

Exercices sur les listes, dictionnaires et fichiers

December 9, 2024

1 Manipulation de listes

1. Créer une liste nommée `nombres` contenant les entiers de 0 à 9 inclus
2. Créer une liste nommée `racines_nombres` contenant la racine carré des éléments de `nombres`
3. Créer une liste nommée `nombres_pairs` contenant tous les nombres pairs de la listes `nombres`
4. Vérifier que l'on a bien $y = x^2$ en prenant a fur et à mesure y dans `nombres` et x dans `racines_nombres`

```
[1]: from math import sqrt, isclose
nombres = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
nombres = list(range(10))

racines_nombres = [sqrt(i) for i in nombres]
racines_nombres

nombres_pairs = []
for i in nombres:
    if i%2==0:
        nombres_pairs.append(i)

print(nombres_pairs)

for i in range(len(nombres)):
    if isclose(nombres[i], racines_nombres[i]**2):
        print(True)
    else:
        print(False)
```

```
[0, 2, 4, 6, 8]
```

```
True
```

```
True
```

```
True
```

```
True
```

```
True
```

```
True
```

```
True
```

```
True
```

True
True

2 Annuaire

Nous allons créer un petit dictionnaire qui contient des informations sur un étudiant.

1. Créer un dictionnaire `etudiant` contenant les clés et valeurs suivantes :
 - 'nom': 'Jean Dupont'
 - 'âge': 20
 - 'filière': 'Informatique'
2. Modifier l'âge pour qu'il soit égal à 21
3. Afficher le genre de l'étudiant si le dictionnaire possède une telle clé sinon afficher un message indiquant que l'on ne connaît pas son genre.

```
[2]: etudiant = {'nom': 'Jean Dupont', 'age': 20, 'filiere': 'Informatique'}

etudiant['age'] = 21

def avec_cle_genre(etudiant):
    for key in etudiant:
        if key == 'genre':
            return True
    return False

avec_cle_genre(etudiant)

if 'genre' in etudiant:
    print(etudiant['genre'])
else:
    print('Pas de clé genre')

etudiant.get('genre', 'Genre non défini')
```

Pas de clé genre

```
[2]: 'Genre non défini'
```

3 Moyennes des notes

Soit le dictionnaire suivant donnant les notes d'un étudiant. Calculez la moyenne de ses notes de deux manières différentes. Calculez à nouveau la moyenne sans la note de biologie.

```
[3]: dico_notes = {
    "math": 14, "programmation": 12,
    "anglais": 16, "biologie": 10,
```

```
"sport": 19
}
```

```
[4]: liste_notes = list(dico_notes.values())
moyenne = sum(liste_notes)/len(liste_notes)
print(moyenne)
```

14.2

```
[5]: liste_notes = []
for key, val in dico_notes.items():
    if key!="biologie":
        liste_notes.append(val)
moyenne = sum(liste_notes)/len(liste_notes)
print(moyenne)
```

15.25

4 Volume d'un cône

1. Ecrire une fonction qui renvoie le volume d'un cône de rayon r et hauteur h .
2. Ecrire une fonction qui renvoie le volume d'un tronc de cône de rayon r_1 et r_2 .
3. Ecrire une seule fonction pour laquelle le tronc de cône est par défaut un cône (i.e. $r_2 = 0$)

On rappelle quel le volume d'un cône est donné par $\frac{1}{3}h\pi r^2$ et celui d'un tronc de cône par $\frac{1}{3}h\pi(r_1^2 - r_2^2)$

```
[6]: from math import pi

def volume_cone(h, r):
    return h*pi*r**2/3

volume_cone(2, 1)

def volume_tronc_cone(h, r_1, r_2=0):
    return h*pi*(r_1**2 - r_2**2)/3

volume_tronc_cone(10, 3, 1)
```

[6]: 83.77580409572782

[]:

5 Lancé de dés

1. Ecrire une fonction `lancer_un_dé` qui prend un argument n et renvoie une liste de n nombres correspondants à n lancers successifs. On utilisera la fonction `randint` du module `random`

2. Ecrire une fonction `lancer_deux_dés` qui prend un argument n et renvoie une liste de n nombres correspondants à la somme de n lancés successifs de deux dés.
3. En prenant $n = 100000$, estimer la probabilité d'obtenir 2, 3, ..., 12.
4. Ecrire une fonction `lancer_plusieurs_dés` qui prend deux arguments n et p et renvoie une liste de n nombres correspondants à la somme de n lancés successifs de p dés.

```
[7]: from random import randint

def lancer_un_de(n):
    return [randint(1, 6) for _ in range(n)]

def lancer_deux_de(n):
    return [randint(1, 6)+randint(1, 6) for _ in range(n)]

N = 100000
test = lancer_deux_de(N)

for val in range(2, 13):
    c = test.count(val)
    print(f'Probabilité {val:2d} : {c/N:.3f}')
```

```
Probabilité 2 : 0.027
Probabilité 3 : 0.057
Probabilité 4 : 0.082
Probabilité 5 : 0.112
Probabilité 6 : 0.139
Probabilité 7 : 0.167
Probabilité 8 : 0.140
Probabilité 9 : 0.110
Probabilité 10 : 0.083
Probabilité 11 : 0.055
Probabilité 12 : 0.029
```

6 Combien La Fontaine a-t-il utilisé de mots différents dans ses fables ?

Télécharger le fichier suivant : <https://www.gutenberg.org/cache/epub/56327/pg56327.txt>

1. Ouvrir le fichier et charger son contenu dans une seule chaîne de caractère et la convertir en minuscule
2. Remplacer dans cette chaîne toutes les ponctuations par des espaces. La liste des ponctuations est donnée par : `?, ., ;, : ! \ ' \" - [] () / « »`
3. Parcourir les mots de la chaîne. Quel mot est le plus long ?
4. Combien de fois le mot `loup` est-il utilisé ?

5. Combien y a-t-il de mots différents ? Quels sont les mots utilisés exactement 25 fois? Pour répondre à ces deux questions, on va créer un dictionnaire dont les clés sont les mots et les valeurs le nombre de fois où ce mot est présent.
6. Ecrire une fonction qui prend comme argument un mot et renvoie, suivant la circonstance, un message du type : Le mot "renard" est utilisé 100 fois. ou Le mot "ordinateur" n'est jamais utilisé.

Pour cet exercices on utilisera les fonctions/méthodes suivantes. `s` est une chaîne de caractère - `s.lower()` - `s.replace(a, b)` - `s.split()` - `s.strip()`

```
[8]: punctuation_list = '?,.,:!\'\"-[]()<>'
```

```
[9]: # Question 1
with open('pg56327.txt', 'r') as f:
    data = f.read()

data = data.lower()

# Question 2
for punctuation in punctuation_list:
    data = data.replace(punctuation, ' ')

#print(data[:1000])
liste_mots = data.split()
longueur_mots = [len(mot) for mot in liste_mots]
print(max(longueur_mots))
```

17

```
[10]: # Question 3
maximum = -1
mot_le_plus_long = ""
for mot in liste_mots:
    if len(mot) > maximum:
        maximum = len(mot)
        mot_le_plus_long = mot
print(mot_le_plus_long)
```

batrachomyomachie

```
[11]: longueur_mots.index(17)
```

```
[11]: 27367
```

```
[12]: liste_mots[longueur_mots.index(17)]
```

```
[12]: 'batrachomyomachie'
```

```
[13]: # Question 4
```

```
liste_mots.count('loup')
```

```
[13]: 115
```

```
[14]: dictionnaire_nombre_mots = {}  
for mot in liste_mots:  
    dictionnaire_nombre_mots[mot] = dictionnaire_nombre_mots.get(mot, 0) + 1
```

```
[15]: nb_mots_differeents = len(dictionnaire_nombre_mots)  
nb_mots_differeents
```

```
[15]: 10499
```

```
[16]: for mot, nb in dictionnaire_nombre_mots.items():  
        if nb==25:  
            print(mot)
```

```
gloire  
ait  
sera  
quelle  
somm  
état  
tes  
corbeau  
queue  
nul  
soins  
nez
```

```
[17]: def fonction(mot):  
        if mot in dictionnaire_nombre_mots:  
            return f'Le mot "{mot}" est utilisé {dictionnaire_nombre_mots[mot]}  
            ↪fois.'  
        return f'Le mot "{mot}" n\'est jamais utilisé'  
  
print(fonction('renard'))  
print(fonction('ordinateur'))
```

```
Le mot "renard" est utilisé 100 fois.  
Le mot "ordinateur" n'est jamais utilisé
```

```
[ ]:
```