PYTHON

C’est quoi Python ?

Un langage de programmation interprété, haut niveau et polyvalent. Ce langage permet d’effectuer de la programmation orientée objet, procédurale et fonctionnelle. Grâce à son vaste écosystème de bibliothèques et de frameworks, il peut être est utilisé dans divers domaines, tels que le développement web, l'analyse de données, l'intelligence artificielle, l'automatisation et la science des données.

Projets réalisés (en autonomie) :

1) Etudier le fonctionnement des fonctions natives de python.

- Moyenne, écart-type, variance d’une liste de données

```
@author: MONDOVY Enzo
"""
listen =[0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,]

def moyenne (L):                                    #calculer la moyenne des éléments d'une liste L
    som=0                                           #affecter à la variable som la valeur 0
    n=len(L)                                        #affecter à la variable n la longueur de la liste L
    for i in range (n):                             #le variant de boucle i prendra les valeurs de chaque éléments de
        som=som+L[i]                                #à chaque itération affecter à som la valeur de som précédente +
        print('ittération =' ,i,' et s = ',som )    #option: afficher le detail des itération
    return som/n                                    #sortie : le resultat de la somme de chaque éléments de la liste
print('Calcul de la moyenne = 210/len(L) = ',moyenne(listen))


print()
def variance (L):                   #permet de savoir si les valeurs d'un liste sont concentrées autour de la moyenne
    n=len(L)                        #affecter à n la longueur de la liste L
    som=0                           #affecter à la variable som la valeur 0
    for i in range (n):             #le variant de boucle i va parcourir tout les indices de la liste
        som= som+L[i]**2            #à chaque itétration affecter à la variable som la valeurs de som précedente + l'
    return (som/L[i])-moyenne(L)**2 #sortie : la formule de la variance = (somme des éléments au carré/nombre d'éléme
print(' calcul de la variance = ',variance(listen))

def ecartype(L):                  #indique à quel point les valeurs d'une liste s'écartent de la moyenne.
    n=len(L)                      #affecter à n la longueur de L
    moy=moyenne(L)                #affecter à moy la valeur de la moyenne de L
    som=0                         #affecter a som la valeur 0
    for i in range (n):           #le variant de boucle i va parcourir tout les indices de la liste
        som=som+(L[i]-moy)**2     #a chaque itération affecter à som la valeur de somme précedente + (l'élément aya
    return (som/n)**0.5           #sortie : racine carré de som/n
```

- Manipulation de données stockées dans une ou plusieurs listes.

```
@author: MONDOVY Enzo
"""
import random as rd                     #import la bibliotèque permettent de générer des valeur aléatoire
liste=rd.sample(range(50),10)       #affecter à liste une liste de 10 valeurs aléatoires comprise entre 0 et 50
print("la liste initiale est : ",liste)                         #afficher la liste initiale
def listemultipliée (L):    #multiplié tout les éléments d'une liste L par 2
    s = []                  #affecter à s une liste vide
    for i in range (len(L)):#le variant de boucle i va parcourir tout les indices la liste
        s.append(L[i]*2)    #à chaque itération multiplié l'élément ayant l'indice i par 2 et le stocké dans la nouve
    return s                #sortie : la nouvelle liste s avec chaque élément multiplié par 2
print("La liste mutipliée par 2 = ",listemultipliée(liste))

def affichageliste(L):      #afficher la liste grace a une fonction (A MODULER)
    s = []                  #affecter à s une liste vide
    for i in range (len(L)):#le variant de boucle i va parcourir toute les indices liste
        s.append(L[i])      #à chaque itération ajouter l'élément ayant l'indice i à la nouvelle liste s
    return s                #sortie : la nouvelle liste s identique à la première
print('La liste initiale = ',affichageliste(liste))


def affichage2 (L):             #parcourir la liste dans le sens inverser et afficher en console chaque itérations
    for i in range (20,0,-1):   #le variant de boucle i parcour les 21 élément de la liste avec un pas inversé
        print('indice',[i])     #afficher les valeurs de i à chaque itération
        print(L[i])             #afficher l'élément correspondant à cette indice
        print(affichage2(liste))

def inverser(L):            #inverser l'ordre des éléments
    new = L[:]              #suppression du lien d'alias, eviter les effets de bord
    n=len(L)                #affecter a une variable la longueur de la liste
    for i in range (n):     #le variant de boucle vas prendre les valeur de chaque indice de la liste
        new[i] = L[-1-i]    #affecter à la liste new l'élément occupant le dernier indice au premier élément de la no
    return new              #sortie : la nouvelle liste inverser
print("la liste inversée est : ",inverser(liste))
```

- Tri à bulle

```
@author: MONDOVY Enzo
"""
#Fonction tri à bulle = ranger un liste de l'ordre croissant ou décroissant
import random as rd                     #import la bibliotèque permettent de générer des valeur aléatoire

Liste=rd.sample(range(1000),1000)         #affecter à "Liste" une liste de 10 nombres aléatoires de compris entre 0 e
print(Liste)
def triabulle(L):
    n=len(L)
    for i in range(n-1,0,-1):           #on parcourt les n-1 premiers éléments de la liste en commencant par le derni
        for j in range(i):              #j allant de 0 au nombre d'élément à comparer
            if L[j]>L[j+1]:             #si l'élément initiale est plus grand que le suivant alors
                L[j],L[j+1]=L[j+1],L[j] #inverser leur place dans la liste

    return L
print(triabulle(Liste))

#_____________________________________________________________________
#Optimisé

def triabulle(L):
    n=len(L)
    for i in range(n-1,0,-1):           #on parcourt les n-1 premiers éléments de la liste en commencant par le derni
        tab_trie=True
        for j in range(i):              #j allant de 0 au nombre d'élément à comparer
            if L[j]>L[j+1]:             #si l'élément initiale est plus grand que le suivant alors
                L[j],L[j+1]=L[j+1],L[j] #inverser leur place dans la liste
                tab_trie=False
            if tab_trie==True:
                return L

    return L
print(triabulle(Liste))
```

- Tracer et manipuler les fonctions mathématiques (bibliothèques : MATPLOTLIB, Pylab)

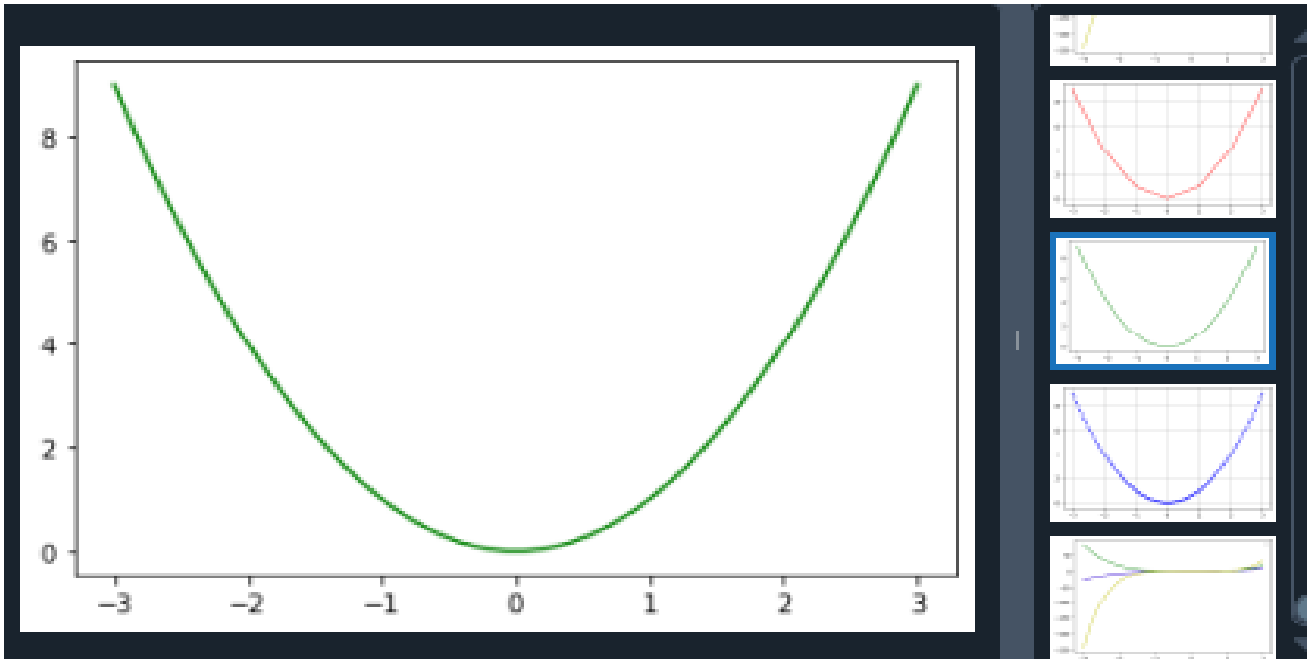
```
@author: MONDOVY Enzo"""
from pylab import *             #importer la bibliotèque pylab permettant de faire des tracés graphiques
#Affichage de la fonction x**2
def f(x):                       #definir la fonction prennant en argument x
    return x**2                 #sortie renvoie x**2
x=[i for i in range(-3,4)]      #abcisse de i allant -3 à 3
y=[f(u) for u in x]             #ordoné f(u) dans l'intervalle de x
plot(x,y,'r')                   #tracer de la courbe en respectant l'ordonné en fonction de l'abcisse, en rouge
grid()                          #affichage du cadrillage
show()                          #affichage du graphique dans l'onglet "plots"
#_____________________________________________________________________
#Affichage de la fonction x**2  (beaucoup plus simple grace à linspace)
x=linspace(-3,3)                #linspace permet de creer un liste d'abcisses (debut,fin)
y=x**2
plot(x,y,'g')
show()
#_____________________________________________________________________
#Formation d'un nuage de points
x=linspace(-3,3,100)            #linspace permet de creer un liste d'abcisses (debut,fin,nombre de point)
y=x**2                          #y=ordonné en fonction de x
plot(x,y,'b.')                  #'b.' affiche des points(.) en rouge('r')
grid()
show()
#_____________________________________________________________________
#Afficher plusieurs courbes sur le même graphe
x=linspace(-3,2)
y=x**3
y2=x**2
y3=x**4
y4=x**5
legend('abcisses','ordonnés')
plot(x,y,'r')
plot(x,y,'b')
plot(x,y3,'g')
plot(x,y4,'y')
```

```
from matplotlib import pyplot as plt

forme = [[255,255,255],[0,0,255]]
plt.imshow(forme, cmap="gray")
plt.show()

from copy import deepcopy

def fliph(image):
    nL=len(image)       #nombre de lignes dans l'image
    nC=len(image[0])    #nombre de colonnes dans l'image
    new=deepcopy(image) #copie dsans lien d'allias
    for i in range(nL):
        for j in range(nC):
            new[i][j]=image[i][-1-j]
    return new
im=fliph(forme)
plt.imshow(im, cmap="gray")
plt.show()
```



- Algorithme de recherche de données :

Rechercher un caractère dans un liste :

```
@author: MONDOVY Enzo
"""
car = input('veuillez un caractère : ')
txt = "abcdef"
def research(car,txt):
    assert type(car)== str, "Le 1er argument de rechercher  doit être un str"
    assert len(car)== 1, "Le 1er argument de rechercher doit être un seul caractère"
    n=len(txt)
    ls=[]
    for i in range(n):
        if car==txt[i]:
            ls.append(i)
    return ls
print(research(car,txt))
```

Rechercher le maximum d’une liste :

```
@author: MONDOVY Enzo
"""
Liste=[1,3,67,0,5,]     #affecter la liste à une variable
def maximum(L):         #definir la fonction max
    n=len(L)            #n prend la valeur de la longueur de la liste
    s=L[0]              #s correpond au premier élément de la liste
    for i in range (n): #boucle inconditionnel qui vas traverser toute la liste
        if s < L[i]:    #si s est plus petit que l'élément suivant alors
            s=L[i]      #l'élement suivant devient s et on répete jusqu'a ce qu'on ait traverser toute la liste
    return s            #s=le max de la liste
print(maximum(Liste))   #affichage de la liste
```

- Jeux (boucles conditionnelles/saisis utilisateurs).
- Création de scénarios

```
@author: MONDOVY Enzo
"""
#Pierre feuille papier ciseaux
import random
manches = int(input("Combien de manchez voulez-vous jouer ?"))
score_joueur=0
score_ordi=0
options=["pierre","feuille","ciseaux"]
while score_joueur < manches and score_ordi < manches:
    choix_joueur=input('Que jouez-vous ?')
    while choix_joueur not in options:
        print("Saisie invalide")
        break
    choix_ordi= random.choice(options)
    if choix_joueur == choix_ordi:
        print("egalité ")
    elif choix_joueur == "pierre" and choix_ordi == "ciseaux"\
        or choix_joueur == "ciseaux" and choix_ordi == "feuille"\
        or choix_joueur == "feuille" and choix_ordi == "pierre":
            print("Vous avez gagné la manche ",choix_joueur," bat ",choix_ordi,".")
            score_joueur+=1
if score_joueur==manches:
    print("Vous avez gagner!")
else:
    print("Vous avez perdu")
```

```
@author: MONDOVY Enzo
"""
#Le juste prix
from random import*
justeprix=randint(1,10)
score=3
tentative=0
print("Devinez un chiffre inscrit entre 1 et 10")
while True and score>0 :
    nombre=int(input())
    tentative+=1
    score-=1
    if nombre>justeprix:
        print("Le prix est trop haut")
    if justeprix>nombre:
        print("Le prix est trop bas")
    if nombre==justeprix:
        print("Vous avez trouvé les juste prix en ",tentative," tentatives avec un score de ",score," points !")
        break
print("Partie terminé le resultat est ",justeprix)
```