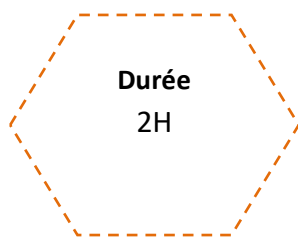
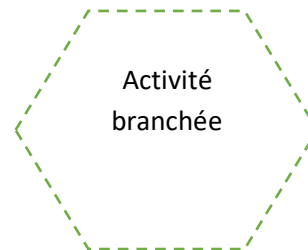


## 07. Traitement de données



Thématique

**LOCALISATION  
CARTOGRAPHIE  
MOBILITE**



### Description de l'activité

Dans cette activité, l'enseignant.e utilise des données ouvertes tirées de data.gouv.fr et les traite avec les élèves, d'une part avec un tableur et d'autre part avec des programmes Python pour extraire et afficher des données pertinentes.

### Objectifs pédagogiques ou compétences

Objectifs généraux	Objectifs intermédiaires	Compétences
<b>Notions de cours</b>	<ul style="list-style-type: none"><li>- Comprendre l'importance des données et de leur traitement automatique</li><li>- Revoir / Découvrir quelques manipulations de base des données (Excel / Python)</li><li>- Découvrir la bibliothèque Folium de Python</li></ul>	<ul style="list-style-type: none"><li>- Manipuler des données ouvertes</li><li>- Traiter des données structurées avec un tableur</li><li>- Traiter automatiquement des données avec un programme informatique</li><li>- Afficher des données préalablement traitées sur une carte</li></ul>

### Matériel et outils

- Fiche activité élève à imprimer
- 1 poste par élève

### Tags

#géolocalisation ; #cartographie ; #donnéesOuvertes ; #tableur ; #bibliothèqueFolium ; #traitementAutomatique

# Déroulé de l'activité

## Introduction : (15 minutes)

- **Présenter les objectifs de la séance (contenu théorique et productions attendues)**
- **Introduire la thématique des données**

L'enseignant.e demande aux élèves s'ils savent ce qu'est une donnée et quand est-ce qu'elles sont « créées » dans leur quotidien.

Les données sont créées à chaque fois que des événements, des actions ou des mesures sont consignés. Elles aident à comprendre, analyser et décider dans de nombreux domaines, mais soulèvent aussi des questions juridiques ou éthiques, notamment sur la protection de la vie privée.

Suggestions de réflexion :

**Achat en ligne** : Quand on achète quelque chose en ligne, les caractéristiques de l'article acheté, le prix, la date et l'heure de l'achat, ainsi que nos informations personnelles et notre mode de paiement sont des données. Cela peut permettre de simplifier un prochain achat, mais aussi de mieux cerner nos préférences.

**Réseaux sociaux** : À chaque fois que l'on poste un message, une photo ou un commentaire sur les réseaux sociaux, on crée des données qui contiennent notre message, la date et l'heure de publication, les éventuels likes ou commentaires, etc.

**Utilisation du smartphone** : Elle produit des données comme les applications qu'on utilise, les sites web qu'on consulte, les appels qu'on fait et les messages qu'on envoie.

**Données de localisation** : Les applications de cartographie et de navigation les enregistrent.

**Historique de navigation web** : Les sites web qu'on visite, les liens sur lesquels on appuie et le temps passé sur chaque page sont consignés.

## Étape 1 – Traitement de données avec tableur (30 minutes)

- **Présentation du site (5 minutes)**

L'enseignant.e présente le site <https://www.data.gouv.fr/> en expliquant à quoi il sert, en montrant plusieurs types de données recueillies dans les thèmes à la une, puis en expliquant comment on peut faire des recherches plus poussées en cliquant sur « Données », tout en montrant l'exemple avec « salles de cinéma », ce qui permettra de retrouver la feuille .csv qui servira pour l'exercice.

**Feuille** : <https://www.data.gouv.fr/fr/datasets/les-salles-de-cinema-en-ile-de-france/> , mise à jour le 6 novembre 2021.

- **Importation des données ouvertes** (<5 minutes)

Les élèves téléchargent le fichier.csv et le renomment « salles.csv ».

- **Les fichiers .csv** (<5 minutes)

L'enseignant.e fait le point sur ce qu'est un fichier .csv, et analyse rapidement le fichier avec les élèves.

- **Traitement des données** (20 minutes)

Les élèves manipulent rapidement 2 formules permettant de mieux analyser les données.

## Étape 2 – Traitement automatisé des données (45 minutes)

Les élèves manipulent plusieurs codes Python pour s'initier au traitement automatique des données.

## Étape 3 – Géolocalisation des données (20 minutes)

Les élèves manipulent plusieurs codes Python pour générer une carte avec les localisations annotées.

## Conclusion (10 minutes)

- **Bilan de la séance :**

Pour clôturer la séance, on peut revenir sur les principales difficultés rencontrées pendant l'activité. Éventuellement, il est possible de finir sur un court échange autour de :

- **L'importance de l'automatisation des données**

L'enseignant.e peut interroger les élèves sur les bénéfices de l'automatisation des données.

Suggestions de discussion :

**Gain de temps et d'efficacité :** L'automatisation permet de réaliser des tâches récurrentes et fastidieuses de manière automatique, et avec une vitesse largement supérieure à celle d'un humain.

**Réduction des erreurs :** Les tâches manuelles sont exposées aux erreurs humaines, et l'automatisation diminue les risques d'erreurs. Cela permet aussi d'assurer une homogénéité dans le protocole de traitement.

**Surveillance et Notification :** L'automatisation permet de contrôler en permanence les données et de déclencher des notifications en cas de limites ou de conditions prédéfinies. Cela permet également de repérer ou anticiper des situations à risque.

## ● Les métiers en lien

L'enseignant.e peut également évoquer les principaux métiers en lien avec l'analyse et le traitement des données. En fonction du temps, on peut demander aux élèves de faire des recherches, et éventuellement construire leur propre « top 3 » des métiers qui les intéressent dans ce domaine. Voici quelques exemples :

- **Analyste de données** : Analyse les données pour identifier des tendances, des modèles et des informations utiles pour la prise de décision.
- **Data Scientist** : Utilise des techniques avancées d'analyse de données, de machine learning et de statistiques pour extraire des prédictions à partir des données.
- **Ingénieur en traitement des données** : Conçoit et développe des systèmes de gestion des données, des bases de données et des pipelines de traitement des données.
- **Analyste de business intelligence** : Collecte, organise et analyse des données pour aider les entreprises à prendre des décisions stratégiques et à surveiller leurs performances.
- **Ingénieur en apprentissage automatique** : Développe des modèles d'apprentissage automatique pour automatiser des tâches et créer des systèmes intelligents.
- **Analyste de données marketing** : Analyse les données marketing pour évaluer l'efficacité des campagnes, identifier les segments de clientèle et améliorer les stratégies marketing.
- **Analyste financier** : Utilise les données financières pour évaluer les performances de l'entreprise, prévoir les tendances économiques et recommander des stratégies d'investissement.
- **Data Engineer** : Gère l'infrastructure de données, construit et optimise les pipelines de données et assure la qualité des données pour les analyses ultérieures.
- **Architecte de données** : Conçoit l'architecture des systèmes de données, définit les flux de données et garantit la cohérence et l'intégrité des données.
- **Analyste en cybersécurité** : Analyse les données de sécurité pour détecter et prévenir les menaces, les attaques et les vulnérabilités dans les systèmes informatiques.
- **Data Analyst en santé** : Analyse les données médicales pour identifier des modèles de santé, améliorer les soins aux patients et prendre des décisions éclairées en matière de santé.
- **Analyste en sciences sociales** : Utilise les données pour étudier les comportements humains, les tendances sociales et les modèles sociétaux.

En évaluation, l'enseignant.e peut demander aux élèves de choisir une feuille parmi plusieurs en choix (ou de leur en imposer une).

- Grâce à Excel, retrouver 2-3 informations en lien avec une quantité d'éléments, par exemple :
  - Dans une ville ou département
  - Créés dans la même décennie
  - De même type
- Avec Python, recréer une carte comme lors de l'étape 3.

Voici quelques propositions de feuilles contenant des coordonnées de localiation :

- [SLA Déchèteries](#)
- [SLA Médiathèques et bibliothèques](#)
- [Clubs du 3<sup>e</sup> âge](#)
- [Stations de métro à Toulouse](#)
- [Localisation des mairies à Lille](#)
- [Liste des établissements enregistrés comme opérateurs de l'alimentation animale](#)
- [Bornes de chargement publiques pour voitures électriques](#)
- [Panneaux à message variable](#)

# Traitement et affichage de données

## Fiche activité - Correction

### Étape 1 – Traitement de données avec tableur

- **Importation des données ouvertes**

Téléchargez le fichier .csv mis à jour le 6 décembre 2021, puis renommez-le « salles ».



- Quand ce fichier a-t-il été créé ?

Dans l'onglet informations : 2 juillet 2017.

- **Les fichiers .csv**

- Qu'est-ce qu'un fichier .csv ?

Un fichier CSV (Comma-Separated Values) est un type de fichier texte utilisé pour stocker des données sous forme de tableaux ou de listes. Chaque ligne du fichier représente une ligne de données, et les valeurs dans chaque ligne sont séparées par des virgules (ou par un autre séparateur, comme un point-virgule). Elles peuvent contenir plusieurs colonnes de données, correspondant généralement à un champ ou une caractéristique spécifique des données : Nom, Âge, Ville, Code postal, Identifiant, Adresse, Adresse mail, ...

- Ouvrez le fichier avec **le bloc-notes**. Que contiennent les deux premières lignes ?

L1 : Descripteurs, L2 et suivantes : données

- Quel type de séparateur est utilisé dans ce fichier ?

Le point-virgule : « ; »

- **Traitement des données**

Importez le fichier **dans un tableur**.

- Quels sont les descripteurs de cette table de données ?

Ce sont tous les descripteurs ligne 1 :

ndeg\_auto; dep; nom; region\_administrative; adresse; code\_insee; commune; situation\_geographique; ecrans; fauteuils; tranche\_d\_entrees; propriétaire; programmeur; categorie\_art\_et\_essai; label\_art\_et\_essai; genre; multiplexe; population\_de\_la\_commune\_2015; unite\_urbaine\_2010; population\_unite\_urbaine\_2015; semaines\_d\_activite\_2020; seances\_2020; entrees\_2020; entrees\_2019; evolution\_entrees\_2020\_2019; art\_et\_essai; nombre\_de\_films\_programmes\_2020; nombre\_de\_films\_inedits\_2020; nombre\_de\_films\_en\_semaine\_1\_2020; pdm\_en\_entrees\_des\_films\_francais\_2020; pdm\_en\_entrees\_des\_films\_americains\_2020; pdm\_en\_entrees\_des\_films\_europeens\_2020; pdm\_en\_entrees\_des\_autres\_films\_2020; films\_art\_et\_essai\_2020; part\_des\_seances\_de\_films\_art\_et\_essai\_2020; pdm\_en\_entrees\_des\_films\_art\_et\_essai\_2020; geo

- Charles habite à Carrières-sur-Seine et souhaite connaître le nombre de salles de cinéma de son département. Quelle formule doit-il écrire en B312 ?

=NB.SI(B2:B310 ;78). En 2023, la réponse est de 30.

- Charles souhaite maintenant connaître le nombre de salles de cinéma à Paris. Quelle formule doit-il écrire en G312 ?

La formule précédente ne fonctionne pas car dans les cases, il y a les arrondissements en plus de « Paris », or cette formule recherche « Paris » précisément.

Il faut écrire =NB.SI(G2:G311;"Paris\*"). L'astérisque (\*) est un caractère générique qui signifie "n'importe quelle série de caractères".

## Étape 2 – Traitement automatique des données

Écrivez le programme suivant :

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 print(descripteurs)
4 ligne = f.readline()
5 print(ligne)
```

- Expliquez les deux lignes qui s'affichent lors de l'exécution du programme.

La ligne 3 affiche le contenu de la variable « descripteurs », qui devrait être la première ligne du fichier CSV contenant les noms de colonnes.

La ligne 5 affiche le contenu de la variable « ligne », qui devrait être la première ligne de données du fichier CSV.

Écrivez le programme suivant :

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 for i in range(10):
4 ligne = f.readline()
5 print(ligne)
```

- Expliquez la ligne 3 et l'affichage obtenu lors de l'exécution de ce programme.

La boucle « for » commence à itérer 10 fois, pour chaque valeur de « i » de 0 à 9. À chaque itération de la boucle, la fonction `readline()` est utilisée pour lire une ligne du fichier après la précédente. Cette ligne est stockée dans la variable « ligne ». La ligne lue à chaque itération de la boucle est ensuite affichée à l'aide de la fonction `print(ligne)`.

Écrivez le programme suivant :

```
f = open("salles.csv","r")
2 descripteurs = f.readline()
3 for i in range(10):
4 ligne = f.readline()
5 donnees = ligne.split(";")
6 print(donnees[2])
```

- Expliquez les lignes qui s'affichent lors de l'exécution de ce programme.

À chaque itération de la boucle « for », la fonction `readline()` est utilisée pour lire une ligne du fichier après la précédente. Cette ligne est stockée dans la variable « ligne ». La ligne lue à chaque itération de la boucle est ensuite divisée en données individuelles en utilisant la méthode `split(";")`, qui suppose que le séparateur dans le fichier CSV est un point-virgule.

La partie de données sélectionnée, en utilisant `donnees[2]`, correspond à la troisième colonne de chaque ligne. Cette partie est ensuite affichée à l'aide de la fonction `print()`.

- Comment modifier le programme pour qu'il affiche les adresses des salles de cinéma ?

Pour afficher les adresses des salles de cinéma, il faut accéder à la colonne qui contient les adresses. En 2023, celle-ci est dans la case « E ». Il faut donc modifier la ligne ainsi : `print(donnees[4])`.

On souhaite étudier toutes les données du fichier. Pour cela, on considère le programme suivant :

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 ligne = f.readline()
4 compteur = 0
5 while ligne != "":
6 donnees = ligne.split(";")
7 print(donnees[3])
8 ligne = f.readline()
9 compteur = compteur + 1
10 print(compteur)
```

- Expliquer l'utilisation de la boucle conditionnelle « while » à la ligne 5.

L'utilisation de la boucle conditionnelle « while » à la ligne 5 est pour effectuer une itération continue (répétée) du bloc de code à l'intérieur de la boucle tant que la condition donnée est vraie. Dans le cas présent, tant que la variable « ligne » n'est pas une chaîne de caractères vide, ce qui signifie qu'il y a encore des lignes à lire dans le fichier.

- Écrivez le programme et expliquez les lignes qui s'affichent lors de son exécution.

Ce code lit les données de chaque ligne du fichier CSV et affiche la quatrième colonne de chaque ligne tout en comptant le nombre total de lignes lues.

- On souhaite calculer le nombre de salles de cinéma dans le 78. Compléter le programme suivant afin qu'il affiche ce nombre.

```
1 f = open("salles.csv", "r")
2 descripteurs = f.readline()
3 ligne = f.readline()
4 compteur = 0
5 while ligne != "":
6     donnees = ligne.split(";")
7     if donnees[1] == "78":
8         compteur += 1
9     ligne = f.readline()
10
11 print(compteur)
```



## Étape 3 – Géolocalisation des données

Carla souhaite visualiser la position des salles de cinéma sur une carte. Pour ce faire, nous allons créer un programme qui utilise les coordonnées de latitude et de longitude à partir du fichier "salles.csv" pour générer une carte dans le fichier "carte\_salles.html". Il est recommandé de créer un dossier où vous placerez le fichier "salles.csv" sans les formules du paragraphe précédent, ainsi que les différents programmes Python ci-dessous.

Écrivez le programme suivant et enregistrez-le dans un fichier nommé "salles.py" :

```
1 import folium
2
3 carte_salles = folium.Map(location=[42.97,2.76],zoom_start=10)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     latitude = float(donnees[9])
10    longitude = float(donnees[10])
11    folium.Marker([latitude,longitude]).add_to(carte_salles)
12    ligne = f.readline()
13 carte_salles.save("carte_salles.html")
```

- Exécutez le programme. Quel logiciel permet de visualiser le fichier « carte\_salles.html » qui se trouve dans le même dossier que votre programme ?

N'importe quel navigateur web permet de visualiser la carte, pas besoin d'un logiciel spécifique car il s'agit du code HTML d'une page web généré par Folium.

- Quel est le problème de la carte qui est affichée ? Comment modifier le programme pour qu'il puisse répondre à la question posée ?

La carte n'est pas centrée sur Paris : il faut modifier la ligne 3 comme sur le code ci-dessous.

- Expliquez les lignes 9, 10 et 11.

Ces lignes permettent de lire les coordonnées de latitude et de longitude à partir du fichier CSV, de créer des marqueurs correspondants sur la carte et de les afficher sur la carte Folium.

Carla ne souhaite voir que les salles de cinéma du département 75. Complétez les lignes 11 et 12 pour obtenir une carte similaire à celle ci-dessous.

```
1 import folium
2
3 carte_salles = folium.Map(location=[48.8534,2.3488],zoom_start=12)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     latitude = float(donnees[9])
10    longitude = float(donnees[10])
11    dep = donnees[...]
12    if dep == "...":
13        folium.Marker([latitude,longitude]).add_to(carte_salles)
14    ligne = f.readline()
15 carte_salles.save("carte_salles.html")
```



# Traitement et affichage de données

## Fiche activité élève

### Étape 1 – Traitement de données avec tableur

- **Importation des données ouvertes**

Téléchargez le fichier .csv mis à jour le 6 décembre 2021, puis renommez-le « salles ».

 **Export au format CSV**  
Mis à jour le 6 décembre 2021 — csv — 339 téléchargements

- Quand ce fichier a-t-il été créé ?

.....

- **Les fichiers .csv**

- Qu'est-ce qu'un fichier .csv ?

.....

.....

.....

.....

.....

.....

.....

- Ouvrez le fichier avec **le bloc-notes**. Que contiennent les deux premières lignes ?

.....

.....

- Quel type de séparateur est utilisé dans ce fichier ?

.....

- **Traitement des données**

Importez le fichier **dans un tableur**.

- Quels sont les descripteurs de cette table de données ?

.....

.....

.....

.....

.....

.....

.....

.....

- Charles habite à Carrières-sur-Seine et souhaite connaître le nombre de salles de cinéma de son département. Quelle formule doit-il écrire en B312 ?

.....

.....

.....

- Charles souhaite maintenant connaître le nombre de salles de cinéma à Paris. Quelle formule doit-il écrire en G312 ?

.....

.....

.....

.....

.....

.....

.....

.....

## Étape 2 – Traitement automatique des données

Écrivez le programme suivant :

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 print(descripteurs)
4 ligne = f.readline()
5 print(ligne)
```

- Expliquez les deux lignes qui s'affichent lors de l'exécution du programme.

Écrivez le programme suivant :

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 for i in range(10):
4 ligne = f.readline()
5 print(ligne)
```

- Expliquez la ligne 3 et l'affichage obtenu lors de l'exécution de ce programme.

Écrivez le programme suivant :

```
f = open("salles.csv","r")
2 descripteurs = f.readline()
3 for i in range(10):
4 ligne = f.readline()
5 donnees = ligne.split(";")
6 print(donnees[2])
```

- Expliquez les lignes qui s'affichent lors de l'exécution de ce programme.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- Comment modifier le programme pour qu'il affiche les adresses des salles de cinéma ?

.....

.....

.....

.....

On souhaite étudier toutes les données du fichier. Pour cela, on considère le programme suivant :

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 ligne = f.readline()
4 compteur = 0
5 while ligne != "":
6 donnees = ligne.split(";")
7 print(donnees[3])
8 ligne = f.readline()
9 compteur = compteur + 1
10 print(compteur)
```

- Expliquer l'utilisation de la boucle conditionnelle « while » à la ligne 5.

.....

.....

.....

.....

.....

.....

.....

.....

.....

- Écrivez le programme et expliquez les lignes qui s'affichent lors de son exécution.

.....

.....

.....

.....

- On souhaite calculer le nombre de salles de cinéma dans le 78. Complétez le programme suivant afin qu'il affiche ce nombre.

```
1 f = open("salles.csv","r")
2 descripteurs = f.readline()
3 ligne = f.readline()
4 compteur = 0
5 while ligne != "":
6 donnees = ligne.split(";")
7 print(donnees[3])
8 ligne = f.readline()
9 compteur = compteur + 1
10 print(compteur)
```

### Étape 3 – Géolocalisation des données

Carla souhaite visualiser la position des salles de cinéma sur une carte. Pour ce faire, nous allons créer un programme qui utilise les coordonnées de latitude et de longitude à partir du fichier "salles.csv" pour générer une carte dans le fichier "carte\_salles.html". Il est recommandé de créer un dossier où vous placerez le fichier "salles.csv" sans les formules du paragraphe précédent, ainsi que les différents programmes Python ci-dessous.

Écrivez le programme suivant et enregistrez-le dans un fichier nommé "salles.py" :

```
1 import folium
2
3 carte_salles = folium.Map(location=[42.97,2.76],zoom_start=10)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     latitude = float(donnees[9])
10    longitude = float(donnees[10])
11    folium.Marker([latitude,longitude]).add_to(carte_salles)
12    ligne = f.readline()
13 carte_salles.save("carte_salles.html")
```

- Exécutez le programme. Quel logiciel permet de visualiser le fichier « carte\_salles.html » qui se trouve dans le même dossier que votre programme ?

.....

.....

.....

- Quel est le problème de la carte qui est affichée ? Comment modifier le programme pour qu'il puisse répondre à la question posée ?

.....

.....

.....

- Expliquez les lignes 9, 10 et 11.

.....

.....

.....

.....



Carla ne souhaite voir que les salles de cinéma du département 75. Complétez les lignes 11 et 12 pour obtenir une carte similaire à celle ci-dessous.

```
1 import folium
2
3 carte_salles = folium.Map(location=[48.8534,2.3488],zoom_start=12)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     latitude = float(donnees[9])
10    longitude = float(donnees[10])
11    dep = donnees[...]
12    if dep == "...":
13        folium.Marker([latitude,longitude]).add_to(carte_salles)
14    ligne = f.readline()
15 carte_salles.save("carte_salles.html")
```

