

Importer une table et rechercher dans une table

■ Introduction

Organiser des données sous forme de tableaux est très répandu et très ancien car on retrouve des tables de données dans les livres de comptes de l'Egypte ancienne.

On peut organiser des données dans un tableau pour présenter des informations croisées, faire ses comptes, etc. Le bulletin d'un élève est une table qui indique pour chaque matière, la moyenne de l'élève, la moyenne de la classe, l'appréciation du professeur. La liste de présence des élèves est également représentée sous forme d'une table indiquant pour chaque élève s'il est présent ou absent ce jour.

En informatique, les tables de données sont rapidement devenues le principal moyen de stocker des données. L'analyse de ces données est utilisée dans de nombreux domaines : calcul scientifique, intelligence artificielle, programmation Web, bio-informatique, informatique financière, etc. De nombreuses données en tables sont disponibles sur les sites de données ouvertes.

Exemple

Voici un exemple de table avec des informations sur des élèves :

prénom	jour	mois	année	sexe	groupe	projet
Loona	29	11	2005	F	1	être heureuse
Quentin	28	5	2005	G	2	manger une glace
Sania	30	7	2005	F	1	gagner au loto
Yanis	14	4	2005	G	1	devenir quelqu'un de célèbre
Nolan	16	4	2005	G	2	avoir une bonne note au prochain devoir
Mathis	12	1	2005	G	1	devenir astronaute
Hippolyte	20	10	2005	G	2	marcher sur la lune
Martial	22	4	2005	G	2	dormir plus longtemps le matin
Arthur	10	8	2005	G	2	payer les autres
Aïris	24	3	2005	F	2	apprendre à piloter une Formule 1

Pour exploiter ces données avec un ordinateur il est nécessaire de les mémoriser dans un fichier.

■ Enregistrement et chargement des données

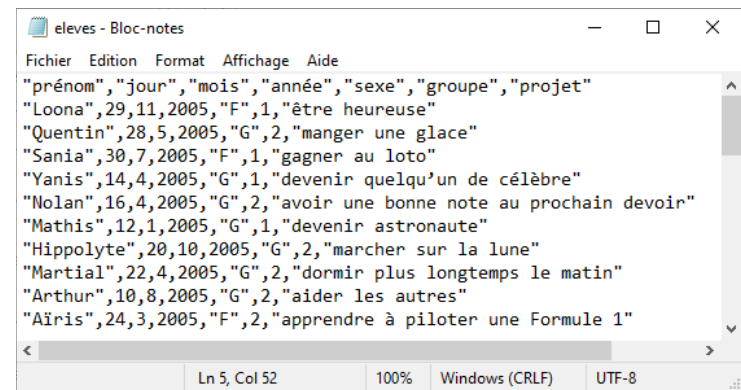
Enregistrer les données dans un fichier

Il existe différents formats permettant de mémoriser les informations d'une table dans un fichier. Un des plus utilisés est le format CSV (pour l'anglais *comma-separated values*, soit "données séparées par des virgules") qui est un fichier texte dans lequel :

- chaque ligne correspond à une ligne de la table ;
- chaque ligne est séparée en champs par une virgule (",") ;
- toutes les lignes du fichier ont le même nombre de champs ;
- on peut utiliser les guillemets droits (") pour délimiter le contenu des champs.

Ainsi, la table précédente peut être mémorisée dans un fichier CSV appelé **eleves.csv** dont voici le contenu :

```
"prénom","jour","mois","année","sexe","groupe","projet"
"Loona",29,11,2005,"F",1,"être heureuse"
"Quentin",28,5,2005,"G",2,"manger une glace"
"Sania",30,7,2005,"F",1,"gagner au loto"
"Yanis",14,4,2005,"G",1,"devenir quelqu'un de célèbre"
"Nolan",16,4,2005,"G",2,"avoir une bonne note au prochain devoir"
"Mathis",12,1,2005,"G",1,"devenir astronaute"
"Hippolyte",20,10,2005,"G",2,"marcher sur la lune"
"Martial",22,4,2005,"G",2,"dormir plus longtemps le matin"
"Arthur",10,8,2005,"G",2,"aider les autres"
"Aïris",24,3,2005,"F",2,"apprendre à piloter une Formule 1"
```



Fichier CSV ouvert avec un éditeur de texte.

Bien qu'étant un simple fichier texte, un fichier CSV peut être importé et exporté par un tableur qui permet d'en visualiser le contenu sous la forme d'un tableau.

Remarques :

- En France, le caractère de séparation des champs est généralement le point-virgule (";") car la virgule est utilisée pour écrire des nombres décimaux. Il faut donc être vigilant sur le caractère de séparation utilisé.
- Le format CSV est apparu en 1972, soit bien avant l'avènement des ordinateurs personnels.

eleves.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Feuille Données Outils Fenêtre Aide

Libération Sans 10

	A	B	C	D	E	F	G
1	prénom	jour	mois	année	sexe	groupe	projet
2	Loona	29	11	2005	F		1 être heureuse
3	Quentin	28	5	2005	G		2 manger une glace
4	Sanja	30	7	2005	F		1 gagner au loto
5	Yanis	14	4	2005	G		1 devenir quelqu'un de célèbre
6	Nolan	16	4	2005	G		2 avoir une bonne note au prochain devoir
7	Mathis	12	1	2005	G		1 devenir astronaute
8	Hippolyte	20	10	2005	G		2 marcher sur la lune
9	Martial	22	4	2005	G		2 dormir plus longtemps le matin
10	Arthur	10	8	2005	G		2 aider les autres
11	Airis	24	3	2005	F		2 apprendre à piloter une Formule 1

Fichier CSV ouvert avec un tableur.

On a utilisé le paramètre `'r'` pour indiquer une ouverture en mode *lecture* (*read* en anglais).

Il faut ensuite lire le fichier pour charger son contenu dans Python. On pourrait pour cela utiliser la méthode `read` qui renvoie le contenu du fichier sous la forme d'une unique chaîne de caractères.

Pour exploiter et manipuler des tables de données mémorisées dans un fichier CSV, il faut commencer par *charger les données*. On pourra ensuite utiliser toute la puissance d'un langage de programmation pour les exploiter.

Charger les données dans Python

Pour importer les données d'un fichier CSV dans Python il faut commencer par ouvrir le fichier avec Python. On utilise pour cela la fonction prédéfinie `open`.

```
In [1]: # on ouvre le fichier eleves.csv dans un objet appelé 'fichier'
fichier = open('eleves.csv', 'r', encoding = 'UTF-8')
```

```
In [2]: texte = fichier.read() # on lit le contenu sous la forme
print(texte) # on affiche la chaîne de caractères 'texte'

# on ferme le flux de lecture
fichier.close()
```

```
"prénom","jour","mois","année","sexe","groupe","projet"
"Loona",29,11,2005,"F",1,"être heureuse"
"Quentin",28,5,2005,"G",2,"manger une glace"
"Sania",30,7,2005,"F",1,"gagner au loto"
"Yanis",14,4,2005,"G",1,"devenir quelqu'un de célèbre"
"Nolan",16,4,2005,"G",2,"avoir une bonne note au prochain d
"Mathis",12,1,2005,"G",1,"devenir astronaute"
"Hippolyte",20,10,2005,"G",2,"marcher sur la lune"
"Martial",22,4,2005,"G",2,"dormir plus longtemps le matin"
"Arthur",10,8,2005,"G",2,"aider les autres"
"Aïris",24,3,2005,"F",2,"apprendre à piloter une Formule 1"
"Léo",28,7,2005,"G",2,"devenir président"
"Romain",23,1,2005,"G",2,"créer un jeu vidéo"
"Tanguy",9,6,2005,"G",1,"gérer sa propre entreprise"
"Nolwenn",24,11,2005,"F",2,"jouer dans le prochain James Boi
"Célian",17,11,2005,"G",1,"gagner le Tour de France"
"Rayane",23,1,2005,"G",1,"devenir champion du monde de bras
"Ancelin",15,1,2007,"G",2,"devenir youtubeur"
"Ewen",16,5,2005,"G",2,"avoir son émission de TV"
"Nolan",16,5,2005,"G",1,"écrire un livre"
"Valentin",8,8,2005,"G",1,"inventer la machine à voyager da
"Mathis",14,1,2005,"G",1,"devenir une rockstar"
"Jade",14,3,2005,"F",2,"développer une application"
"Charles",23,1,2005,"G",2,"ouvrir un restaurant"
"Sylliane",7,6,2005,"G",1,"aller à Poudlard"
"Clément",6,12,2005,"G",1,"monter le tapis rouge à Cannes"
"Kilian",3,11,2005,"G",1,"gagner un Oscar"
"Haroun",27,5,2004,"G",1,"devenir un joueur professionnel de
"Hugo",10,1,2005,"G",2,"sortir un film"
"Lois",17,11,2005,"G",2,"créer un objet révolutionnaire"
"Agathe",15,7,2005,"F",2,"devenir championne du monde d'éche
"Baptiste",10,12,2005,"G",1,"manger des frites tous les jou
"Sébastien",14,1,2005,"G",1,"devenir champion olympique"
"Nicolas",24,4,2006,"G",2,"participer à The Voice"
"Maé",18,8,2005,"F",2,"enregistrer un album"
"Nolan",25,7,2005,"G",1,"acheter une Ferrari"
"Timothé",11,1,2006,"G",1,"participer au trophée Jules Vern
"Inès",13,12,2005,"F",1,"battre le record du monde du marath
"Kylian",13,3,2005,"G",2,"battre Usain Bolt"
```

Cependant, ce n'est pas ainsi que l'on va procéder car exploiter cette grande chaîne de caractères n'est pas pratique. En effet, il sera beaucoup simple et efficace de manipuler des *types construits* comme les tableaux et surtout les dictionnaires.

Ainsi, on va plutôt lire le contenu du fichier pour construire un `tableau` dans lequel chaque élément correspond à une ligne de la table. Il est possible de représenter chaque ligne de la table (donc chaque élève) par un `tableau` ou par un `dictionnaire`.

Le chargement dans un tableau de dictionnaire est préférable car il est alors facile d'accéder aux données par leurs clés (voir le notebook de cours pour le chargement dans un tableau de tableaux).

Pour cela, la bibliothèque standard de Python contient un module `csv` qui contient des fonctions très pratiques pour lire et écrire des fichiers CSV. La mémorisation dans un tableau devenant alors très rapide. Voici comment on procède.

Chargement dans un tableau de dictionnaires

La méthode `DictReader` permet de lire la table ligne à ligne et renvoie successivement pour chaque ligne un dictionnaire des champs de cette ligne. La première ligne du fichier a servi à obtenir les clés utilisées pour chaque enregistrement.

On peut alors construire par compréhension le tableau `eleves` qui mémorise toutes les données de la table, chaque élément de ce tableau étant un dictionnaire correspondant à une ligne de la table de données.

```
In [3]: import csv # pour utiliser le module csv

# Ouverture du fichier
fichier = open('eleves.csv', 'r', encoding = 'UTF-8')

# Lecture de son contenu avec la méthode DictReader qui r
table = csv.DictReader(fichier, delimiter=',') # on préc

# Construction par compréhension du tableau représentant
eleves = [dict(ligne) for ligne in table] # chaque éléme

# Fermeture du flux de lecture
fichier.close()

eleves # pour voir le contenu du tableau 'eleves'
```

```
Out[3]: [{ 'prénom': 'Loona',
  'jour': '29',
  'mois': '11',
  'année': '2005',
  'sexe': 'F',
  'groupe': '1',
  'projet': 'être heureuse'},
{ 'prénom': 'Quentin',
  'jour': '28',
  'mois': '5',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'manger une glace'},
{ 'prénom': 'Sania',
  'jour': '30',
  'mois': '7',
  'année': '2005',
  'sexe': 'F',
  'groupe': '1',
  'projet': 'gagner au loto'},
{ 'prénom': 'Yanis',
  'jour': '14',
  'mois': '4',
  'année': '2005',
  'sexe': 'G',
  'groupe': '1',
  'projet': 'devenir quelqu'un de célèbre'},
{ 'prénom': 'Nolan',
  'jour': '16',
  'mois': '4',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'avoir une bonne note au prochain devoir'},
{ 'prénom': 'Mathis',
  'jour': '12',
  'mois': '1',
  'année': '2005',
  'sexe': 'G',
  'groupe': '1',
  'projet': 'devenir astronaute'},
```

```
{ 'prénom': 'Hippolyte',
  'jour': '20',
  'mois': '10',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'marcher sur la lune'},
{ 'prénom': 'Martial',
  'jour': '22',
  'mois': '4',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'dormir plus longtemps le matin'},
{ 'prénom': 'Arthur',
  'jour': '10',
  'mois': '8',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'aider les autres'},
{ 'prénom': 'Aïris',
  'jour': '24',
  'mois': '3',
  'année': '2005',
  'sexe': 'F',
  'groupe': '2',
  'projet': 'apprendre à piloter une Formule 1'},
{ 'prénom': 'Léo',
  'jour': '28',
  'mois': '7',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'devenir président'},
{ 'prénom': 'Romain',
  'jour': '23',
  'mois': '1',
  'année': '2005',
  'sexe': 'G',
  'groupe': '2',
  'projet': 'créer un jeu vidéo'},
```

```
{'prénom': 'Tanguy',
 'jour': '9',
 'mois': '6',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'gérer sa propre entreprise'},
{'prénom': 'Nolwenn',
 'jour': '24',
 'mois': '11',
 'année': '2005',
 'sexe': 'F',
 'groupe': '2',
 'projet': 'jouer dans le prochain James Bond'},
{'prénom': 'Célian',
 'jour': '17',
 'mois': '11',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'gagner le Tour de France'},
{'prénom': 'Rayane',
 'jour': '23',
 'mois': '1',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'devenir champion du monde de bras de fer'},
{'prénom': 'Ancelin',
 'jour': '15',
 'mois': '1',
 'année': '2007',
 'sexe': 'G',
 'groupe': '2',
 'projet': 'devenir youtubeur'},
{'prénom': 'Ewen',
 'jour': '16',
 'mois': '5',
 'année': '2005',
 'sexe': 'G',
 'groupe': '2',
 'projet': 'avoir son émission de TV'},
```

```
{'prénom': 'Nolan',
 'jour': '16',
 'mois': '5',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'écrire un livre'},
{'prénom': 'Valentin',
 'jour': '8',
 'mois': '8',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'inventer la machine à voyager dans le temps'},
{'prénom': 'Mathis',
 'jour': '14',
 'mois': '1',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'devenir une rockstar'},
{'prénom': 'Jade',
 'jour': '14',
 'mois': '3',
 'année': '2005',
 'sexe': 'F',
 'groupe': '2',
 'projet': 'développer une application'},
{'prénom': 'Charles',
 'jour': '23',
 'mois': '1',
 'année': '2005',
 'sexe': 'G',
 'groupe': '2',
 'projet': 'ouvrir un restaurant'},
{'prénom': 'Sylliane',
 'jour': '7',
 'mois': '6',
 'année': '2005',
 'sexe': 'G',
 'groupe': '1',
 'projet': 'aller à Poudlard'},
```

```
{'prénom': 'Clément',  
  'jour': '6',  
  'mois': '12',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'monter le tapis rouge à Cannes'},  
{'prénom': 'Kilian',  
  'jour': '3',  
  'mois': '11',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'gagner un Oscar'},  
{'prénom': 'Haroun',  
  'jour': '27',  
  'mois': '5',  
  'année': '2004',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'devenir un joueur professionnel de billard'},  
{'prénom': 'Hugo',  
  'jour': '10',  
  'mois': '1',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '2',  
  'projet': 'sortir un film'},  
{'prénom': 'Lois',  
  'jour': '17',  
  'mois': '11',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '2',  
  'projet': 'créer un objet révolutionnaire'},  
{'prénom': 'Agathe',  
  'jour': '15',  
  'mois': '7',  
  'année': '2005',  
  'sexe': 'F',  
  'groupe': '2',  
  'projet': "devenir championne du monde d'échecs"},
```

```
{'prénom': 'Baptiste',  
  'jour': '10',  
  'mois': '12',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'manger des frites tous les jours à la cantine'},  
{'prénom': 'Sébastien',  
  'jour': '14',  
  'mois': '1',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'devenir champion olympique'},  
{'prénom': 'Nicolas',  
  'jour': '24',  
  'mois': '4',  
  'année': '2006',  
  'sexe': 'G',  
  'groupe': '2',  
  'projet': 'participer à The Voice'},  
{'prénom': 'Maé',  
  'jour': '18',  
  'mois': '8',  
  'année': '2005',  
  'sexe': 'F',  
  'groupe': '2',  
  'projet': 'enregistrer un album'},  
{'prénom': 'Nolan',  
  'jour': '25',  
  'mois': '7',  
  'année': '2005',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'acheter une Ferrari'},  
{'prénom': 'Timothé',  
  'jour': '11',  
  'mois': '1',  
  'année': '2006',  
  'sexe': 'G',  
  'groupe': '1',  
  'projet': 'participer au trophée Jules Verne'},
```



```
{'prénom': 'Inès',
 'jour': '13',
 'mois': '12',
 'année': '2005',
 'sexe': 'F',
 'groupe': '1',
 'projet': 'battre le record du monde du marathon'},
{'prénom': 'Kylia',
 'jour': '13',
 'mois': '12'}
```

Chaque ligne est représentée par un dictionnaire et on peut accéder à n'importe quelle donnée. Par exemple, comme la clé correspondant au sexe est `sexe`, on peut accéder au sexe du deuxième élève avec l'instruction

```
In [4]: eleves[1]['sexe']
```

```
Out[4]: 'G'
```

On peut aussi accéder au projet de Sania avec l'instruction

```
In [5]: eleves[2]['projet']
```

```
Out[5]: 'gagner au loto'
```

Même en mémorisant chaque ligne de la table dans un dictionnaire, tout n'est pas encore parfait. En effet, pour trouver le projet de Justine, il faut savoir que la ligne la concernant a pour index 10 dans le tableau. Je vous laisse imaginer ce que cela peut donner pour trouver un index dans des fichiers de plusieurs dizaines de milliers de lignes... Heureusement, il est possible de chercher plus efficacement des données dans une table, c'est ce qui est expliqué ci-après.

■ Rechercher un élément dans une table

Une fois les données chargées dans une table, il est possible de travailler sur ces données en utilisant les opérations de manipulation sur les tableaux et les dictionnaires. En particulier, on va pouvoir effectuer une recherche en *parcourant le tableau* ligne par ligne, c'est-à-dire dictionnaire par dictionnaire.

Présence d'un élément

On cherche ici à détecter la présence d'un élément dans une table (un élève dans notre cas). Nous allons utiliser des techniques de recherche dans un tableau comme celles dans le chapitre "Algorithmes de parcours séquentiel d'un tableau" (Thème 7, Chapitre 1).

Pour rappel, voici la fonction `appartient` qui renvoie `True` ou `False` selon la présence ou l'absence d'un élément `v` dans un tableau `T`.

```
def appartient(v, T):
    """Renvoie True si v appartient au tableau T, False sinon."""
    for e in T:
        if e == v:
            return True
    return False
```

On va légèrement adapter cette fonction pour chercher un élève désigné par son prénom dans notre table.

```
In [6]: def appartient(p, eleves):
        '''
        Renvoie True si le prénom p est dans le tableau eleve
        p est un chaîne de caractères et eleves est un tablea
        '''
        for eleve in eleves:
            if eleve['prénom'] == p:
                return True
        return False
```

Explications : on passe en revue tous les éléments du tableau (chaque dictionnaire `eleve`) et on teste si l'un d'entre eux possède un attribut `'prénom'` égal à `p`, autrement dit si un des élèves a le prénom `p`.

On peut alors appeler cette fonction pour tester la présence d'un élève.

Pensez à exécuter à nouveau, si besoin, la première cellule du paragraphe précédent intitulé *Chargement dans un tableau de dictionnaires* qui charge les données de notre table dans le tableau `eleves`.

```
In [7]: appartient('Charles', eleves)
```

```
Out[7]: True
```

```
In [8]: appartient('Marie', eleves)
```

```
Out[8]: False
```

Récupérer une donnée simple selon un critère

On peut adapter la fonction précédente pour qu'elle renvoie le projet de l'élève dont le prénom est entré en paramètre.

```
In [9]: def projet_de(p, eleves):
        '''
        Renvoie le projet de l'élève prénommé p dans le table
        p est un chaîne de caractères et eleves est un tablea
        '''
        for eleve in eleves:
            if eleve['prénom'] == p: # si un élève a le pré
                return eleve['projet'] # on renvoie le projet
        return None
```

```
In [10]: projet_de('Sania', eleves)
```

```
Out[10]: 'gagner au loto'
```

Vous remarquerez qu'avec cette façon de faire, il n'est plus nécessaire de connaître la position de Sania (son indice) dans la table, son prénom suffit désormais.

Récupérer une donnée simple selon plusieurs critères

Il est possible de préciser la recherche en combinant les conditions sur plusieurs attributs. On peut par exemple, rechercher le prénom d'un élève né l'année `a` et le mois `m`.

```
In [11]: def nom_eleve(a, m, eleves):
        ''' a et m sont des entiers et eleves est un tableau
        for eleve in eleves:
            if int(eleve['année']) == a and int(eleve['mois'])
                return eleve['prénom']
        return None
```

```
In [12]: nom_eleve(2005, 4, eleves)
```

```
Out[12]: 'Yanis'
```

Remarques :

- On a pris soin de convertir les valeurs des attributs `'année'` et `'mois'` en type `int` pour comparer des variables de même type.
- Dans notre table, il y avait trois élèves nés en avril 2005 mais seul celui dont la ligne apparaît en premier a été renvoyé.

Les fonctions proposées parcourent la table ligne par ligne jusqu'à trouver la donnée correspondant aux critères de recherche, et la renvoie. Cependant, plusieurs lignes de la table (ici plusieurs élèves) peuvent correspondre à la recherche effectuée et on aimerait pouvoir rechercher dans *toutes* ces lignes (par exemple, renvoyer les prénoms de *tous* les élèves nés une certaine année et pas seulement le premier rencontré). Nous allons voir comment procéder dans le paragraphe suivant.

■ Sélection de lignes

Toutes les opérations précédentes permettent de produire des résultats simples : afficher le projet d'un élève, ou son sexe, ou son prénom...

Or, dans le domaine du traitement de données, il est très souvent nécessaire d'extraire différentes données d'une table pour en construire une nouvelle. Par exemple, extraire tous les élèves qui sont nés un mois donné afin de leur souhaiter un bon anniversaire.

Pour cela, il suffit de sélectionner les lignes vérifiant une certaine condition pour produire une nouvelle table. Cette condition peut combiner plusieurs critères. Pour créer cette nouvelle table, on peut avantageusement la construire par compréhension.

Syntaxe

On utilise la syntaxe classique d'une construction par compréhension mais en ajoutant une condition avec un `if` :

```
selection = [ligne for ligne in table if criteres]
```

Ainsi, on crée un nouveau tableau `selection` qui contient uniquement les lignes du tableau `table` de départ qui vérifient la condition `criteres`, et pas les autres.

Par exemple, on peut créer un nouveau tableau `nes_en_aout` qui contient tous les (dictionnaires des) élèves de la table `eleves` qui sont nés en août.

```
In [13]: nes_en_aout = [eleve for eleve in eleves if int(eleve['mo  
nes_en_aout # pour voir le tableau
```

```
Out[13]: [{ 'prénom': 'Arthur',  
            'jour': '10',  
            'mois': '8',  
            'année': '2005',  
            'sexe': 'G',  
            'groupe': '2',  
            'projet': 'aider les autres'},  
          { 'prénom': 'Valentin',  
            'jour': '8',  
            'mois': '8',  
            'année': '2005',  
            'sexe': 'G',  
            'groupe': '1',  
            'projet': 'inventer la machine à voyager dans le temps'},  
          { 'prénom': 'Maé',  
            'jour': '18',  
            'mois': '8',  
            'année': '2005',  
            'sexe': 'F',  
            'groupe': '2',  
            'projet': 'enregistrer un album'}]
```

Ce tableau `nes_en_aout` représente la nouvelle table de données suivante.

prénom	jour	mois	année	sexe	projet
Arthur	10	8	2005	G	aider les autres
Valentin	8	8	2005	G	inventer la machine à voyager dans le temps
Maé	18	8	2005	F	enregistrer un album

On a sélectionné les lignes *complètes* correspondant à certains critères. Cela signifie que toutes les colonnes ont été ajoutées à la table créée. On peut aussi très facilement construire un nouveau tableau ne contenant que certaines colonnes de la table de départ, ce que nous allons voir maintenant.

■ Sélectionner des lignes et des colonnes

Projection sur une ou plusieurs colonnes

Si on ne souhaite sélectionner que certaines colonnes de la table de départ, on dit que l'on fait une *projection* sur ces colonnes .

On peut construire un nouveau tableau en faisant une projection sur une seule colonne, par exemple les prénoms.

```
In [14]: t0 = [eleve['prénom'] for eleve in eleves]
t0
```

```
Out[14]: ['Loona',
'Quentin',
'Sania',
'Yanis',
'Nolan',
'Mathis',
'Hippolyte',
'Martial',
'Arthur',
'Aïris',
'Léo',
'Romain',
'Tanguy',
'Nolwenn',
'Célian',
'Rayane',
'Ancelein',
'Ewen',
'Nolan',
'Valentin',
'Mathis',
'Jade',
'Charles',
'Sylliane',
'Clément',
'Kilian',
'Haroun',
'Hugo',
'Lois',
'Agathe',
'Baptiste',
'Sébastien',
'Nicolas',
'Maé',
'Nolan',
'Timothé',
'Inès',
'Kylian']
```

On peut aussi projeter sur plusieurs colonnes, en reconstruisant des dictionnaires ne contenant que les attributs souhaités, par exemple les prénoms et les projets :

```
In [15]: t1 = [{'prénom': eleve['prénom'], 'projet': eleve['projet']}
t1 # pour voir le contenu
```

```
Out[15]: [{'prénom': 'Loona', 'projet': 'être heureuse'},
{'prénom': 'Quentin', 'projet': 'manger une glace'},
{'prénom': 'Sania', 'projet': 'gagner au loto'},
{'prénom': 'Yanis', 'projet': 'devenir quelqu'un de célèbre'},
{'prénom': 'Nolan', 'projet': 'avoir une bonne note au prochain'},
{'prénom': 'Mathis', 'projet': 'devenir astronaute'},
{'prénom': 'Hippolyte', 'projet': 'marcher sur la lune'},
{'prénom': 'Martial', 'projet': 'dormir plus longtemps le week-end'},
{'prénom': 'Arthur', 'projet': 'aider les autres'},
{'prénom': 'Aïris', 'projet': 'apprendre à piloter une Formule 1'},
{'prénom': 'Léo', 'projet': 'devenir président'},
{'prénom': 'Romain', 'projet': 'créer un jeu vidéo'},
{'prénom': 'Tanguy', 'projet': 'gérer sa propre entreprise'},
{'prénom': 'Nolwenn', 'projet': 'jouer dans le prochain Jarvi'},
{'prénom': 'Célian', 'projet': 'gagner le Tour de France'},
{'prénom': 'Rayane', 'projet': 'devenir champion du monde de cyclisme'},
{'prénom': 'Ancelin', 'projet': 'devenir youtubeur'},
{'prénom': 'Ewen', 'projet': 'avoir son émission de TV'},
{'prénom': 'Nolan', 'projet': 'écrire un livre'},
{'prénom': 'Valentin',
 'projet': 'inventer la machine à voyager dans le temps'},
{'prénom': 'Mathis', 'projet': 'devenir une rockstar'},
{'prénom': 'Jade', 'projet': 'développer une application'},
{'prénom': 'Charles', 'projet': 'ouvrir un restaurant'},
{'prénom': 'Sylliane', 'projet': 'aller à Poudlard'},
{'prénom': 'Clément', 'projet': 'monter le tapis rouge à Cannes'},
{'prénom': 'Kilian', 'projet': 'gagner un Oscar'},
{'prénom': 'Haroun', 'projet': 'devenir un joueur professionnel de football'},
{'prénom': 'Hugo', 'projet': 'sortir un film'},
{'prénom': 'Lois', 'projet': 'créer un objet révolutionnaire'},
{'prénom': 'Agathe', 'projet': 'devenir championne du monde de natation'},
{'prénom': 'Baptiste',
 'projet': 'manger des frites tous les jours à la cantine'},
{'prénom': 'Sébastien', 'projet': 'devenir champion olympique de ski'},
{'prénom': 'Nicolas', 'projet': 'participer à The Voice'},
{'prénom': 'Maé', 'projet': 'enregistrer un album'},
{'prénom': 'Nolan', 'projet': 'acheter une Ferrari'},
{'prénom': 'Timothé', 'projet': 'participer au trophée Jules et Jim'},
{'prénom': 'Inès', 'projet': 'battre le record du monde du 100m'},
{'prénom': 'Kylian', 'projet': 'battre Usain Bolt'}]
```

Combiner les sélections sur les lignes et les colonnes

Il est bien sûr possible de combiner une sélection sur les colonnes avec une sélection sur les lignes. Ainsi, l'instruction

```
In [16]: t2 = [eleve['prénom'] for eleve in eleves if eleve['sexe']
t2 # pour voir le contenu
```

```
Out[16]: ['Quentin',
'Yanis',
'Nolan',
'Mathis',
'Hippolyte',
'Martial',
'Arthur',
'Léo',
'Romain',
'Tanguy',
'Célian',
'Rayane',
'Ancein',
'Ewen',
'Nolan',
'Valentin',
'Mathis',
'Charles',
'Sylliane',
'Clément',
'Kilian',
'Haroun',
'Hugo',
'Lois',
'Baptiste',
'Sébastien',
'Nicolas',
'Nolan',
'Timothé',
'Kylian']
```

permet de créer un tableau contenant les prénoms des garçons. On a donc sélectionné la première colonne contenant les prénoms des élèves (avec `eleve[prénom]`) mais seulement pour les lignes correspondant à des garçons (avec le critère `if eleve['sexe'] == 'G'`).

On peut bien sûr aussi sélectionner plusieurs colonnes tout en sélectionnant certaines lignes. Par exemple, on peut sélectionner les attributs `'prénom'` et `'projet'` mais uniquement des filles.

```
In [17]: t3 = [{ 'prénom': eleve[ 'prénom'], 'projet': eleve[ 'projet']
t3 # pour voir le contenu
```

```
Out[17]: [{ 'prénom': 'Loona', 'projet': 'être heureuse'},
{ 'prénom': 'Sania', 'projet': 'gagner au loto'},
{ 'prénom': 'Aïris', 'projet': 'apprendre à piloter une Forx',
{ 'prénom': 'Nolwenn', 'projet': 'jouer dans le prochain Jar',
{ 'prénom': 'Jade', 'projet': 'développer une application'},
{ 'prénom': 'Agathe', 'projet': "devenir championne du monde",
{ 'prénom': 'Maé', 'projet': 'enregistrer un album'},
{ 'prénom': 'Inès', 'projet': 'battre le record du monde du
```

■ Bilan

- Les tables de données peuvent être mémorisées dans des fichiers et notamment dans des fichiers au format CSV.
- Pour traiter ces données (avec Python, ou avec un autre langage) il est nécessaire de *lire le fichier* et de *charger les données*. On a vu que le module **CSV** permettait facilement de mémoriser les données d'une table dans un tableau de dictionnaires.
- On peut ensuite utiliser toutes les fonctionnalités du langage de programmation pour le traitement des données. En particulier, on peut y *rechercher des informations*, *sélectionner les colonnes* en fonction des attributs qui nous intéressent mais aussi *sélectionner des lignes* en fonction de certains *critères*.
- Ces traitements permettent de créer de nouvelles tables que l'on peut aussi *écrire* dans un fichier CSV pour garder une trace pérenne des informations calculées (fait en activité).
- Nous verrons au cours d'un TP l'intérêt de faire un *prétraitement* adapté des données pour répondre aux problématiques suivantes :
 - les données chargées dans un tableau le sont sous forme de chaînes de caractères, mêmes les nombres. Il peut donc être intéressant de les convertir une fois pour toutes au départ et éviter les conversions de type dans tous les traitements.
 - pour les données ouvertes obtenues sur des sites spécialisés, il convient d'être vigilant en cas de données manquantes ou de données particulières. On s'assurera donc que les données sont *cohérentes* avant de commencer à les traiter.
- Tout ce qui est fait sur le traitement des données en classe de Première est un avant-goût des bases de données qui seront étudiées en classe de Terminale.