

TP6 : Gestion des fichiers

Compte-rendu de travaux pratiques

Réalisé par :

Boulahrouf Hamza

Dans le cadre du cours de Python

Encadré par : Pr. Ouafae EL BOUHADI

10 novembre 2025

Table des matières

1	Introduction	2
1.1	Objectifs du TP	2
1.2	Contexte	2
2	Exercice 1 : Création et écriture d'un fichier	2
2.1	Énoncé	2
2.2	Solution implémentée	2
2.3	Résultat	2
3	Exercice 2 : Lecture et affichage formaté	2
3.1	Énoncé	2
3.2	Solution implémentée	3
3.3	Résultat	3
4	Exercice 3 : Création d'un dictionnaire géographique	3
4.1	Énoncé	3
4.2	Solution implémentée	3
4.3	Résultat	3
5	Exercice 4 : Calcul d'une distance simplifiée	4
5.1	Énoncé	4
5.2	Solution implémentée	4
5.3	Résultat	4
6	Exercice 5 : Sauvegarde dans un nouveau fichier	4
6.1	Énoncé	4
6.2	Solution implémentée	4
6.3	Résultat	4
7	Exercice 6 : Lecture filtrée	5
7.1	Énoncé	5
7.2	Solution implémentée	5
7.3	Résultat	5
8	Exercice 7 : Résumé général	5
8.1	Énoncé	5
8.2	Solution implémentée	5
8.3	Résultat	6
9	Bilan des compétences acquises	6
9.1	Compétences techniques maîtrisées	6
9.2	Compétences méthodologiques	6
10	Conclusion	6
10.1	Réalisations accomplies	6

1 Introduction

1.1 Objectifs du TP

Ce TP avait pour objectifs principaux :

- Maîtriser la création, lecture et écriture de fichiers en Python
- Manipuler les dictionnaires et structures de données complexes
- Implémenter des algorithmes de calcul géographique simples
- Apprendre à filtrer et analyser des données

1.2 Contexte

Le travail consiste à manipuler des données géographiques de villes marocaines, à calculer des distances approximatives et à générer des rapports statistiques.

2 Exercice 1 : Création et écriture d'un fichier

2.1 Énoncé

Créer le fichier `villes.txt` et y écrire les données géographiques des villes.

2.2 Solution implémentée

```
1 #Créez le fichier villes.txt en mode écriture ("w").
2 fichier_ville=open("villes.txt","w")
3 #Écrivez-y les lignes suivantes :
4 fichier_ville.write("Rabat,34.020882,-6.841650\n")
5 fichier_ville.write("Casablanca,33.573110,-7.589843\n")
6 fichier_ville.write("Fès,34.033333,-5.000000\n")
7 fichier_ville.write("Tanger,35.759465,-5.833954\n")
8 fichier_ville.write("Agadir,30.427755,-9.598107\n")
9
10 fichier_ville.close()
```

2.3 Résultat

Fichier `villes.txt` créé avec succès contenant 5 villes marocaines et leurs coordonnées GPS.

3 Exercice 2 : Lecture et affichage formaté

3.1 Énoncé

Lire le fichier `villes.txt` et afficher les informations de manière formatée.

3.2 Solution implémentée

```
1 #1. Ouvrez le fichier villes.txt en lecture.
2 with open("villes.txt","r") as villes:
3     lv=villes.readlines()
4
5 #2
6 for i in lv:
7     ii=i.split(",")
8     print(f"La ville de {ii[0]} est située à la latitude {ii[1]} et la
9         longitude {ii[2]}")
10
11 #3. Comptez le nombre total de villes dans le fichier.
12 print(f"le nombre total de villes dans le fichier est {len(lv)}")
```

3.3 Résultat

La ville de Rabat est située à la latitude 34.020882 et la longitude -6.841650
La ville de Casablanca est située à la latitude 33.573110 et la longitude -7.589843
La ville de Fès est située à la latitude 34.033333 et la longitude -5.000000
La ville de Tanger est située à la latitude 35.759465 et la longitude -5.833954
La ville de Agadir est située à la latitude 30.427755 et la longitude -9.598107
le nombre total de villes dans le fichier est 5

4 Exercice 3 : Création d'un dictionnaire géographique

4.1 Énoncé

Créer un dictionnaire où la clé est le nom de la ville et la valeur est un tuple (latitude, longitude).

4.2 Solution implémentée

```
1 #1. Lisez le fichier villes.txt.
2 with open("villes.txt","r") as villes:
3     lv=villes.readlines()
4
5 #2. Créez un dictionnaire nommé coordonnees, où la clé est le nom de la
6     ville
7     # et la valeur est un tuple contenant (latitude, longitude).
8 coordonnees={}
9 for i in lv :
10     coordonnees[i.split(",")[0]]=(float(i.split(",")[1]),float(i.split(",")
11         [2]))
12
13 #3. Affichez ce dictionnaire.
14 print(coordonnees)
```

4.3 Résultat

{'Rabat': (34.020882, -6.84165), 'Casablanca': (33.57311, -7.589843),

```
'Fès': (34.033333, -5.0), 'Tanger': (35.759465, -5.833954),
'Agadir': (30.427755, -9.598107)}
```

5 Exercice 4 : Calcul d'une distance simplifiée

5.1 Énoncé

Implémenter une fonction calculant la distance approximative entre deux villes.

5.2 Solution implémentée

```
1 #Écrivez une fonction
2 def distance(ville1, ville2):
3     if ville1 not in coordonnees.keys() or ville2 not in coordonnees.
4     keys():
5         print("une ou les deux ville n'existe pas dans le dict")
6     else:
7         resultat = abs(coordonnees[ville1][0] - coordonnees[ville2][0])
8         + abs(coordonnees[ville1][1] - coordonnees[ville2][1])
9         return resultat
10
11 #Utilisez cette fonction pour afficher la distance entre Rabat et
12 Casablanca.
13 distance("Rabat", "Casablanca")
```

5.3 Résultat

```
1.1959650000000001
```

6 Exercice 5 : Sauvegarde dans un nouveau fichier

6.1 Énoncé

Créer un fichier `distances.txt` contenant les distances de chaque ville depuis Rabat.

6.2 Solution implémentée

```
1 with open("distances.txt", "w") as distancesv:
2     for v in coordonnees.keys() - "Rabat":
3         distancesv.write((f"{v} : {distance('Rabat', v)} \n"))
4
5 with open("distances.txt", "r") as distancesv:
6     print(distancesv.read())
```

6.3 Résultat

```
Rabat : 0.0
Casablanca : 1.1959650000000001
Tanger : 2.7462789999999997
```

Fès : 1.85410099999999982

Agadir : 6.349584

7 Exercice 6 : Lecture filtrée

7.1 Énoncé

Afficher uniquement les villes situées à plus de 2 degrés de Rabat.

7.2 Solution implémentée

```
1 #À partir du fichier distances.txt, affichez uniquement les villes
   situées à plus de 2 degrés de Rabat.
2 with open("distances.txt","r") as distancesv:
3     d=distancesv.readlines()
4 for i in d:
5     ii=i.split(":")
6     if float(ii[1])>2:
7         print(f"{ii[0]} : {ii[1]}")
```

7.3 Résultat

Tanger : 2.74627899999999977

Agadir : 6.349584

8 Exercice 7 : Résumé général

8.1 Énoncé

Comptabiliser les villes par catégorie de distance et afficher un résumé statistique.

8.2 Solution implémentée

```
1 #Comptez combien de villes sont à moins de 2°, entre 2° et 5°, et au-
   delà de 5°.
2 with open("distances.txt","r") as distancesv:
3     d=distancesv.readlines()
4 n1=n2=n3=0
5 for i in d:
6     ii=i.split(":")
7     if float(ii[1])<2:
8         n1+=1
9     if 2<=float(ii[1])<=5:
10        n2+=1
11    if float(ii[1])>5:
12        n3+=1
13
14 #Affichez un petit résumé sous forme de texte.
15 print(f"Le nombre des villes à moins de 2 degree : {n1}, Le nombre des
   villes entre 2 degree et 5 degree : {n2}, Le nombre des villes au-
   delà de 5 degree : {n3}.")
```

8.3 Résultat

Le nombre des villes à moins de 2 degree : 3, Le nombre des villes entre 2 degree et 5
Le nombre des villes au-delà de 5 degree : 1.

9 Bilan des compétences acquises

9.1 Compétences techniques maîtrisées

- **Gestion des fichiers** : Ouverture en modes lecture/écriture, utilisation de `with open()`
- **Manipulation de strings** : Méthodes `split()`, `readlines()`, f-strings
- **Structures de données** : Dictionnaires, tuples, listes
- **Algorithmie** : Calcul de distance, filtrage, comptage

9.2 Compétences méthodologiques

- Analyse et correction d'erreurs syntaxiques
- Validation des données d'entrée
- Génération de rapports statistiques
- Organisation du code en fonctions réutilisables

10 Conclusion

Ce TP a permis de maîtriser les concepts fondamentaux de la gestion des fichiers en Python avec un excellent niveau de réussite :

10.1 Réalisations accomplies

- Création, lecture et écriture de fichiers avec gestion correcte des ressources
- Manipulation avancée des dictionnaires et structures de données complexes
- Implémentation et correction d'algorithmes de calcul
- Analyse statistique et filtrage de données géographiques
- Génération de rapports formatés et exploitation des résultats