

Nom Prénom :

ACTIVITE N°1

≈ 15 min

On a enregistré les données d'un répertoire téléphonique au format CSV et JSON.

```
"nom";"numéro";"adresse"  
'Pierre';'0614122178';'5, rue du fort'  
'Pauline';'0612166114';'18, rue de la poste'  
'Kilian';'0664122432';'2, avenue des oeillets'
```

```
{  
  {  
    'nom':'Pierre',  
    'numéro':'0614122178',  
    'adresse':'5, rue du fort'  
  },  
  {  
    'nom':'Pauline',  
    'numéro':'0612166114',  
    'adresse':'18, rue de la poste'  
  },  
  {  
    'nom':'Kilian',  
    'numéro':'0664122432',  
    'adresse':'2, avenue des oeillets'  
  }  
}
```

1/ Quels sont les descripteurs du répertoire téléphonique ?

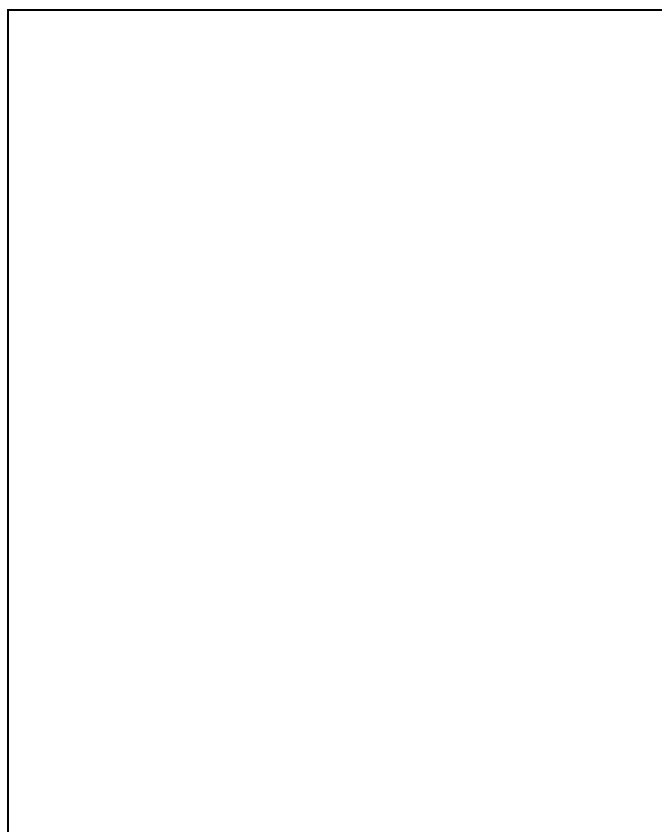
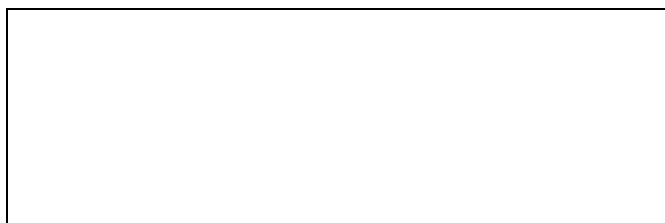
2/ Quelles sont les différentes valeurs du descripteur nom ?

3/ Nous avons récupéré les données suivantes, présentées sous forme de tableau, sur des aliments. Ecrire les fichiers CSV et JSON correspondants.

Aliment	Nutriscore	Calories/100g
gaufre	C	291
Yaourt	A	65
Muesli	A	175
Pain de mie	B	234

Fichier CSV

Fichier JSON



ACTIVITE N°2

≈ 30 min

Sur internet, saisissez l'adresse : <https://data.education.gouv.fr>, puis rendez-vous sur l'onglet **Données** et recherchez et sélectionnez dans la liste les données relatives à la **Réussite au baccalauréat selon l'Age**. On prendra la sous rubrique **Tableau**.

1/ Ces données sont-elles issues de l'Open Data ?

2/ Sur la gauche du site, on peut effectuer des filtrages. En utilisant cette fonctionnalité, donnez le nombre de personnes de 21 ans qui ont eu leur BAC en 1997.

3/ Rendez-vous sous l'onglet **Export** et cliquez sur le jeu de **données Excel** pour télécharger ces données sur votre machine. Puis avec le tableur, analysez et représentez sous la forme graphique adaptée afin de répondre à la question suivante : « **Comment a évolué la réussite au BAC des jeunes de 18 ans au cours de la première décennie du XXI^{ème} siècle ?** »

ACTIVITE N°3

≈ 30 min

En autonomie, reprendre en Python les scripts du cours que l'on appliquera sur les fichiers CSV donnés en Ressources.

En utilisant le fichier « **villes.csv** » donné en ressources, programmer de façon à répondre aux problématiques suivantes :

- 1/ Donnez un tableau contenant toutes les villes de Haute-Loire qui ont une altitude minimum supérieure à 1500 m.
- 2/ Donnez dans un tableau les villes de Haute-Loire qui ont une densité d'habitant inférieure à 50 (dans le tableau ainsi créé, on aura 3 colonnes : le nom de la ville, la densité de la population et l'altitude minimum).
- 3/ Donnez toutes les villes françaises ayant une densité de population supérieure à 50 et une altitude minimum supérieure à 1500 m.
- 4/ Donnez le nombre moyen d'habitants altiligériens en 2012.
- 5/ Triez les données de façon à obtenir par lecture la ville française ayant l'altitude minimum la plus faible.
- 6/ Quelle est la ville française ayant la densité de population la plus forte ?

Nous disposons de 2 fichiers CSV (donnés en Ressources) :

- Un fichier client : « **client.csv** »
- Un fichier commandes : « **commande.csv** »

On souhaite au final obtenir une seule table représentant la fusion de ces 2 fichiers. A vous de jouer. On pourra écrire cette table fusionnée dans un nouveau fichier CSV à l'aide de la fonction Exportation donnée en bas de page. L'ordre a-t-il une importance lors de la fusion ?

- Si vous avez fusionné à partir de client, vous avez peut-être remarqué qu'entre autres, Mme Élodie Gaulin (n° de client 895) est bien présente mais pas de commande correspondante. Pourquoi d'après vous ?
- De la même manière, aucune trace de la commande n° 1324 du 01/02/2017 dans la table fusionnée, pourquoi d'après vous ?
- On remarquera aussi que Mme Neufville a fait 2 commandes. D'où un problème de structurer correctement les données, donnée en dehors des descripteurs.

```
def Exportation(fic_csv, entetes, tbl):  
    f = open(fic_csv, 'w')  
    ligneEntete = ";" .join(entetes) + "\n"  
    f.write(ligneEntete)  
    for valeur in tbl:  
        ligne = ";" .join(valeur) + "\n"  
        f.write(ligne)  
    f.close()
```