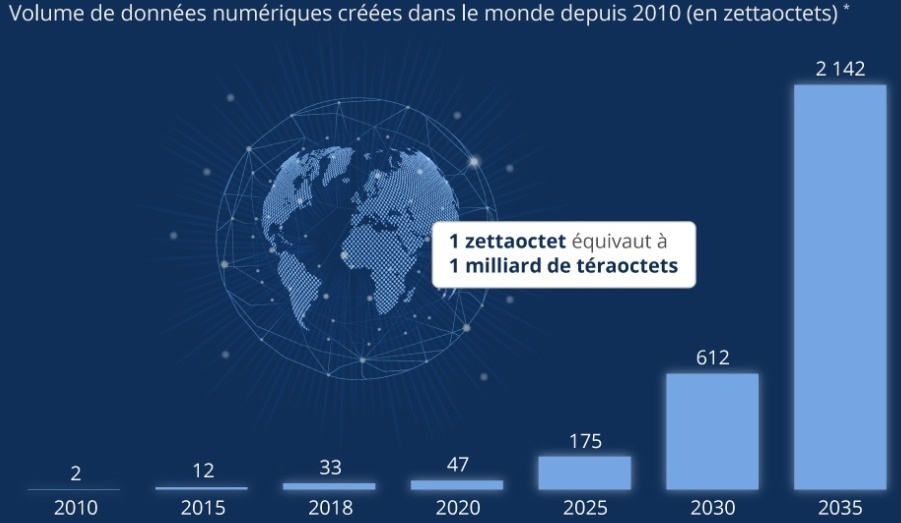
|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 5h | **Traitement de données en tables** |  |
| **Objectif** : traiter les données en tables | | |
| **Matériel**: Python | | |

**Les données**

Une donnée est une information qui peut être personnelle (lieu de naissance) ou pas (nombre d'habitant à Angers).

Aujourd’hui presque toutes les données utilisées dans le monde sont numériques (créées ou collectées directement sur un appareil informatique) ou en voie de numérisation (numérisation d’archives par exemple).

Les quantités de données collectées sont gigantesques et en croissance exponentielle.

Les entreprises comme Google font leur richesse grâce aux données que nous leur fournissons gratuitement. En analysant nos recherches sur le Web, Google connaît nos goûts et grâce à des algorithmes de recommandation et des publicités ciblées, il arrive à influencer nos achats.

Remarque : les bases de données et le langage SQL sont au programme de terminale.

Structure des données

Ci-dessous 3 formats (type de fichier textuel) très utilisés pour structurer des données :

|  |  |  |
| --- | --- | --- |
| **CSV**  Comma Separated Values | **JSON**  JavaScript Object Notation | **XML**  Extensible Markup Language |
|  |  |  |
| Le séparateur peut être une virgule, un point virgule, une tabulation ... | Chaque donnée possède son descripteur. | Chaque donnée possède son descripteur. |

**Donnée :** c'est une [information](https://fr.wikipedia.org/wiki/Information) utilisable par un appareil informatique.

**Collection :** ensemble de données partageant des caractéristiques communes.

**Table :** moyen de disposer de manière structurée les données d'une collection.

**Descripteur** : nom d'une catégorie de données ayant des caractéristiques communes.

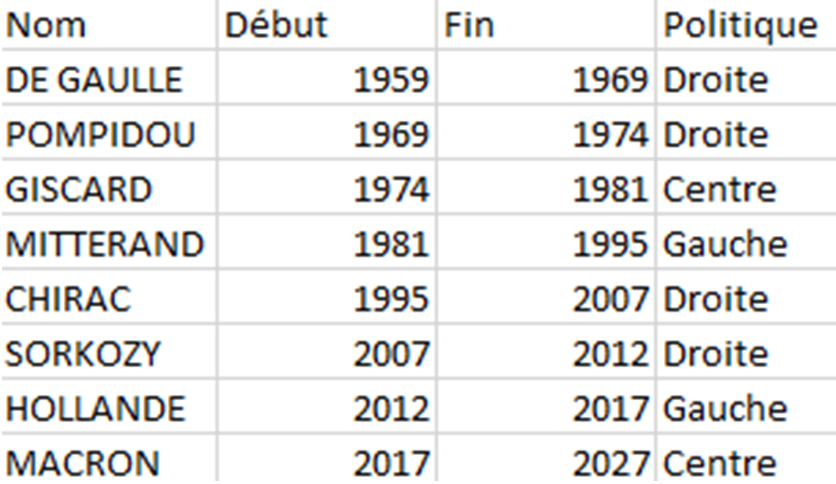
**Clé**: élément qui permet d'identifier un enregistrement.

**Enregistrement** : ensemble de valeurs liées à une clé.

**Valeur** : élément lié à un descripteur dans un enregistrement.

**Traitement de données :** c'est une série de [processus](https://fr.wikipedia.org/wiki/Processus_(informatique)) qui permettent d'extraire des [information](https://fr.wikipedia.org/wiki/Information)s ou d'en produire d’autres à partir de [données](https://fr.wikipedia.org/wiki/Donnée) brutes.

**Exemple avec la collection des présidents de la 5ième république Française :**



**Descripteurs**

**Clé** de l’enregistrement décrivant la présidence de Jacques CHIRAC

**Valeur** du descripteur **Fin** pour la clé **CHIRAC**

**Enregistrement** lié à la clé **HOLLANDE**

Dans le tableau suivant, mettez les descripteurs\* correspondant en face de leur collection :

|  |  |
| --- | --- |
| **Collection :** | **Descripteurs :** |
| Liste des élèves sur Pronote |  |
| Arbre généalogique |  |
| Classification périodique des éléments |  |
| Pages jaunes |  |
| Dictionnaire |  |

**\*Liste des descripteurs :**

département, commune, nom, numéro de téléphone

mot, définition, prononciation ...

ascendant, descendant, nom …

nom, numéro, symbole …

nom, date de naissance, langues vivantes ...

**Composition d'un fichier CSV**

Un fichier au format **csv** est un document **texte.**

La première ligne d'un fichier **CSV** contient les **descripteurs** puis chaque nouvelle ligne contient les **enregistrements** associés aux **descripteurs**.

Les valeurs sont séparées par une virgule, un point-virgule, une tabulation … au choix.

* Ouvrez le fichier **table.csv** dans l'**éditeur de texte** bloc note.
* Ouvrez le fichier **table.csv** dans le **tableur** Excel ou CALC.

En France pour que la mise automatique sous forme de tableau soit effective, dans un tableur, on utilise le **point-virgule** comme séparateur.

Les anglo-saxons utilisent la **virgule** comme séparateur.

Les Français utilisent le **point-virgule** comme séparateur car la **virgule** est utilisée pour les nombres décimaux (les anglo-saxons utilisent le **point** pour les nombres décimaux).

 

**Import d'une table**

* Testez le code suivant permettant l'import d’un fichier CSV existant, en lecture ('r') :

L=[]

f=open('table.csv','r')

for ligne in f:

L.append(ligne)

print(L)

f.close()

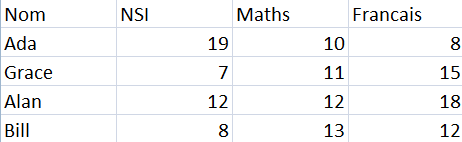
* Décrivez le contenu de la liste L.

Note : le caractère '\n' correspond au passage à la ligne suivante dans la table.

**Code 1\*** : Ecrivez la fonction import\_table(fichier) qui permet d’ajouter chaque ligne de la table (fichier) dans une liste.

Cette fonction retournera la liste L.

Instruction pour éliminer '\n' de la chaine de caractères : ligne=ligne.strip()



🡪 L = ['Nom;NSI;Maths;Francais', 'Ada;19;10;8', 'Grace;7;11;15', 'Alan;12;12;18', 'Bill;8;13;12']

**Code 2\*\*** : Modifiez la fonction import\_table(fichier) pour que chaque valeur de la table devienne un élément de la liste L.

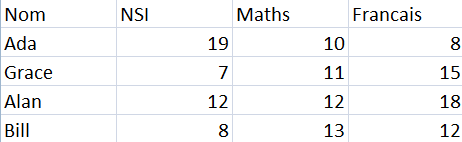
Cette fonction retournera la liste L.

Pour récupérer séparément les valeurs des colonnes de la table on va les stocker dans des variables :

Nom,NSI,Maths,Francais = ligne.split(';')

L.append(Nom)

...



🡪 tab = ['Nom', 'NSI', 'Maths', 'Francais', 'Ada', '19', '10', '8', 'Grace', '7', '11', '15', 'Alan', '12', '12', '18', 'Bill', '8', '13', '12']

**Recherche dans une table**

* Testez le code suivant permettant l'import d’un fichier CSV existant, en lecture ('r') :

f=open('table.csv','r')

lignes\_sans\_premiere = f.readlines()[1:]

print(lignes\_sans\_premiere)

f.close()

* Que constatez-vous ?

**Code 3\*\*** : Ecrivez la fonction recherche\_eleves(fichier, note\_ref) qui permet de créer la liste LnomsEleves des élèves présents dans le fichier, ayant un note supérieure à note\_ref en NSI.

Cette fonction retournera la liste LnomsEleves.

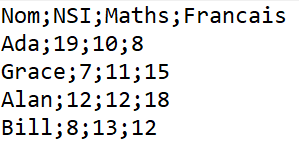
Attention : avec le splitchaque élément est transformé en chaîne de caractères donc 17  "17".

Attention : "7" > "17" alors que 7 < 17.

**Tri d'une table issue d'un fichier CSV**

**Code 4\*\*** : Ecrivez la fonction table\_listes(fichier) qui permet de convertir un fichier CSV en une liste de listes où chaque liste correspond à une ligne du fichier CSV sauf la première.

Cette fonction retournera la liste de listes LL.

fichier : 

🡪 LL = [['Ada', 19.0, 10.0, 8.0], ['Grace', 7.0, 11.0, 15.0], ['Alan', 12.0, 12.0, 18.0], ['Bill', 8.0, 13.0, 12.0]]

Ce qui est différent de :

L = ['Nom', 'NSI', 'Maths', 'Francais', 'Ada', '19', '10', '8', 'Grace', '7', '11', '15', 'Alan', '12', '12', '18', 'Bill', '8', '13', '12']

* Testez le code suivant (donnez une valeur entre 1 et 3 à l’indice i), que constatez-vous ?

from operator import itemgetter

print("Réultat :\n",sorted(LL,key=itemgetter(i)))

**Code 5\*** : Ecrivez le code pour trier LL selon l'ordre croissant des notes de NSI.

Rangez le résultat dans le fichier table\_tri.csv.

Code de la fonction creation\_csv() permettant de créer un fichier CSV, en écriture ('w') avec le ";" comme séparateur :

import csv    
 def creation\_csv(fichier, LL):   
     f=open(fichier,'w',newline='')  
     ligne = csv.writer(f, delimiter=';')

ligne.writerow(['Nom', 'NSI', 'Maths', 'Francais'])  
     for liste in LL:  
         ligne.writerow(liste)  
     f.close()

creation\_csv('table\_tri.csv',LL)

**Supprimer les doublons dans un fichier CSV**

* Créez à la main le fichier table\_avec\_doublon.csv qui correspond au fichier table.csv dans lequel vous dupliquerez l’enregistrement (=la ligne) "Ada". Ceci afin de créer un doublon.

Les doublons sont fréquents dans les tables notamment lorsqu’elles sont complétées par plusieurs personnes.

**Code 6\*\*** : Ecrivez la fonction supprimer\_doublon(LL) permettant de supprimer les doublons (**si 2 lignes sont égales** → suppression d'une des 2 lignes) de la liste de listes LL.

La liste de listes LL sera créée à partir du fichier table\_avec\_doublon.csv.

Cette fonction ne retournera rien.

Aide : pour compter le nombre d'occurrences dans une liste vous pouvez utiliser la méthode count().

LL = [**['Ada', 19.0, 10.0, 8.0]**, ['Grace', 7.0, 11.0, 15.0], ['Alan', 12.0, 12.0, 18.0], ['Bill', 8.0, 13.0, 12.0], **['Ada', 19.0, 10.0, 8.0]**]

🡪 LL = [['Grace', 7.0, 11.0, 15.0], ['Alan', 12.0, 12.0, 18.0], ['Bill', 8.0, 13.0, 12.0], **['Ada', 19.0, 10.0, 8.0]**]

**Code 7\*** : Ecrivez le code pour créer le fichier table\_sans\_doublon.csv correspondant à table\_avec\_doublon.csv auquel la ligne "Ada" aura été enlevée.

**Code 8\*\*\* (facultatif)** : Ecrivez la fonction supprimer\_doublon2(LL) permettant de supprimer les doublons (**si 2 lignes ont le même noms** → suppression d'une des 2 lignes) de la liste de listes LL.

La liste de listes LL sera créée à partir du fichier table\_avec\_doublon2.csv (=table\_sans\_doublon.csv dont il faudra changer des notes pour Ada).

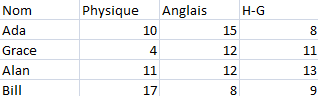
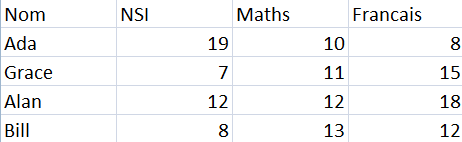
Cette fonction ne retournera rien.

LL = [**['Ada', 9.0, 17.0, 18.0]**, ['Grace', 7.0, 11.0, 15.0], ['Alan', 12.0, 12.0, 18.0], ['Bill', 8.0, 13.0, 12.0], **['Ada', 19.0, 10.0, 8.0]**]

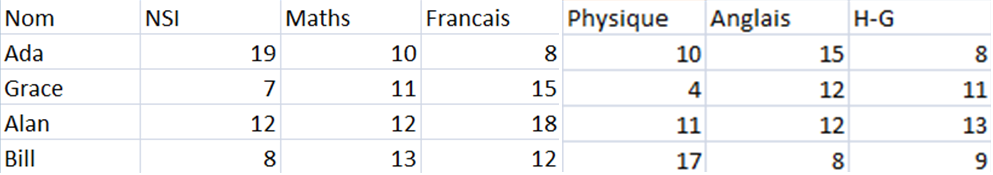
🡪 LL = [['Grace', 7.0, 11.0, 15.0], ['Alan', 12.0, 12.0, 18.0], ['Bill', 8.0, 13.0, 12.0], **['Ada', 19.0, 10.0, 8.0]**]

**Fusion de tables CSV**

Table table.csv existante : Table table\_bis.csv existante :



Nous allons chercher à fusionner (joindre) les 2 tables pour obtenir la table suivante :



**Code 8\*\*** : Ecrivez la fonction liste\_vers\_liste\_de\_dicos(LL) qui transforme LL en une liste de dictionnaires.

Cette fonction retournera la liste de dictionnaires L\_dicos.

Rappel sur les dictionnaires :

dic={}

dic['Nom']="Ada"

LL = [['Ada', 19.0, 10.0, 8.0],['Grace', 7.0, 11.0, 15.0],['Alan', 12.0, 12.0, 18.0],['Bill', 8.0, 13.0, 12.0]] **→** L\_dicos=[{'Nom':'Ada','ISN':19,'Maths':10,'Francais':8},{'Nom':'Grace','ISN':7,'Maths':11,'Francais':15},{'Nom':'Alan','ISN':12,'Maths':12,'Francais':18},{'Nom':'Bill','ISN':8,'Maths':13,'Francais':12}]

LL\_bis = [{'Nom':'Ada', 'Physique':'10.0','Anglais':'15.0','H-G':'8.0'},{'Nom':'Grace', 'Physique':'4.0','Anglais':'12.0','H-G':'11.0'},{'Nom':'Alan', 'Physique':'11.0','Anglais':'12.0','H-G':'13.0'},{'Nom':'Bill', 'Physique':'17.0','Anglais':'8.0','H-G':'9.0'}

**→**

L\_dicos\_bis=[{'Nom':'Ada','Physique':10,'Anglais':15,'H\_G':8},{'Nom':'Grace','Physique':4,'Anglais':12,'H\_G':11},{'Nom':'Alan','Physique':11,'Anglais':12,'H\_G':13},{'Nom':'Bill','Physique':17,'Anglais':8,'H\_G':9}]

**Code 9\*\*\*** : Ecrivez la fonction fusion(L\_dicos,L\_dicos\_bis) qui fusionne les 2 listes de dictionnaires L\_dicos et L\_dicos\_bis.

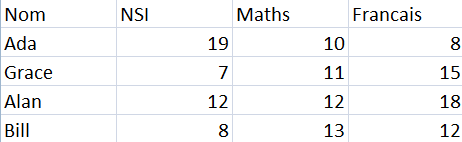
Cette fonction retournera la liste de dictionnaires L\_dicos\_fusion.

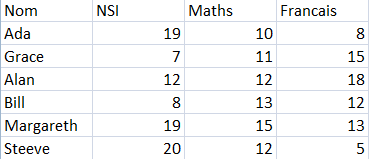
L\_dicos\_fusion=[{'Nom': 'Ada', 'NSI': '19', 'Maths': '10', 'Francais': '8', 'Physique': '10', 'Anglais': '15', 'H-G': '8'}, {'Nom': 'Grace', 'NSI': '7', 'Maths': '11', 'Francais': '15', 'Physique': '4', 'Anglais': '12', 'H-G': '11'}, {'Nom': 'Alan', 'NSI': '12', 'Maths': '12', 'Francais': '18', 'Physique': '11', 'Anglais': '12', 'H-G': '13'}, {'Nom': 'Bill', 'NSI': '8', 'Maths': '13', 'Francais': '12', 'Physique': '17', 'Anglais': '8', 'H-G': '9'}]

**Code 10\*\*** : Ecrivez la fonction fusion\_csv(L\_dico\_fusion) qui crée le fichier table\_fusion.csv qui est la table CSV contenant la fusion des 2 listes de dictionnaires.

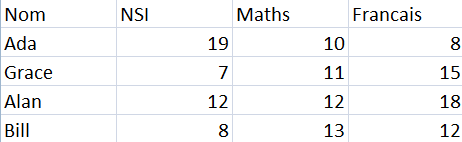
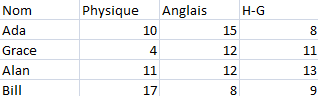
Nous venons de réaliser une **fusion horizontale** : ajout des descripteurs de plusieurs tables.

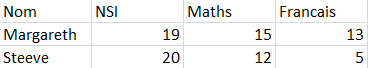
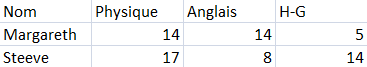
Il existe aussi la **fusion verticale** : ajout des enregistrements de plusieurs tables.

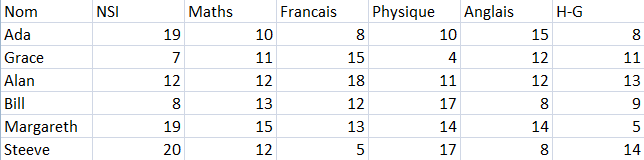
 + 

= 

Et enfin il existe la **fusion totale** : fusion horizontale + fusion verticale

 +  +

 + 

= 

**Remarque :**

Il est possible de trouver des enregistrements ayant la même clé → dans ce cas il faudra choisir un enregistrement plutôt qu'un autre où on considèrera que c'est une incohérence et on supprimera les enregistrements ayant la même clé.