

Ds - While – Fichiers- Dictionnaires

1- EXERCICE 1.:

On donne ci-contre le script python d'une fonction :

On exécute cette fonction dans la console :

```
>>> banque(1500)
```

```
1 def banque(seuil) :
2     annee = 2024
3     somme = 1000
4     while somme < seuil :
5         annee = annee + 1
6         somme = somme * 1.1
7     return annee
```

Question : Exécuter ce code « à la main » et écrire les valeurs prises par les variables (reproduire le tableau ci-contre sur la feuille de copie). Quel est le nombre retourné ?

seuil	annee	somme
1500	2024	1000
	2025	1100
	2026	1210
	2027	1331
	2028	1464.1
	2029	1610.51

Le nombre retourné est 2029

2- EXERCICE 2.:

On exécute ci-contre la fonction *identification()*. Elle demande à l'utilisateur de saisir son mot de passe. Si ce n'est pas le bon, la demande est réitérée sans fin ... jusqu'à la saisie du bon mot de passe. Dans ce cas, cette fonction retourne True.

Question : Ecrire sur feuille de copie un script python de cette fonction.

```
>>> identification()
Entrer votre mot de passe : branly
Entrer votre mot de passe : nsi
Entrer votre mot de passe : jesaisplus
Entrer votre mot de passe : jaioublié
Entrer votre mot de passe :
Entrer votre mot de passe : ahjemesouviens
Entrer votre mot de passe : bientotLesvacances
Entrer votre mot de passe : bientotLesVacances
Bienvenue, on continue .....
True
```

```
def identification():
    saisie = ""
    while saisie != "bientotLesVacances" :
        saisie = input("Entrer votre mot de passe : ")
    print("Bienvenue, on continue .....")
    return True
```

Corrigé

3- EXERCICE 3. :

On donne la variable suivante :

```
phrase = "La beauté est dans les yeux de celui qui regarde"
```

1- Donner la ligne python qui permet d'obtenir, à partir de cette variable, la variable suivante nommée *liste* :

```
>>> liste
['La', 'beauté', 'est', 'dans', 'les', 'yeux', 'de', 'celui', 'qui', 'regarde']
```

```
>>> liste = phrase.split(" ")
```

Corrigé

2- Donner la ligne python qui permet d'obtenir, à partir de cette variable, la variable suivante nommée *L* :

```
>>> L
['La beauté', 'dans les yeux de celui qui regarde']
```

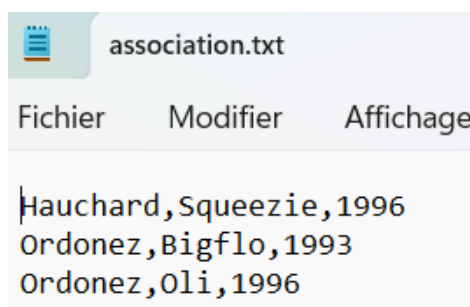
```
>>> L = phrase.split(" est ")
```

Corrigé

4- EXERCICE 4. :

La fonction *identite()* est exécutée dans la console, 2 fois de suite. Durant l'exécution, on demande à l'utilisateur de saisir les informations données sur la une copie d'écran ci-contre :

Après ces 2 exécutions, le fichier *association.txt* qui n'existait pas au départ, a été créé et il contient alors les informations suivantes :



```
association.txt
Fichier  Modifier  Affichage
Hauchard, Squeezeie, 1996
Ordonez, Bigflo, 1993
Ordonez, Oli, 1996
```

Question : Le script python de la fonction *identite()* est donné ci-dessous. Réécrire **sur votre copie** ce script, en le completant.

```
>>> identite()
Nom (ou rien écrire): Hauchard
Prénom : Squeezeie
Année de naissance : 1996
-----
Nom (ou rien écrire): Ordonez
Prénom : Bigflo
Année de naissance : 1993
-----
Nom (ou rien écrire):
-----
fin saisie
-----
```

```
>>> identite()
Nom (ou rien écrire): Ordonez
Prénom : Oli
Année de naissance : 1996
-----
Nom (ou rien écrire):
-----
fin saisie
-----
```

```
def identite():
    fichier = open("association.txt","a",encoding='utf-8')
    nom = "x"
    while nom != "" :
        nom = input("Nom (ou rien écrire): ")
        if nom != "" :
            prenom = input("Prénom : ")
            annee = input("Année de naissance : ")
            ligne = nom + ',' + prenom + ',' + annee + '\n'
            fichier.write(ligne)
        print("-----")
    print("fin saisie\n -----")
    fichier.close()
```

Corrigé

5- EXERCICE 5.:

Un dictionnaire contient les éléments suivants :

```
>>> dic
{'saison': 'pronom', 'dormir': 'verbe', 'grand': 'adjectif', 'du': 'préposition'}
```

- 1- Par rapport au vocabulaire utilisé pour les dictionnaires, qu'est 'dormir' pour ce dictionnaire dic?
'dormir' est une clé du dictionnaire dic.
- 2- Quelle ligne python doit-on écrire pour réaliser la mise à jour suivante de son contenu :

```
>>> dic
{'saison': 'nom', 'dormir': 'verbe', 'grand': 'adjectif', 'du': 'préposition'}
```

```
dic["saison"]="nom"
```

- 3- Donner le script python (boucle for ...) qui permet d'afficher dans la console, **tout** le contenu du dictionnaire, suivant le modèle ci-contre :

```
for cle in dic :
    print(f"{cle} --> {dic[cle]}")
```

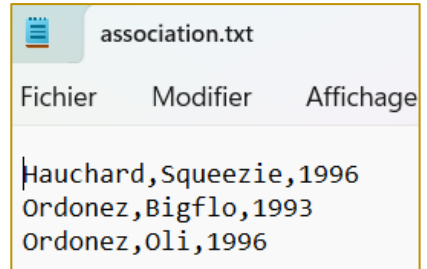
```
>>> (executing file "ex5.py")
saison --> nom
dormir --> verbe
grand --> adjectif
du --> préposition
```

6- EXERCICE 6. :

La fonction *moyenne()* est exécutée dans la console :

```
>>> moyenne()
1995.0
```

Cette fonction va lire les lignes du fichier *association.txt*, dont le contenu est donné ci-contre. Il renvoie la moyenne des nombres qui sont en fin de ligne.



Question : Le script python de la fonction *moyenne()* est donné ci-après. Réécrire **sur votre copie** ce script, en le complétant.

```
def moyenne() :
    fichier = open("association.txt", "r", encoding='utf-8')
    N = 0
    somme = 0
    ligne = fichier.readline()
    while ligne != "" :
        ligne = ligne[:-1]
        ligneEnListe = ligne.split(",")
        somme = somme + int(ligneEnListe[-1])
        N = N + 1
        ligne = fichier.readline()
    moy = somme / N
    fichier.close()
    return moy
```

Corrigé

7- EXERCICE 7. :

La fonction *valeurMot()* prend en argument un string composé de lettres. Cette fonction retourne le nombre de points que rapporte ce mot au scrabble. Les points alloués à chaque lettre sont donnés sur la figure ci-contre :

On donne ci-dessous 2 exécutions de cette fonction, dans la console :

```
>>> valeursMot("KiMono")
16
>>> valeursMot("pomme")
```

A ₁	B ₃	C ₃	D ₂	E ₁
F ₄	G ₂	H ₄	I ₁	J ₈
K ₁₀	L ₁	M ₂	N ₁	O ₁
P ₃	Q ₈	R ₁	S ₁	T ₁
U ₁	V ₄	W ₁₀	X ₁₀	Y ₁₀
		Z ₁₀		

On donne les 2 premières lignes du script python de cette fonction.

```
def valeursMot(mot) :
    points = [1,3,3,2,1,4,2,4,1,8,10,1,2,1,1,3,8,1,1,1,1,4,10,10,10,10]
    dic = {}
```

Question : Réécrire **sur votre copie** ce script, en le complétant. On n'autorise pas l'utilisation des fonctions natives de python qui permettent de transformer une majuscule en minuscule, ou le contraire.

```
def valeursMot(mot) :
    points = [1,3,3,2,1,4,2,4,1,8,10,1,2,1,1,3,8,1,1,1,1,4,10,10,10,10]
    dic = {}
    majuscules = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    minuscules = "abcdefghijklmnopqrstuvwxyz"

    for i in range(26) :
        dic[majuscules[i]] = points[i]
        dic[minuscules[i]] = points[i]
    p = 0
    for l in mot :
        p = p + dic[l]
    return p
```

Corrigé