

Exercices d'application

Exercice 1 : QCM d'application

1. Le dictionnaire {1:'un', 2:'deux', 3:'trois'} est de longueur 6

- ☐ VRAI ☐ FAUX (lg → 3)

2. Soit le dictionnaire d dont les clés sont des noms de pays et les valeurs des capitales. Si la France n'est pas dans le dictionnaire, l'instruction suivante provoque une erreur :

```
d['France']='Paris'
```

- ☐ VRAI ☐ FAUX
(cree un nouvel enregistrement)

3. Soit d un dictionnaire vide. L'instruction suivante provoque une erreur :

```
v=d['un']
```

- ☐ VRAI ☐ FAUX

4. Comment peut-on accéder à la valeur associée à une clé dans un dictionnaire?

Choix

- ☐ Il faut parcourir le dictionnaire avec une boucle à la recherche de la clé.
- ☐ On peut y accéder directement à partir de la clé.
- ☐ On ne peut pas accéder à une valeur contenue dans un dictionnaire à partir d'une clé.
- ☐ Il faut d'abord déchiffrer la clé pour accéder à un dictionnaire.

5. Voici un dictionnaire de langues :

```
dico={"anglais":"english","allemand":"deutsch","breton":"brezhoneg"}
```

On souhaite ajouter une langue en plus et obtenir le dictionnaire suivant :

```
{"anglais":"english","allemand":"deutsch","breton":"brezhoneg","espagnol":"español"}
```

Quelle instruction permet d'ajouter le dernier élément "espagnol":"español" dans le dictionnaire ?

Choix

- ☐ Ce n'est pas possible car un dictionnaire n'est pas modifiable
- ☐ dico.append("espagnol":"español")
- ☐ dico["espagnol"]="español"
- ☐ dico+=["espagnol":"español"]

6. Quelle est la valeur affichée par l'exécution du script suivant ?

```
ports = {'ftp' : 21, 'http' : 80}
ports['imap'] = 142
print(ports['imap'])
```

Choix

- ☐ Key not found ☐ 142
☐ {'imap' : 142} ☐ 2

7. Soit la suite d'instructions suivantes :

```
fruit = {} "definit un dico vide fruit"
```

```
def addone(index):
```

```
    if index in fruit: "Test si index est une cle du dico fruit"
```

```
        fruit[index] += 1 "Si oui ajoute 1 au nbre"
```

```
    else:
```

```
        fruit[index] = 1 "Sinon cree nouvelle entree et affecte valuer 1"
```

```
addone('Apple')
```

```
addone('Banana')
```

```
addone('Orange')
```

```
addone('Orange')
```

```
addone('Orange')
```

```
code ?
```

Choix

- ☐ {'Apple': 1, 'Banana': 1, 'Orange': 3}
- ☐ {'Apple', 'Banana', 'Orange', 'Orange', 'Orange'}
- ☐ {0:'Apple', 1:'Banana', 2:'Orange'}
- ☐ {1:'Apple', 1:'Banana', 3:'Orange'}

Exercice 2 : Le Scrabble

Le Scrabble est un jeu de plateau où l'on doit former des mots avec un tirage aléatoire de lettres, chaque lettre valant un certain nombre de points.

Le plateau ci-dessous donne les points des 11 premières lettres de l'alphabet.

Lettre	A	B	C	D	E	F	G	H	I	J	K
Points	1	3	3	2	1	4	2	4	1	8	5

On considère le programme suivant :

```
def calculPoints(mot) :
    scrabble = {'A':1, 'B':3, 'C':3,
                'D':2, 'E':1, 'F':4, 'G':2,
                'H':4, 'I':1, 'J':8, 'K':5}
    somme = 0
    for i in mot :
        if i in scrabble :
            somme+=scrabble[i]
        else :
            return 'error'
    return somme
```

1. **Indiquer** le type de la variable scrabble. **Expliquer** l'affectation de cette variable.
2. **Indiquer** le type de la variable i. **Justifier** la présence du test if dans ce programme.
3. **Déterminer** la valeur de retour de l'appel calculPoints('KAKI'). **En déduire** le rôle de ce programme.

Exercice 3 : Base de données

Une entreprise possède une base de données dans laquelle elle enregistre les informations personnelles de ses clients sous le format :

nom, prénom, âge, taille, poids

Elle initialise cette base de données par la ligne suivante :

```
registre = [{'nom':'HILIUM',
              'prenom':'Noa', 'age':27, 'taille':178,
              'poids':'80'}, {'nom':'TRESTERE',
                              'prenom':'Lucie, 'age':30, 'taille':167,
                              'poids':'54'}]
```

1. **Indiquer** le type de la variable registre
2. **Indiquer** le type de registre[0]. **Déterminer** la valeur retournée par l'instruction registre[0]['prenom']
3. **Déterminer** l'instruction qui renvoie l'âge de Noa HILIUM.
4. **Déterminer** l'instruction qui permet d'ajouter à cette base de données les informations de Kevin Dupont, 24 ans, mesurant 1,87m pour 95kg.

Exercice 4 : Tableau physico-chimique

Le tableau suivant représente des informations physico-chimiques sur des éléments simples : température d'ébullition (Te), et de fusion (Tf), numéro de masse (Z) et masse atomique (M)

	Te	Tf	Z	M
Au	2970	1063	79	196,96
Ga	2237	29,8	31	69,7

5. **Affecter** les données de ce tableau à un dictionnaire Python, nommé tabPhys, de façon à ce que l'instruction `tabPhys['Au']['Z']` renvoie la valeur 79

Exercice 5 : Analyse de texte

Objectif de l'exercice : Définir une fonction qui analyse un texte et compte les occurrences des différents mots de ce texte.

Pour atteindre cet objectif les deux fonctions suivantes seront utilisées :

```
def occurrences(texte):
    ''' Compte le nombre d'occurrence de chaque mot d'un texte
    arg : texte : texte a etudier (string)
    retour : dictionnaire des mots (cle type str) associes au nombre d'occurrence
            (valeur type int)
    '''
    dico_occur = {}
    for mot in texte.split():
        mot = mot.lower() # convertit le mot en minuscule
        mot = mot.strip(",.?!?-( )<>_") # enleve ponctuations collees aux mots
        if mot in dico_occur: # teste si mot connu
            dico_occur[mot] += 1
        else:
            dico_occur[mot] = 1
    return dico_occur

def ouvrir_fichier(fichier):
    ''' Retourne le texte contenu dans un fichier texte
    Arg : fichier : nom du fichier txt
    Retour : texte (type str)
    '''
    with open(fichier, 'r') as f:
        texte = f.read()
    return texte

texteAEtudier = ouvrir_fichier("LesMiserables.txt")
dico = occurrences(texteAEtudier)
```

Pour les questions suivantes, le fichier LesMiserables.txt disponible sur le site <http://www.gutenberg.org> sera utilisé.

1. **Déterminer** l'instruction permettant de renvoyer le nombre de mots différents dans Les Misérables.
2. **Écrire** une fonction `mot_le_plus_frequent(dico)` qui renvoie le mot le plus fréquent dans le dictionnaire dico associant des mots à leur nombre d'occurrences.
3. **Écrire** une fonction `le_plus_frequent(dico, n)` qui renvoie le mot de n lettres le plus fréquent d'après le dictionnaire dico. S'il n'y a pas de mot de longueur n, renvoyer un texte vide.
4. **Écrire** une fonction `plus_frequents(dico)` qui affiche les messages suivants :

```
>>> plus_frequents(dico)
1 lettre(s) : MOT avec NB
2 lettre(s) : MOT avec NB
3 lettre(s) : MOT avec NB
4 lettre(s) : MOT avec NB
```

Où MOT et NB sont remplacés par les bonnes valeurs.

5. **Déterminer** le nombre de lettres maximum à traiter.

