

Applications Android

Données en table

Christophe Viroulaud

Première - NSI

Tab 01

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

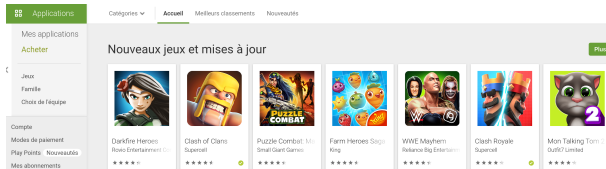


FIGURE 1 – Magasin d'applications

Chaque carte contient plusieurs informations :

- ▶ nom
- ▶ description
- ▶ note
- ▶ image
- ▶ ...

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Options de séparateur

☐ Largeur fixe ☒ Séparé par

☒ Tabulation ☒ Virgule ☒ Point-virgule ☐ Espace ☐ Autre

☐ Fusionner les séparateurs ☐ Espaces superflus

Séparateur de chaîne de caractères :

Autres options

☐ Formater les champs entre guillemets comme texte ☐ Détecter les nombres spéciaux

Champs

Type de colonne :

	Standard	Standard	Standard	Standard
1	Name	Category	Rating	Reviews
2	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159
3	Coloring book moana	ART_AND_DESIGN	3.9	967
4	U Launcher Lite - FREE Live Cool Themes, Hide Apps	ART_AND_DESIGN	4.7	87510
5	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644
6	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967
7	Paper flowers instructions	ART_AND_DESIGN	4.4	167
8	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178
9	Infinite Painter	ART_AND_DESIGN	4.1	26815

FIGURE 2 – Données stockées dans un fichier texte

Comment manipuler un jeu de données par
programmation ?

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

1. Données en table

1.1 Présentation

1.2 Lire un fichier avec Python

1.3 Lire un fichier `csv` avec Python

2. Manipuler les données

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

À retenir

Un fichier de données regroupe des informations organisées.

Dans un fichier `csv` (**C**omma **S**eparated **V**alues), chaque ligne représente une nouvelle **entité** du jeu de données. Les informations sont séparées par une virgule, un point-virgule, une tabulation...

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 1 :

1. Télécharger et décompresser l'annexe `app-android.zip` sur le site <https://cviroulaud.github.io>
2. Avec le bloc-notes de Windows ouvrir le fichier `notes.csv`
3. Ouvrir le logiciel *LibreOffice Calc*.
4. Depuis ce logiciel, ouvrir à nouveau le fichier `notes.csv`

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

"Nom",	"Prénom",	"Moyenne"
"Turing" ,	"Alan" ,	"19"
"Von Neuman" ,	"John",	"18"
"Dijkstra" ,	"Edsger",	"17.5"
"Church" ,	"Alonso",	"19"

Tableau 1 – Chaque ligne représente une nouvelle entité

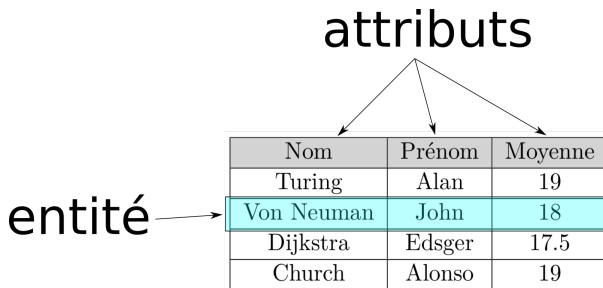


FIGURE 3 – LibreOffice reconnaît les fichiers `csv`

À retenir

Un **attribut** identifie les données d'une même colonne.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 2 :

1. Ouvrir le fichier `googleplaystore.csv` avec LibreOffice Calc.
2. Trouver le séparateur utilisé.
3. Repérer les attributs.

Options de séparateur

☐ Largeur fixe ☒ Séparé par

☒ Tabulation ☒ Virgule ☒ Point-virgule ☐ Espace ☐ Autre

☐ Fusionner les séparateurs ☐ Espaces superflus

Séparateur de chaîne de caractères :

Autres options

☐ Formater les champs entre guillemets comme texte ☐ Détecter les nombres spéciaux

Champs

Type de colonne :

	Standard	Standard	Standard	Standard
	Name	Category	Rating	Reviews
1	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159
2	Coloring book moana	ART_AND_DESIGN	3.9	967
3	U Launcher Lite - FREE Live Cool Themes, Hide Apps	ART_AND_DESIGN	4.7	87510
4	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644
5	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967
6	Paper flowers instructions	ART_AND_DESIGN	4.4	167
7	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178
8	Infinite Painter	ART_AND_DESIGN	4.1	26815

FIGURE 4 – Fichier csv ouvert avec LibreOffice. Le séparateur est la virgule.

Remarque

Quand le nombre de données est important, un tableur est moins approprié pour étudier les données.

- ▶ *Name* : nom,
- ▶ *Category* : catégorie d'application,
- ▶ *Rating* : note (sur 5),
- ▶ *Reviews* : nombre de commentaires,
- ▶ *Installs* : nombre d'installations.

Remarque

Chaque ligne représente une application Android

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

1. Données en table

1.1 Présentation

1.2 Lire un fichier avec Python

1.3 Lire un fichier `csv` avec Python

2. Manipuler les données

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

À retenir

Un fichier de données est une **entrée** dans un programme. Quand on ouvre un fichier externe dans un programme, il est verrouillé. Il faut le libérer à la fin de l'utilisation.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 # ouvrir le fichier
2 fichier = open("notes.csv")
3
4 # utiliser le fichier
5
6 # libérer le fichier
7 fichier.close()
```

Code 1 – Ouvrir et fermer

Activité 3 :

1. Créer le fichier `notes.py` dans le même dossier que `notes.csv`
2. Écrire le code 3.

```
1 # ouvrir le fichier
2 fichier = open("notes.csv")
3
4 # utiliser le fichier
5
6 # libérer le fichier
7 fichier.close()
```

Code 2 – Ouvrir et fermer

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 # ouvrir le fichier
2 with open("notes.csv") as notes:
3     # utiliser le fichier
4
5 # fermeture automatique
```

Code 3 – Autre méthode

1. Données en table

1.1 Présentation

1.2 Lire un fichier avec Python

1.3 Lire un fichier `csv` avec Python

2. Manipuler les données

Données en table

Présentation

Lire un fichier avec Python

**Lire un fichier `csv` avec
Python**

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

À retenir

Un programme Python peut ouvrir tous types de fichiers externes (image, texte...). Il existe de nombreuses bibliothèques adaptées pour chaque type.

La bibliothèque `csv` permet de **lire** ou **écrire** ligne après ligne dans un fichier `csv`.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 4 : Tester le code :

```
1 import csv
2 # ouvrir le fichier
3 fichier = open("notes.csv")
4
5 lecteur = csv.reader(fichier)
6 for ligne in lecteur:
7     print(ligne)
8
9 # libérer le fichier
10 fichier.close()
```

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

À retenir

La méthode **reader** de la bibliothèque **csv** est un **itérateur**. À chaque appel (itération), l'itérateur lit une nouvelle ligne et la renvoie sous forme de tableau.

Attention : un itérateur se vide au fur et à mesure. Il ne peut être parcouru qu'une fois.

Remarque

La fonction **range** est un itérateur qui fournit des entiers.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier **csv** avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Remarque

La méthode `reader` ne distingue pas les attributs.

```
1 ['nom', 'prenom', 'moyennes']  
2 ['Turing', 'Alan', '19']  
3 ['Von Neuman', 'John', '18']  
4 ['Dijkstra', 'Edsger', '17.5']  
5 ['Church', 'Alonso', '19']
```

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 5 :

1. Remplacer la méthode `reader` par `DictReader`.
2. Observer les objets obtenus.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 lecteur = csv.DictReader(fichier)
2 for ligne in lecteur:
3     print(ligne)
```

```
1 OrderedDict([('nom', 'Turing'), ('prenom', '
    Alan'), ('moyennes', '19')])
2 OrderedDict([('nom', 'Von Neuman'), ('prenom'
    , 'John'), ('moyennes', '18')])
3 OrderedDict([('nom', 'Dijkstra'), ('prenom',
    'Edsger'), ('moyennes', '17.5')])
4 OrderedDict([('nom', 'Church'), ('prenom', '
    Alonso'), ('moyennes', '19')])
```

Code 4 – Dictionnaires ordonnés

Remarque

Les dictionnaires ordonnés seront vus comme de simples dictionnaires.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

```
1 {"nom": "Turing", "prenom": "Alan", "moyennes": "19"}  
2 {"nom": "Von Neuman", "prenom": "John", "moyennes": "18"}  
3 {"nom": "Dijkstra", "prenom": "Edsger", "moyennes": "17.5"}  
4 {"nom": "Church", "prenom": "Alonso", "moyennes": "19"}
```

Code 5 – OrderedDict → dict

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier **csv** avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Remarque

La méthode `DictReader` identifie la première ligne comme la liste des attributs du fichier.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 6 :

1. Créer un fichier Python `appgoogle.py`
2. Dans le programme, ouvrir le fichier `googleplaystore.csv`
3. Construire le tableau `applications` de dictionnaires. Chaque dictionnaire contiendra les informations d'une application.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 import csv
2 # charge le fichier dans le programme
3 fichier = open("googleplaystore.csv", encoding="
    utf8")
4
5 # crée un itérateur sur les données
6 lecteur_donnees = csv.DictReader(fichier)
7
8 applications = []
9 for ligne in lecteur_donnees:
10     applications.append(ligne)
11
12 # libère le fichier externe
13 fichier.close()
```

1. Données en table

2. Manipuler les données

2.1 Valider les données

2.2 Rechercher des données

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

À retenir

Par défaut les données chargées dans un programme par la bibliothèque `csv` sont des chaînes de caractère.

Remarque

Il appartient au développeur de traiter les données afin qu'elles soient utilisables dans le programme.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier `csv` avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Dans le programme `notes.py`, la moyenne doit être considérée comme un nombre flottant.

```
1  eleves = []
2  # Pour chaque ligne
3  for ligne in lecteur_donnees:
4      un_eleve = {}
5      # Pour chaque couple de la ligne
6      for cle, val in ligne.items():
7          # type correctement la moyenne
8          if cle == "moyennes":
9              val = float(val)
10             un_eleve[cle] = val
11
12     eleves.append(un_eleve)
```

Code 6 – Typage des données

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 for ligne in lecteur_donnees:
```

Code 7 – ligne est un dictionnaire (OrderedDict)

```
1 un_eleve = {}  
2 # Pour chaque couple de la ligne  
3 for cle, val in ligne.items():  
4     # type correctement la moyenne  
5     if cle == "moyennes":  
6         val = float(val)  
7     un_eleve[cle] = val
```

Code 8 – Analyse des données d'un élève

```
1 eleves.append(un_eleve)
```

Code 9 – Le dictionnaire traité est ajouté au tableau

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 7 :

1. Quels attributs des applications Android demandent un traitement ?
2. Modifier alors le programme `appgoogle.py` pour typer correctement les informations récoltées.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

- ▶ *Rating* est une note moyenne ; c'est un flottant.
- ▶ *Installs* et *Reviews* sont des entiers.

```
1 applications = []
2 for ligne in lecteur_donnees:
3     une_appli = {}
4     for cle, val in ligne.items():
5         # validation des données
6         if cle == "Rating":
7             val = float(val)
8         if cle == "Installs" or cle == "Reviews":
9             val = int(val)
10
11     une_appli[cle] = val
12
13 applications.append(une_appli)
```

Code 10 – Tentative de conversion des données

```
1 ValueError: invalid literal for int() with base 10: 'NaN'
```

Code 11 – erreur de traitement

À retenir

Les données brutes doivent être validées par le programmeur. Il devra :

- typer correctement les informations,
- vérifier la cohérence des informations.

Name	Category	Rating	Reviews	Installs
CX Network	BUSINESS	NaN	0	NaN

Tableau 2 – Des valeurs particulières

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 applications = []
2 for ligne in lecteur_donnees:
3     une_appli = {}
4     for cle, val in ligne.items():
5         # validation des données
6         if cle == "Rating":
7             if val == "NaN":
8                 val = -1.0 # non noté
9             else:
10                val = float(val)
11        if cle == "Installs" or cle == "Reviews":
12            if val == "NaN":
13                val = 0
14            else:
15                val = int(val)
16
17        une_appli[cle] = val
18
19    applications.append(une_appli)
```

1. Données en table

2. Manipuler les données

2.1 Valider les données

2.2 Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Une action courante sur un jeu de données est de sélectionner certaines lignes en fonction d'un critère. C'est le rôle d'un moteur de recherche.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 8 :

1. Écrire la fonction `trouver_app(mot_cle: str, tab: list) → list` qui renvoie la liste des applications du tableau `tab` dont le nom contient `mot_cle`. Indications :
 - ▶ L'instruction `in` permet de vérifier si une sous-chaîne est présente dans une chaîne de caractère.
 - ▶ Tous les mots commencent par une majuscule.
2. Chercher toutes les applications dont le nom contient le mot *Photo*.
3. Compter le nombre d'applications renvoyées.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 def trouver_app(mot_cle: str, tab: list) -> list:
2     """
3     renvoie les applications contenant le 'mot_cle'
4
5     chaque mot commence par une majuscule dans la
6     table
7     """
8     res = []
9     for app in tab:
10         # app est un dictionnaire
11         if mot_cle in app["Name"]:
12             res.append(app)
13     return res
```


Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 applications = trouver_app("Photo", applications)
2 len(applications)
```

Code 12 – Appel de la fonction

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Activité 4 :

4. Écrire la fonction `meilleur_app_notee(tab: list) → dict` qui renvoie l'application la mieux notée du tableau `tab`.
5. Trouver l'application photo la mieux notée.

Correction

```
1 def meilleur_app_notee(tab: list) -> dict:
2     """
3     renvoie l'application avec la meilleure note de
4     tab
5     """
6     note_maxi = 0
7     meilleure_app = None
8     for app in tab:
9         if app["Rating"] > note_maxi:
10             note_maxi = app["Rating"]
11             meilleure_app = app
12     return meilleure_app
```

```
1 applications = trouver_app("Photo", applications)
2 meilleur_app_notee(applications)
```

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les
données

Valider les données

Rechercher des données

Sélectionner

Agréger

Code 13 – Appels des fonctions

On peut imaginer proposer à l'utilisateur toutes les applications notées au-dessus de la moyenne.

Activité 5 :

1. Écrire la fonction `moyenne_note(tab: list) → float` qui calcule la note moyenne des applications de `tab`. Le résultat sera arrondi à deux chiffres significatifs.
2. Dans le programme principal, construire le tableau des applications photo dont la note est strictement supérieure à la moyenne.

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

Données en table

Présentation

Lire un fichier avec Python

Lire un fichier csv avec
Python

Manipuler les données

Valider les données

Rechercher des données

Sélectionner

Agréger

```
1 def moyenne_note(tab: list) -> float:
2     """
3     renvoie la note moyenne des apps de tab
4     """
5     somme = 0
6     nb = 0
7     for app in tab:
8         # Si l'app a déjà été notée
9         if app["Rating"] >= 0:
10             somme += app["Rating"]
11             nb += 1
12     return round(somme/nb, 2)
```

```
1 app_photos = trouver_app("Photo", applications)
2
3 moyenne = moyenne_note(app_photos)
4
5 meilleures_app = []
6 for app in app_photos:
7     if app["Rating"] > moyenne:
8         meilleures_app.append(app)
```

Code 14 – Sélection des meilleures applications photos

```
1 meilleures_app = [app for app in app_photos
2                   if app["Rating"] > moyenne]
```

Code 15 – Version par compréhension