

TD1 – Programmation en Python

Notez bien que dans ce TD les exercices se suivent et se complètent.

Exercice 1

Ecrire un programme python permettant de calculer et afficher un IMC à partir de la masse (kg) et la taille (m) saisies par l'utilisateur. Par la suite le programme devra afficher le libellé de la classe d'IMC de cet utilisateur.

Indice de masse corporelle (IMC)	Interprétation (d'après l'OMS)
< 18,5	Insuffisance pondérale (maigreur)
18,5 à 25	Corpulence normale
25 à 30	Surpoids
30 à 35	Obésité modérée
35 à 40	Obésité sévère
> 40	Obésité morbide ou massive

Exercice 2

Ecrire un programme python permettant de simuler une prise de masse de 1% à chaque passage dans un fast-food. Le programme devra demander à l'utilisateur le nombre de passage dans un mois puis afficher le nombre de point d'IMC gagné par rapport à l'IMC calculé dans Exercice 1.

Sachant qu'une heure de sport permet de perdre 0.5% de sa masse et que l'utilisateur ne peut faire que 3h de sport par jour, le programme devra afficher le nombre de jours nécessaire au retour à une corpulence normale (compris entre 18,5 et 25).

Exercice 3

Écrire un programme python qui demande à l'utilisateur de remplir deux listes de 8 mots. Le programme devra compter le nombre de mots en commun aux deux listes. Puis afficher le pourcentage de similitude entre ces deux listes.

Exercice 4

Écrire un programme python qui demande à l'utilisateur de saisir 25 nombres réels positifs, puis qui calcule et affiche la moyenne, la variance ainsi que l'écart-type de ce jeu de données. Dans une première version définissez 3 fonctions et dans une seconde version vous utilisez la fonctionnalité std de numpy.

Exercice 5

Télécharger puis extraire dans un dossier dédié le 78970_2016_10.zip. Ce dernier contient les données de radiosondage (par ballon météo) pour le mois d'octobre 2016. Chaque jour est stocké dans un fichier. Placer les données dans un dossier nommé « data ».

Ouvrir un terminal, puis installer les modules suivants avec `pip install nom_module` :

```
pip install numpy matplotlib windrose openpyxl
```

C'est bon, vous êtes prêts à travailler.

Dans votre code python, utilisez la librairie os pour lister les fichiers de radiosondage présents dans le dossier data :

```
import os
for file in os.listdir("./data/"): #liste les fichiers
    if ('wyo' in file) and file.endswith(".csv"): #teste les extensions
        print(file) #affiche le nom du fichier
```

Dans votre code python, utilisez la librairie numpy pour lire le contenu d'un fichier et le placer dans un tableau :

```
def getDataFile(fname):
    data = np.loadtxt(fname, delimiter=";", dtype=str)
    data=data[1:,:]
    return data.astype(float)
```

Les fichiers csv utilise des « , » au lieu de « . » pour les valeurs décimale. Pour faire le remplacement automatique exécuter la fonction suivante avant de commencer à utiliser les fichiers :

```
def formatFile():
    for file in os.listdir("./data/"): #liste les fichiers
        if ('wyo' in file) and file.endswith(".csv"): #teste les extensions
            with open("./data/"+file, 'r') as File :
                filedata = File.read()
                filedata = filedata.replace(',', '.')
            with open("./data/"+file, 'w') as File1:
                File1.write(filedata)
```

a) Ecrire une fonction python qui calcul et renvoie la moyenne des MIXR de surface (<100m) pour un fichier donné.

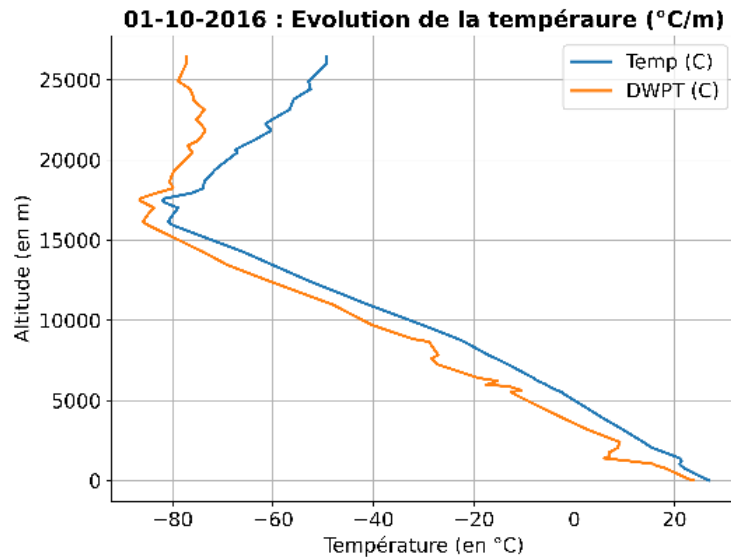
b) Ecrire une fonction qui renvoie le niveau de pression en hPa de passage à des températures négatives (à voir dans Temp (C)) pour un fichier donné.

c) Ecrire une fonction qui renvoie la vitesse maximale en nœud (voir dans SKNT (knot)) pour un fichier donné.

Pour tracer des graphiques, vous devez utiliser la librairie matplotlib, les codes suivants permettent de créer différentes courbes, compléter les pour répondre aux questions, puis intégrer les dans votre rapport.

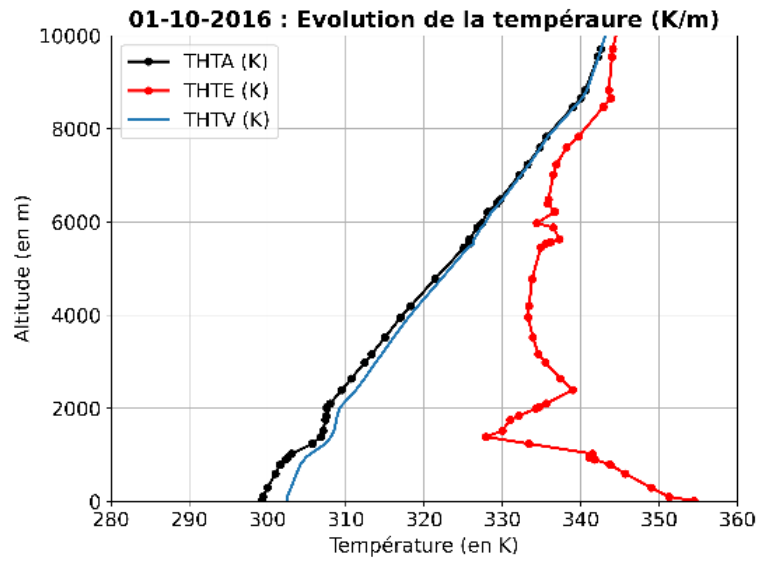
```
##Graphique 1
import matplotlib.pyplot as plt
fs=14                                     #taille du texte
lw=2                                     #épaisseur courbes
plt.rcParams.update({'font.size': fs})   #taille de la police
fig=plt.figure(figsize=(8,6))           #proportion de la figure
plt.plot(x2, y1, label='nom_courbe1', linewidth=lw) #courbe1
plt.plot(x2, y2, label='nom_courbe2', linewidth=lw) #courbe2
plt.xlabel('nom_axe1', fontsize=fs)      #nom de l'axe des abscisses
plt.ylabel('nom_axe2', fontsize=fs)      #nom de l'axe des ordonnées
plt.title('titre_graphique', fontsize=fs+2, fontweight="bold")
plt.grid(True)                          #afficher grille
plt.legend()                            #afficher legende
fig.savefig('nom_img.png', dpi=600)     #enregistrer fichier
plt.show()                              #afficher graphique
```

d) Ecrire un programme python qui permet de reproduire ce graphique, à partir du code donné en exemple mais pour le fichier en date du 21-10-2016.



```
##Graphique 2
import matplotlib.pyplot as plt
fs=14                                     #taille du texte
lw=2                                     #épaisseur courbes
ms=10                                    #taille puce
fig=plt.figure(figsize=(8,6))            #proportion de la figure
plt.plot(x1, y1, '-.', label='nom_courbe1', color='black', linewidth=lw, markersize=ms)
plt.plot(x2, y2, '-.', label='nom_courbe2', color='red', linewidth=lw, markersize=ms)
plt.plot(x3, y3, label='nom_courbe3', linewidth=lw)
plt.xlabel('nom_axe1', fontsize=fs)       #nom de l'axe des abscisses
plt.ylabel('nom_axe2', fontsize=fs)       #nom de l'axe des ordonnées
plt.title('titre', fontsize=fs+2, fontweight="bold")
plt.grid(True)
plt.legend()
plt.xlim((limit_x1,limit_x2))            #limitation de l'axe des abscisses
plt.ylim((limit_y1,limit_y2))            #limitation de l'axe des ordonnées
fig.savefig('nom_fichier.png', dpi=600)  #enregistrer fichier
plt.show()                               #afficher graphique
```

e) Ecrire un programme python qui permet de reproduire ce graphique, à partir du code donné en exemple mais pour le fichier en date du 21-10-2016.



```
##Graphique 3
import matplotlib.pyplot as plt
from windrose import WindroseAxes as wra
fs=14                                #taille du texte
ax = wra.from_ax()                   #création figure rose des vents
fig = plt.gcf()                      #récupération de l'objet figure
ax.bar(x1, x2, bins=[1, 3, 6, 10, 16, 21, 27, 33, 40, 47, 55, 63], normed=True)
plt.grid(True)                       #affichage d'une grille
plt.title("titre", fontsize=fs+6, fontweight="bold")    #définition titre
plt.legend()                         #définition legende
fig.savefig('nom_fichier.png', dpi=600)    #enregistrer fichier
plt.show()                            #afficher graphique
```