

Compte - Rendu

Compte Rendu des Ateliers 1 et 2 Du module de Python

Lieu : El Mahjoub

Date : Jaber

Filière et année : LST-GI 2021-2022

Atelier 1 :

Cet Atelier est basé sur le travail avec les fonctions en Python.

Exercice 1 :

```
# Exercice1

def fact(i):          # Définir la fonction factoriel
    if i==0:
        return 1
    else:
        return (i*(fact(i-1)))

lim=int(input("Enter number of terms: "))
sum=0
for i in range(1,lim+1):    # Boucle qui compte la somme de chaque partie et les partie precedente.
    sum=sum+fact(i)/i

print("the sum of the serie is: ",sum) #Affichage du resultat
```

```
Enter number of terms: 10
the sum of the serie is: 409114.0
```

```
Process finished with exit code 0
```

Exercice2 :

```
# exercice2

# programme qui converti un nombre du decimal en binaire

def conv(x):
    if x>1:          # condition pour prendre les elements different de 0 et 1
        conv(x//2)
    print(x % 2,end=" ") #affichage du reste de la division pour avoir 0 ou 1

x=int(input("entrer la valeur de x:")) #affichage du resultat
conv(x)
```

```
entrer la valeur de x:75
```

```
1 0 0 1 0 1 1
```

```
Process finished with exit code 0
```

Exercice 3 :

```
#_exercice 3:

def sum(x):
    if x==1:          # condition d'arret pour la fonction recursive elle retourne 0 ou 1
        return 1
    elif x<=0:
        print("entrer un nombre positive")_# demander de l'utilisateur de taper une valeur positive
    else:
        return (x+sum(x-1))          #Retun la somme des entiers precedents

x=int(input("entrer un nombre positive: "))
print(sum(x))
```

```
entrer un nombre positive: 8
```

```
36
```

```
entrer un nombre positive: -1
```

```
entrer un nombre positive
```

```
None
```

Exercice 4 :

```
# Exercice 4:

def fibo(n):
    if n==0:          # condition d'arrêt pour la fonction fibonacci
        return 0
    elif n==1:
        return 1
    else:
        return fibo(n-2)+fibo(n-1) # L'appelle à nouveau de la fonction fibo

a=int(input("Veuillez entrer un nombre de 1 à n: "))
print("Votre suite fibonacci est:")
for i in range(a):
    print(fibo(i),end=" ")      # Boucle pour parcourir les elements precedents
                                # Affichage de resultat
```

```
Veuillez entrer un nombre de 1 à n: 15
Votre suite fibonacci est:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
Process finished with exit code 0
```

Exercice 5 :

```
# Exercice 5:

def combien(x):
    if x//10==0:      # on fait la division par 10 puis on incremente 1 chaque fois
        return 1
    else:
        return 1+combien(x/10)

n=int(input("Entrez votre nombre:"))
print("votre chiffre contient:",combien(n),"chiffres")
```

```
Entrez votre nombre:2021
votre nombre contient: 4 chiffres

Process finished with exit code 0
```

Exercice 6 :

```
# exercice 6:
"""
Programme qui compare deux elements adjacents et les inversent dans le sens croissant en utilisant deux boucles qui parcourent la liste
"""

def bull_tri(x):
    for i in range(len(x)):
        for j in range(len(x)-i-1):
            if x[j]>=x[j+1]:
                r=x[j]
                x[j]=x[j+1]
                x[j+1]=r
        print(x)
x=[98, 22, 1, 89, 4, 75, 0, 2]
bull_tri(x)
```

```
[0, 1, 2, 4, 22, 75, 89, 98]
```

```
Process finished with exit code 0
```

Exercice 7 :

```
# exercice 7:

def rev(a):
    b=[]                                # creer une liste vide.
    for i in range(len(a)):
        b.append(a[-i-1])              # on ajoute chaque fois les caracteres dans la liste B
        b[i]=a[-i-1]                  # on remplace les elements de la liste a dans b

    print(b[i],end=" ")

a=str(input("Enter votre chaine de caractere: "))
rev(a)
```

```
Enter votre chaine de caractere: Python
n o h t y P
Process finished with exit code 0
|
```

Exercice 8 :

```
# exercice 8:

def freq(x,n):
    return x.count(n)                # on retourne le count du chaine de caractere et on le stocke dans x
x=str(input("entrer votre chaine de caractere: "))
n=str(input("entrer votre caractere:"))
print(freq(x,n))
```

```
entrer votre chaine de caractere: programing
entrer votre caractere:r
2
```

Exercice 9 :

```
# exercice 9:

def trouver(e, m):
    for i in range(len(m)):           # boucle i pour parcourir les index i dans len(matrice)
        for j in range(len(m[i])):   # boucle j pour parcourir les index j
            if m[i][j] == e:          # condition de trouver l'element.
                return (i,j)          #return les index de e.

m=[[3,27],[-1,15],[1,2]]              #exemple de matrice donnee.
print(trouver(-1,m))
```

```
(1, 0)
```

```
Process finished with exit code 0
```

Atelier 2 :

Dans cet atelier on va travailler sur les listes, les sets et les dictionnaires :

Exercice 1 :

```
#Exercice 1:

def afficher (l1,l2):
    l3=[]
    for i in l1:
        if l1.index(i)%2 ==0:
            l3.append(i)
    for i in l2:
        if l2.index(i)%2 !=0:
            l3.append(i)
    print(l3)

l1=[4,8,12,16,20,24,28]
l2=[3,6,9,12,15,18,21]

afficher(l1,l2)
```

```
[4, 12, 20, 28, 6, 12, 18]
```

```
Process finished with exit code 0
```

Exercice 2 :

```
def dev_inv(liste):
    v=len(liste)//3          # on devise la longueur de la liste par 3
    s=v*2
    l1=liste[v:]             #la liste l1 contient les 3 premiers elements
    l2=liste[v:s]            #la liste l2 contient les 3 suivants elements
    l3=liste[s:]             ##la liste l13 contient les 3 derniers elements
    print(l1[::-1],l2[::-1],l3[::-1])  #on affiche les 3 listes reversed

l1=[1,2,3,4,5,6,7,8,9]
dev_inv(l1)
```

```
[3, 2, 1] [6, 5, 4] [9, 8, 7]
```

```
|
```

```
Process finished with exit code 0
```

Exercice 3 :

```
#Exercice 3:

l=[11, 45, 8, 11, 23, 45, 23, 45, 89]
d={}
for i in l:
    if l.count(i)>1: # condition pour savoir les elements qui se repete plus de fois
        d.update({i:l.count(i)})
        del(i)      # on supprime l element pour ne pas le repeter
    else:
        d.update({i:l.count(i)}) # on ajoute un element au dictionnaire
print(d,end=" ")
```

```
{11: 2, 45: 3, 8: 1, 23: 2, 89: 1}
Process finished with exit code 0
```

Exercice 4 :

```
#Exercice 4:

#Dans ce programme on a utilise la fonction intersection du set pour trouver set1 inter sets et puis faire la difference entre eux pour avoir la nouvelle set3
set1={23,42,65,57,78,83,29}
set2={57,83,29,67,73,43,48}
t=set1.intersection(set2)
set3=(set1-t)
print("L'intersection du deux sets est:" t)
print("La nouvelle set est: " set3)
```

```
L'intersection du deux sets est: {57, 83, 29}
La nouvelle set est: {65, 42, 78, 23}

Process finished with exit code 0
```

Exercice 5 :

```
#exercice 5:
a=[47,64,69,37,76,83,95,97]
d={'Khadija':47,'Imane':69,'jaber':76,'Abir':97}
b=[]
for value in d.values(): #boucle for pour parcourir les value du dictionnaire
    for i in a:
        if i==value: #condition pour tester
            b.append(value) #ajouter les valeurs dans b
print(b,end=" ")
```

```
[47, 69, 76, 97]
Process finished with exit code 0
|
```