Exercice Python Liste

Python Cheat Sheet

1 B A BA

1. Dans une liste stocker les 101 premiers entiers naturels
2. Dupliquer cette liste (et pas son adresse)

Dans une fonction :

1. Supprimer l'élément du milieu de la liste dupliqué
   Dans une autre fonction :
2. Mélanger la liste principale
   Dans une autre fonction :
3. Trouver à quel indice se trouve l'élément supprimé de la seconde liste

Correction

```
liste1 = list(range(101))
liste2 = liste1[:]

def middle():
        liste2.pop(int(len(liste2)/2))

middle()

## Une façon de faire
def melanger():
    liste_melange=[]
    for i in range(len(liste2)):
        a=random.randint(0,len(liste2)-1)
        liste_melange.append(liste2[a])
        liste2.pop(a)
    return liste_melange

liste1=melanger()
## Une autre façon de le faire
def shuffle():
        for i in range(500):
                a=random.randint(0,len(liste1)-1)
                liste1.append(liste1[a])
                liste1.pop(a)
shuffle()

def whereIsBrian(listeComplete,listeIncomplete):
    for i in range(len(listeComplete)):
        isIn = False
        for j in listeIncomplete:
            if j==listeComplete[i]:
                isIn = True
        if not isIn:
            return i
```

```
chiffre = whereIsBrian(liste1,liste2)
```

2 Listing

Écrire une fonction pour générer des mails de façon aléatoire. Pour celà :

1. Trouver un moyen d'écrire un nom de domaine de façon aléatoire dans une liste prédéfinie (gmail, yahoo, wanadoo, hotmail, caramail, etc.) avec une terminologie aleatoire (.fr, .com, .ru, .net, .li, .eu, etc.)

```
def nomDeDomaine():
        ##Fonction à écrire
```

2. Trouver un moyen d'écrire un mot de façon aléatoire dans une fonction avec comme parametre la taille du mot. (L'idéal étant d'alterner consonne et voyelle)

```
def motAleatoire(taille):
        ##Écrire la fonction pour un mot aléatoire
```

3. Combiner les deux pour écrire une adresse mail de façon aléatoire, cette adresse mail devra comporter 4 parties :

```
[nom_aleatoire],[@],[nom_de_domaine][terminologie]
```

4. Générer une liste de 100 adresses mails aléatoires

Correction

```
import random

def nomDeDomaine():
        liste_domaine=["gmail", "yahoo", "wanadoo", "hotmail", "caramail"]
        liste_pays = [".fr", ".com", ".ru", ".net", ".li", ".eu"]
        a=random.randint(0,len(liste_domaine)-1)#ici j'utilise les indices
        b=random.choice(liste_pays)#ici je random sur les valeurs
        return liste_domaine[a]+b
```

```
import string

def motAleatoire(taille):
        mot=""
        liste_voyelle = ["a","e","u","i","o","y"]
        liste_consonne = string.ascii_lowercase # j'initialise ma liste de consonne
        for i in liste_consonne:#j'enleve les voyelles des consonnes
                if i is in liste_voyelle:
                        liste_consonne.remove(i)
        voyant = False #Je met en place un interrupteur pour alterner entre voyelle et
consonne.
        for i in range(taille):
                a = random.choice(liste_voyelle)
                b = random.choice(liste_consonne)
                if voyant:
                        mot+=a
                else:
                        mot+=b
                voyant = not voyant
```

```
        return mot
```

```
liste_mail = []
for i in range(100):
        taille = random.randint(2,12)
        liste_mail.append(motAleatoire(taille)+nomDeDomaine())
```

3 Fibonacciception

1. Dans une liste stocker les 100 premiers éléments de la suite de Fibonacci.
2. Créer une nouvelle liste de taille 98 où vous additionnez les éléments de la liste de Fibonacci, à la manière de la suite de Fibonacci

```
liste_fibo = ##suite de fibonacci
liste_fibo_fibo = ##suite de fibonacci à partir de la suite de fibonacci
## la liste_fibo_fibo est construit comme suivant :
liste_fibo_fibo[n]=liste_fibo[n-1]+liste_fibo[n+1] ## la liste devra être plus petite de deux éléments (un avant, un après) pour fonctionner
```

3. Duppliquez la liste liste_fibo_fibo, et supprimez à l'intérieur tous les nombres présent aussi dans la liste liste_fibo.
4. Additionnez tous les nombres restants.

Correction

```
liste_fibo = [0,1]#j'initialise la liste de fibonacci
#Je construit la liste
for i in range(2,100):
        liste_fibo.append(liste_fibo[i-2]+liste_fibo[i-1])
liste_fibo_fibo=[]
#Je construit la fibo fibo
for i in range(1,99):
        liste_fibo_fibo.append(liste_fibo[i-1]+liste_fibo[i+1])
#Je duplique la liste
liste_fibo_fibo2 = liste_fibo_fibo[:]
#Je supprime les éléments qui se retrouvent dans les deux
for i in range(len(liste_fibo_fibo2)):
        if liste_fibo_fibo2[i] in liste_fibo:
                liste_fibo_fibo2.pop(i)
#J'additionne
somme=0
for i in liste_fibo_fibo2:
        somme+=i
```

4 Une année bien chargée ?

Dans une année il existe 365 jours, vous allez créer une liste de cette taille où vous pourrez stocker des évènements dans chacune des cases.
Une journée est composées de 3 moments clefs, le matin, l'après midi, et le soir. Vous pourrez construire une journée en ajoutant ces trois moments clefs sous forme de liste, dans chacune des cases de votre calendrier.

1. Construire une fonction qui créer une journée (matin, après midi et soir) et qui retourne un tableau de valeur avec les activités choisies :

1. Se reposer
2. Se distraire
3. Travailler

```
def journee(matin,apres_midi,soir):
        return ##retourne les activités que vous avez choisi de faire sous forme de tableau
```

Vous pouvez décider de mettre en paramètre directement le chiffre correspondant à l'activité, sinon vous pourrez coder à l'intérieur de la fonction un interpretteur en fonction de ce que vous allez rentrer à la main.

Lorsque vous créer des fonctions semaines, mois, et années, concatennez votre liste principale de façon à ce que cette liste ai une taille de 365 à la fin de l'année

2. Construire une fonction pour créer une semaine (un tableau composé de 7 jours). Vous pouvez décider de réaliser des semaines équilibrées (1) ou de repos (2), ou de travail(3). Une semaine équilibré aura une répartition équilibré des différentes taches par jour, une semaine de repos ressemble à une semaine de vacances, une semaine de travaille est une semaine de rush, avec de fort temps de travail.
3. Construire une fonction de la même manière qui pourra créer des mois (un tableau composé de 4 semaines)
4. Et enfin construire une fonction qui permettra de créer une année (un tableau composé de 12 mois).
5. Simulons notre année en parcourrant notre année jour par jour.

```
def simulation():
        travail = 10
        repos = 10
        joie = 10
        for i in annee:
                ## Si dans notre journée nous travaillons, nous gagnons 2 point de travail
mais perdons 2 points de repos et 1 de joie
                ## Si dans notre journée nous nous distrayons nous gagnons 2 point de joie
mais perdons 1 point de travail et 1 de repos
                ## Si dans notre journée nous nous reposons, nous gagnons 3 points de repos
mais perdons 1 points de travail
```

6. Si pendant notre simulation nos points de joie deviennent inférieur à 0, nous devenons triste.
   Alors nous ne pouvons plus gagner plus de points de travail.
   Si pendant notre simulation nos points de repos deviennent inférieur à 0 nous devenons fatigué.
   Alors nous ne pouvons plus gagner de points de travail ni de joie.
   Si pendant notre simulation nos points de travail deviennent inférieur à 0 nous devenons chômeur.
   Alors nous ne pouvons plus gagner de point de joie.
7. Quel est votre meilleur score à la fin de l'année ?
8. Comment optimiser notre année en se basant sur la simulation, *i.e.* créer notre calendrier en même temps que la simulation.

Correction

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 31 09:14:54 2022

@author: nicolas
"""
import random
```

```
def journee():
    a=random.randint(1,3)
    b=random.randint(1,3)
    c=random.randint(1,3)
    return list([a,b,c])

def semaine():
    liste_semaine=[]
    for i in range(7):
        liste_semaine.append(journee())
    return liste_semaine

def mois():
    liste_mois = []
    for i in range(4):
        liste_mois.append(semaine())
    return liste_mois

def annee():
    liste_annee = []
    for i in range(12):
        liste_annee.append(mois())
    return liste_annee

annee_aleatoire = annee()

def simulation(annee):
    travail = 10
    repos = 10
    joie = 10
    for m in annee:
        for s in m:
            for i in s:
                for j in i:
                    if j==1:
                        if repos<0:
                            print("fatigué")
                        elif joie<0:
                            print("triste")
                        else:
                            travail+=2
                        repos-=2
                        joie-=1
                    if j==2:
                        if travail <0:
                            print("chomeur")
                        elif repos<0:
                            print("fatigué")
                        else:
                            joie+=2
                        repos-=1
                        travail-=1

                    if j==3:
                        repos+=3
                        travail-=1
    return(travail,repos,joie)

test = simulation(annee_aleatoire)
```

L'optimisation peut se faire de plein de façon. Ici on va simplifier le problème en disant qu'une année est composée de 365 jour. Et nous allons créer une journée au fil de la simulation.

```
def minimumIndice(liste):
    mini=liste[0]
    id_mini=0
    for i in range(len(liste)):
        if liste[i]<mini:
            mini=liste[i]
            id_mini=i
    return id_mini

def simulationOptimisee():
    travail = 10
    repos = 10
    joie = 10
    annee=[]
    ##Premier jour aléatoire, pas obligatoire
    annee.append(journee())
    for a in range(365):
        journee_suivante=[]
        for j in annee[a]:
            if j==1:
                repos+=3
                travail-=1

            if j==2:
                if travail <0:
                    print("chomeur")
                elif repos<0:
                    print("fatigué")
                else:
                    joie+=2
                repos-=1
                travail-=1

            if j==3:
                if repos<0:
                    print("fatigué")
                elif joie<0:
                    print("triste")
                else:
                    travail+=2
                repos-=2
                joie-=1
            score = [repos,joie,travail]
            activite_suivante = minimumIndice(score)+1
            journee_suivante.append(activite_suivante)
        print(journee_suivante)
        print(score)
        annee.append(journee_suivante)

    return score

test2=simulationOptimisee()
```

5 La pagie de Moudlard

Vous êtes un magicien en herbe et lancez des sorts à vos amis. Vous êtes farceur et votre sort favori est le Mutismum. Ce sort permet de faire begayer quelqu'un, et chez vous... celà se traduit par une modification des lettres dans les phrases, par exemple les ***p*** deviennent des ***m*** et vis versa. Créez ce sort sous python :

```
def mutismum(phrase):
        ###ecrire le fonctionnement

print(mutismum(input("Vous vouliez dire ? ")))
```

Correction

```
def mutismum(phrase):
        phrase=list(phrase) #Je transforme ma phrase en liste
        for i in range(len(phrase)):#Je boucle sur mes indices
                if phrase[i]=="m":#Si j'ai un m
                        phrase[i]="p"#Alors je le remplace par un p, si je bouclais sur les
valeurs cela ne marcherait pas
                elif phrase[i]=="p":
                        phrase[i]="m"
        return("".join(phrase))#Ici je transforme ma liste en une phrase à nouveau

print(mutismum(input("Vous vouliez dire ? ")))
```

6 Trier, trier

1. Construire la listes des 100 premiers nombres naturels qu'on appellera liste1.
2. Mélanger cette liste de façon aléatoire dans une autre liste qu'on appellera liste_melangee.
3. Ecrire une méthode pour trouver le maximum et son indice dans la liste mélangée, garder le maximum en mémoire dans une variable, son indice dans une autre variable.
4. Créer une liste vide qu'on appellera liste_triee
5. Supprimer ce maximum, en même temps l'ajouter à la nouvelle liste.
6. Recommencer les étapes 3,4,5 sur toutes les valeurs de la liste_melangee.
7. Quel liste obtenons nous à la fin ?

Recommencer toutes les étapes en cherchant le minimum plutot que le maximum.

Bravo, vous avez effectuer votre premier tri.

Correction

```
import random

liste1 = list(range(100))
liste_melangee = liste1[:]
random.shuffle(liste_melangee)

def maximum(liste):
    maxi=liste[0]
    id_maxi = 0
    for i in range(len(liste)):
        if liste[i]>maxi:
            maxi=liste[i]
            id_maxi=i
    return maxi,id_maxi

maxi_liste_melangee, id_maxi_liste_melangee =maximum(liste_melangee)
```

```
liste_triee=[]

for i in range(len(liste_melangee)):
    maximum_courant,id_maximum_courant=maximum(liste_melangee)
    liste_triee.append(maximum_courant)
    liste_melangee.pop(id_maximum_courant)
```

7

Tableau à double entrée

Comment écrire un tableau avec deux valeurs par cases : par exemple garder en mémoire la trajectoire d'une balle, avec son *x* et son *y*.

```
x = list(range(0,100))
y = []
for i in x:
        y.append(-((i)**2)-2*i+1000)
```

Transformer ces deux listes en une liste à double entrée.

Correction

```
z=[x,y]
## Ainsi
z[0][0]
## Me permet d'accéder à la première valeur du premier tableau
```

8

Un tableau de multiplication

Construire un tableau de multiplication en utilisant un tableau de tableau. Par exemple :

```
table_multiplication = #notre tableau

##Pour trouver le résultat de 3*4 il suffira d'écrire :
tablea_multiplication[3][4]
```

Correction

```
table_multiplication=[]
for i in range(10):
    colonne=[]
    for j in range(10):
        colonne.append(i*j)
    table_multiplication.append(colonne)

print(table_multiplication[3][4])
```

9

Opérations élémentaires de matrices

Soient deux matrices :

```
A = [list(range(100)),list(range(100,0,-1))]
B = [list(range(100,200)),list(range(200,100,-1))]
```

Comment est-ce que je peux additionner mes deux matrices ? À savoir, additionner chacun des éléments par rapport à leur indice. Vous pouvez créer une fonction pour ça.

Comment multiplier les matrices ? **Attention la multiplication de matrice est plus complexe que l'addition** Créer une fonction pour ça.

Correction

L'addition de deux matrices se fait comme suivant :
https://fr.wikipedia.org/wiki/Addition_matricielle

```
A = [list(range(2)),list(range(3,5))]
B = [list(range(3,5)),list(range(2))]

def addition(A,B):
    if len(A)!=len(B) or len(A[0])!=len(B[0]):
        print("Ces matrices ne sont pas aditionnables")
        return False
    C = len(A)*[len(A[0])*[0]]
    for line in range(len(A)):
        for row in range(len(A[0])):
            C[line][row]=A[line][row]+B[line][row]

    return C

C = addition(A,B)
```

La multiplication est un peu plus complexe : https://fr.wikipedia.org/wiki/Produit_matriciel

```
def multiplication(A,B):
    if len(A)!=len(B[0]) or len(A[0])!=len(B):
        print("Ces matrices ne sont pas multipliables")
        return False
    C = len(A)*[len(B[0])*[0]]
    for line in range(len(A)):
        for row in range(len(A[0])):
            for elt in range(len(A)):
                C[line][row]+=A[line][elt]+B[elt][row]

    return C

D = multiplication(A, B)
```