OBJECTIFS : L'objectif de ce TP est d'appliquer ce qui a été vu en cours sur le chapitre des fichiers

DOCUMENT A RENDRE : Ce travail est évalué. Vous en rédigerez un compte-rendu et vous le transférerez en fin d'activité **par l'intermédiaire de l'onglet transfert** du site https://nsibranly.fr/ en utilisant le code : **tp10** . Ce compte-rendu contiendra :

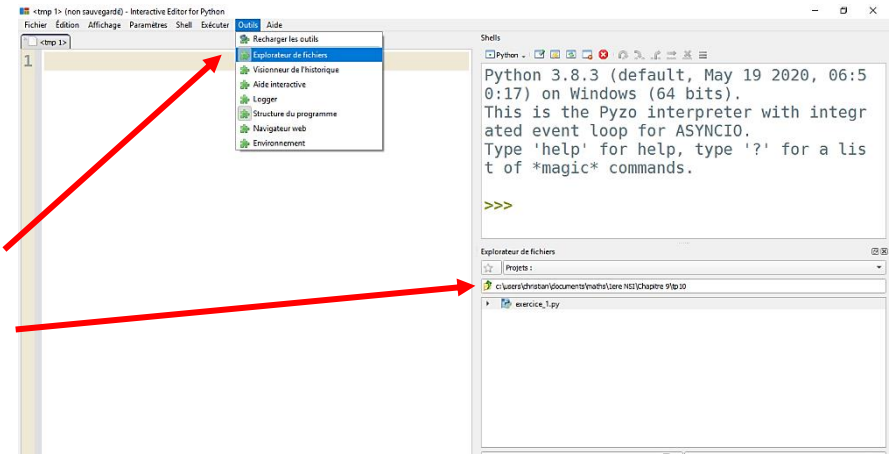
- les réponses aux différentes questions posées,
- les captures d'écran **des morceaux de codes** écrits **et** celles **des résultats des exécutions** données dans le shell. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows.

1. Familiarisation avec les instructions permettant d'écrire dans un fichier texte :

⇨ Ouvrir un nouveau fichier *.py* et le sauvegarder sous le nom *exercice_1.py* .

⇨ Dans Pyzo, dans l'onglet *Outils,* afficher la fenêtre *Explorateur de fichier*. Aller dans répertoire de travail dans lequel se trouve ce fichier *exercice_1.py.*

Ne pas oublier de cocher la case dans l'onglet *Exécuter.*



⇨ Utiliser ce qui a été vu en cours pour écrire dans ce fichier *exercice_1.py,* le code python qui permet de créer un fichier nommé exercice_1.txt et qui contiendra :

Vous pouvez bien sûr modifier les noms proposés.

exercice_1.py | exercice_1.txt

```
1 nom, prenom, note_en_nsi
2 Riondet,Mélusine,5
3 Gottrant,Benjamin,14
4 Boivert,Félix,12
```

exercice_1.py | exercice_1.txt

```
1 f = open("exercice_1.txt","w",encoding="utf-8")
2
3
4
5
6
7 f.close()
```

Permet d'écrire dans le fichier les caractères é è ê ç ù …. (norme utf-8)

2. Familiarisation avec les instructions permettant de lire dans un fichier texte :

⇨ Ouvrir un nouveau fichier *.py* et le sauvegarder sous le nom *exercice_2.py* .

2.1. Utilisation de la fonction read() :

Le code ci-dessous permet d'afficher dans le shell :

```
>>> (executing file "exercice_2.py")
 <class 'str'>

nom, prenom, note_en_nsi
Riondet,Mélusine,5
Gottrant,Benjamin,14
Boivert,Félix,12
```

exercice_2.py | exercice_1.txt

```
1 f = open("exercice_1.txt","r",encoding='utf8')
2
3 f.close()
4
5 print("\n",type(contenu),"\n")
6 print(contenu)
```

⇨ Compléter ce code.

2.2. Utilisation de la fonction readline() :

Le code ci-dessous permet d'afficher dans le shell :

```
>>> (executing file "exercice_2b.py")
nom, prenom, note_en_nsi

Riondet,Mélusine,5

Gottrant,Benjamin,14

Boivert,Félix,12
```

exercice_2b.py | exercice_1.txt

```
1 # Lecture du fichier
2 f = open("exercice_1.txt","r",encoding='utf8')
3 c = f.readline()
4 while c != "" :
5
6
7 f.close()
```

2.3. Stockage des valeurs dans une liste double :

Le code ci-dessous permet d'afficher dans le shell :

```
>>> (executing file "exercice_2c.py")
['nom', ' prenom', ' note_en_nsi\n']
['Riondet', 'Mélusine', '5\n']
['Gottrant', 'Benjamin', '14\n']
['Boivert', 'Félix', '12\n']
```

⇨ Compléter ce code :

exercice_2c.py | exercice_1.txt

```
1  # Lecture du fichier
2  f = open("exercice_1.txt","r",encoding='utf8')
3  lst = []
4  c = f.readline()
5  while c != "" :
6      c = c.split(",")
7
8
9  f.close()
10
11 # Affichage des valeurs lues
12 for l in lst :
13     print(l)
```

2.4. Nettoyage du dernier élément de ces sous-listes :

⇨ Après avoir fait fonctionner le code précédent, exécuter les instructions suivantes dans le *shell*, les unes après les autres et observer le résultat :

```
>>> lst[2]
>>> lst[2][-1]
>>> lst[2][2]
>>> lst[2][2].split()
>>> lst[2][2].split()[0]
>>> lst[2][2][:-1]
```

⇨ Modifier le code précédent en conséquence afin d'obtenir à présent :

```
>>> (executing file "exercice_2d.py")
['nom', ' prenom', ' note_en_nsi']
['Riondet', 'Mélusine', '5']
['Gottrant', 'Benjamin', '14']
['Boivert', 'Félix', '12']
```

2.5. Utilisation de cette liste pour effectuer un calcul de moyenne :

⇨ Ouvrir un nouveau fichier *.py* et le sauvegarder sous le nom *exercice_3.py* .

⇨ Compléter le code suivant qui permet :

- à partir de la fonction *lecture()* de lire le fichier exercice_1.txt,
- à partir de la fonction *moyenne()* de calculer la moyenne des élèves figurant dans le fichier
- à partir de la fonction *eleve_moyen()* de déterminer quel élève a une note qui est la plus proche de la moyenne calculée.

L'exécution donnera dans le shell :

```
>>> (executing file "exercice_3.py")
La moyenne est de :  10.33
Boivert Félix a comme note :  12
```

Le programme principal sera le suivant :

```
# Main
eleves = lecture("exercice_1.txt")
moy = moyenne(eleves)
print("La moyenne est de : ",round(moy,2))
i = eleve_moyen(eleves,moy)
print(eleves[i][0],eleves[i][1],"a comme note : ", eleves[i][2])
```

A vous de définir les différentes fonctions (les écrire les unes après les autres en testant régulièrement avec des *print()* le contenu des variables, listes …) . Ne pas oublier d'indiquer au début de chaque fonction, un commentaire indiquant les entrées / sorties de celle-ci :

```
# Fonctions
def lecture(nom) :
    '''
        Lit le fichier dont le nom est en argument.
        Renvoie une liste comprenant pour chaque ligne
        lue, le nom, le prénom et la note sur 20
    '''
    f = open(nom,"r",encoding='utf8')
    lst = []
    c = f.readline()
    while c != "" :
        c = c.split(",")
        c[2] = c[2][:-1]

    return lst
```

```
def moyenne(liste) :
    '''
        Renvoie la moyenne des notes des élèves
        enregistrés dans la liste qui est en
        argument.
    '''
    
    return moy
```

Remarque : les valeurs lues dans le fichier sont du type « *string* ». Pour réaliser des calculs sur la note qui est donc un string, il s'agit de la convertir en nombre réel. On utilise à cet effet la fonction native de Python float() : `float(liste[i][2])`

```
def eleve_moyen(liste,moy) :
    '''
        Renvoie l'indice de l'élève de la liste
        mise en argument pour lequel la note est la
        plus proche de la valeur moyenne des notes
        mise en argument.
    '''
    
    return indice
```

2.6. Utilisation de cette liste pour effectuer un calcul de moyenne :

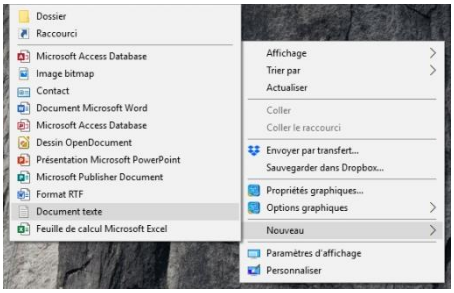
⇨ Télécharger le fichier *world_nsi.txt* proposé sur le site nsibranly.fr . Après l'avoir dézippé, copier ce fichier **dans votre répertoire** de travail.

⇨ Ce code contient 150 000 lignes correspondants à tous les élèves de nsi de notre planète. Jeter un coup d'œil dans ce fichier pour vérifier que les données y sont écrites dans le même format que dans le fichier *exercice_1.txt*. Utiliser votre code pour déterminer la moyenne de ces notes et les nom et prénom de l'élève dont la note est la plus proche de la moyenne.

3. Traitement d'un fichier .csv téléchargé sur un data public. :

⇨ Aller sur le site github : https://github.com/fivethirtyeight/data qui propose des datas U.S. originaux.

Aller dans la catégorie births et copier le fichier US_births_2000-2014_SSA.csv en cliquant sur Raw pour ensuite copier l'ensemble de la page (ctrl A , puis ctrl C) dans un fichier texte que vous nommerez *naissance.txt* .




```
year,month,date_of_month,day_of_week,births
2000,1,1,6,9083
2000,1,2,7,8006
2000,1,3,1,11363
2000,1,4,2,13032
2000,1,5,3,12558
```

Ce fichier contient pour chacune des années entre 2000 et 2014, pour chaque mois de l'année et jour du mois, le nombre de naissance aux U.S.A. On trouve aussi sur chacune des lignes, juste avant le nombre de naissance, le jour de la semaine : 1 pour lundi, 2 pour mardi, …. et 7 pour dimanche.

⇨ Créer un code composé du programme principal suivant :

```
# Main
naissance = lecture("naissances.txt")
n_total = total(naissance)
print("Il y a eu ",n_total," naissances aux U.S. entre 2000 et 2014")
pourcentage = stat(naissance,"mardi",n_total)
print(round(pourcentage,1)," % des naissances, c'était le mardi")
```

A l'exécution il doit donner dans le shell :

```
>>> (executing file "exercice_4.py")
Il y a eu  62176233  naissances aux U.S. entre 2000 et 2014
16.5  % des naissances, c'était le mardi
```

Ce code sera enregistré sous le nom *exercice_4.py* .