete': 'Helicon', 'NoCabine': '6'},
    {'NomAlien': 'Zzzzzzzz', 'Sexe': 'F', 'Planete': 'Aurora', 'NoCabine': '7'},
    {'NomAlien': 'Arghh', 'Sexe': 'M', 'Planete': 'Nexon', 'NoCabine': '8'},
    {'NomAlien': 'Joranum', 'Sexe': 'F', 'Planete': 'Euterpe', 'NoCabine': '9'},]
```

1. On travaille avec le tableur LibreOffice Calc de la suite LibreOffice qui produit des fichiers au format odt. Quelle est la première ligne de la feuille de calcul obtenue dans un tableau à partir de cette liste ?
2. Quelle commande lancer pour obtenir le fichier CSV correspondant ?
3. Quelle est la deuxième ligne du fichier CSV correspondant ?
4. Quelle valeur trouve-t-on à la cellule C8 de la feuille de calcul correspondante ?
5. Par quelle commande obtient-on cette valeur à partir de la liste `base_aliens` ?
6. Une erreur de saisie s'est produite : Joranum provient en fait de la planète Aurora. Quelle commande exécuter pour modifier le fichier correspondant du tableur.

7. Ajouter une ligne ou une colonne à une table

On dispose de la table suivante au format CSV dans le répertoire courant sous le nom : `./Groupe1.csv`

Nom	Anglais	Info	Maths
Joe	17	18	16
Zoé	15	17	19
Max	19	13	14

On donne la fonction ci-dessous :

```
import csv
def depuis_csv(fichier):
    lecteur = csv.DictReader(open(fichier), 'r'):
    return [dict(ligne) for ligne in lecteur]
```

1. Comment obtenir la liste de dictionnaires correspondante en utilisant cette fonction ?
2. Ajouter les notes de l'élève Rose qui a eu 17 en mathématiques, 18 en informatique et 19 en anglais.
3. On voudrait ajouter une colonne contenant les moyennes de chaque élève afin d'obtenir le tableau suivant :

Nom	Anglais	Info	Maths	Moyenne
Joe	17	18	16	17.0
Zoé	15	17	19	17.0
Max	19	13	14	15.3
Rose	19	18	17	18.0

On doit renvoyer une nouvelle table qui ne modifie pas la table d'origine. Pour effectuer une copie d'une liste d'objets complexes (ici une liste de dictionnaires), on peut utiliser la fonction `deepcopy` de la bibliothèque `copy`. La fonction à créer pourra donc avoir la structure suivante qu'il s'agit de compléter :

```
from copy import deepcopy
def ajoute_moyenne_ligne(table):
    new_table = deepcopy(table)
    # compléter le code

    return table
```

Pour obtenir l'affichage d'un flottant arrondi à deux chiffres, on peut utiliser la méthode `format`. Par exemple :

```
>>> '{:.2f}'.format(314/100) # .2f indique un flottant arrondi à 2 décimales
'3.14'
```

4. Ajouter une ligne qui contient les moyennes par matière. Vous devez obtenir la table suivante :

Nom	Anglais	Info	Maths	Moyenne
Joe	17	18	16	17.0
Zoé	15	17	19	17.0
Max	19	13	14	15.3
Rose	19	18	17	18.0
Moyenne	17.5	16.5	16.5	16.83

8. Sélectionner, trier, joindre

On dispose de la table `BaseAgents`

NomAgent	VilleAgent
Branno	Terminus
Darell	Terminus
Demerzel	Terminus
Seldon	Uco
Dornick	Terminus
Hardin	Kalgan
Trevize	Terminus
Pelorat	Kalgan
Riose	Terminus

On a aussi la table `BaseGardiens` :

NoCabine	NomAgent
1	Branno
2	Darell
3	Demerzel
4	Seldon
5	Dornick
6	Hardin
7	Trevize
8	Pelorat
9	Riose

1. Renvoyer `BaseTerminus`, une table extraite de `BaseAgents` ne contenant que les lignes dont l'attribut `VilleAgent` vaut "Terminus".
2. Renvoyer `BaseAlpha`, une table dérivée de `BaseAgents` triée selon l'ordre alphabétique du nom des agents.
3. Renvoyer `BaseComplete`, la table contenant le numéro de cabine, la ville et le nom de l'agent.
4. Renvoyer `BaseVille`, la table contenant le numéro de cabine et la ville des agents.
5. Renvoyer `BaseImpair`, la table contenant le nom et la ville des agents ne venant pas de terminus et dont le numéro de cabine est impair.

À noter : On utilisera les fonctions vues en cours. Les tables seront données sous forme de liste de dictionnaires.

Objectif BAC

9. Les hommes en noir

Voici un problème concret qui va permettre d'utiliser tous les outils introduits précédemment et de se rapprocher de ce à quoi peut ressembler la gestion d'une base de donnée.

Le sujet

Chaque jour, l'organisation des "Hommes en noir" (HEN) doit gérer les allées et venues des extraterrestres sur Terre. En arrivant, un extraterrestre est accueilli dans une cabine et surveillé par un gardien. Pour les aider à s'organiser, les HEN disposent de sept tabs de données résumant les informations essentielles sur les extraterrestres et les gardiens.

Table 1. La table `BaseAliens` donne des renseignements sur les extraterrestres

NoCabine	NomAlien	Planete	Sexe
1	Zorglub	Trantor	M
2	Blorx	Euterpe	M
3	Urxiz	Aurora	M
4	Zbleurdite	Trantor	F
5	Darnerane	Trantor	M
6	Mulzo	Helicon	M
7	Zzzzzzzz	Aurora	F
8	Arghh	Nexon	M
9	Joranum	Euterpe	F

Table 2. La table `BaseAgents` donne le nom et la ville des agents

NomAgent	VilleAgent
Branno	Terminus
Darell	Terminus
Demerzel	Terminus
Seldon	Uco
Dornick	Terminus
Hardin	Kalgan
Trevize	Terminus
Pelorat	Kalgan
Riose	Terminus
Palver	Siwenna
Amaryl	Arcturus

Table 3. La table `BaseGardiens` affecte à chaque cabine un gardien.

NoCabine	NomAgent
1	Branno
2	Darell
3	Demerzel
4	Seldon
5	Dornick
6	Hardin
7	Trevize
8	Pelorat
9	Riose

Table 4. la table `BaseMiams` donne l'aliment à servir à chaque extraterrestre.

Aliment	NomAlien
Bortsch	Zorglub
Bortsch	Blorx
Zoumise	Urxiz
Bortsch	Zbleurdite
Schwanstucke	Darnerane

Aliment	NomAlien
Kashpir	Mulzo
Kashpir	Zzzzzzzz
Zoumise	Arghh
Bortsch	Joranum

Table 5. La table `BaseCabines` précise dans quelle allée se trouve chaque cabine.

NoAllee	NoCabine
1	1
1	2
2	3
1	4
2	5
2	6
2	7
1	8
1	9

Table 6. La table `BaseResponsables` précise l'agent responsable de chaque allée.

NoAllee	NomAgent
1	Seldon
2	Pelorat

Table 7. La table `BaseVilles` précise la planète sur laquelle se trouve chaque ville.

Planete	Ville
Trantor	Terminus
Euterpe	Arcturus
Helicon	Kalgan
Euterpe	Hesperos
Gaia	Siwenna

À noter

Dans tout l'exercice on pourra utiliser les fonctions introduites dans le cours.

1. Mettre en forme

Comment entrer ces tables afin de pouvoir utiliser les outils mis au point en cours ?

2. Extraction des données

- Comment obtenir l'ensemble des gardiens ?
- Comment obtenir l'ensemble des villes dont sont originaires les gardiens ?
- Comment obtenir l'ensemble des triplets (numéro de cabine, extraterrestre, gardien) pour chaque cabine ?
- Comment obtenir l'ensemble de tous les extraterrestres de l'allée 2 ?
- Comment obtenir la liste des extraterrestres dont les gardiens sont originaires de la planète Trantor ?
- Comment obtenir l'ensemble des gardiens des extraterrestres féminins qui mangent du Bortsch ?

3. Tests

- Existe-t-il un aliment qui commence par la même lettre que le nom de l'extraterrestre qui le mange ?
- Est-ce que tous les extraterrestres qui ont un 'x' dans leur nom ont un gardien qui vient de terminus ?

La feuille de route

1. Mettre en forme des données

Il y a plusieurs possibilités : ouvrir un logiciel comme libreOffice ou Excel et recopier ces données puis les exporter au format CSV. On peut également, dans un éditeur de texte simple, créer des fichiers CSV. On peut aussi rentrer directement les tables comme liste de dictionnaires mais c'est un peu fastidieux.

2. Extraire des données structurées en table

- a. Il faut parcourir la table `baseGardien` et ne garder que les noms des agents. L'utilisation d'un ensemble défini par compréhension est appropriée. On peut également penser à effectuer une projection de `BaseGardiens` en ne gardant que les noms.
- b. Attention ! Tous les agents ne sont pas des gardiens. Il faut d'abord joindre les deux tables `BaseGardiens` et `BaseAgents` puis ne retenir que les villes dans un ensemble défini par compréhension.
- c. Les tables `BaseGardien` et `BaseAliens` ont en commun le numéro des cabines. Il s'agit donc de joindre ces deux tables.
- d. Il faut cette fois joindre les tables `BaseAliens` et `BaseCabines` et sélectionner les lignes contenant l'allée 2. On peut utiliser une condition `if` ou bien la fonction `select` vue en cours.
- e. Ici le nombre de tables impliquées dans la requête augmente : il faut lier `BaseAliens` et `BaseGardiens` par le numéro de cabine, puis relier cette nouvelle table à `BaseAgents` par le nom de l'agent pour obtenir la ville d'origine du gardien et enfin joindre cette dernière table à `BaseVilles` pour obtenir la planète d'origine.

À noter : *Dans la dernière jointure les noms des attributs désignant les villes sont différents.*

- f. Ici, il s'agit de trois jointures, mais cette fois il y a deux conditions à vérifier : l'extraterrestre est féminin ET l'aliment est le bortsch.

3. Tester des situations.

- a. Il faut à nouveau joindre `BaseMiams`, `BaseAliens` et `BaseGardiens`. La première lettre d'un mot s'obtient avec `mot[0]`

Pour le test en lui même, on peut former un ensemble de tests et vérifier que `True` appartient à cet ensemble ou bien construire une boucle `while`.
- b. Cette fois on doit vérifier que notre condition est toujours vraie, donc que `False` n'appartient pas à notre ensemble de tests. On reprend la jointure effectuée à la question **2.e** entre `BaseAliens`, `BaseGardiens` et `BaseAgents`.