

Fiche 3

Dictionnaires en Python

Exercice 1: On considère un dictionnaire contenant les notes (on suppose que toutes les notes sont entre 0 et 20). Par exemple :

```
mon_dico = { "note_1" : 12 , "note_2" : 15 , "note_3" : 14 , "note_4" : 14 ,
"note_5" : 13 }
```

1. Écrire la fonction `max_min` qui prend un tel dictionnaire, et renvoie le dictionnaire contenant le minimum et le maximum des notes.

Exemples :

```
>>> min_max(mon_dico)
{'max': 15, 'min': 12}
```

Correction

```
def statistique(dico) :
    dico_stats = {"min" : 21 , "max" : -1 , "somme" : 0 }

    for note in dico.values() :
        if note <= dico_stats["min"] :
            dico_stats["min"] = note
        if note >= dico_stats["max"] :
            dico_stats["max"] = note
        dico_stats["somme"] += note
    moyenne = dico_stats["somme"]/len(dico)
    ecart = dico_stats["max"]-dico_stats["min"]
    dico_stats["moyenne"] = moyenne
    dico_stats["ecart"] = ecart
    return dico_stats
```

2. Écrire la fonction `statistique` qui prend un tel dictionnaire, et renvoie le dictionnaire contenant le minimum, le maximum, la moyenne ainsi que l'écart entre le minimum et le maximum.(la fonction ne devra lire qu'une fois le dictionnaire.)

Exemples :

```
>>> statistique(mon_dico)
{'ecart': 3, 'max': 15, 'min': 12, 'moyenne': 13.6, 'somme': 68}
```

Correction

```
def min_max(dico) :
    dico_min_max = {"min" : 21 , "max" : -1 }
    for note in dico.values() :
        if note <= dico_min_max["min"] :
            dico_min_max["min"] = note
        if note >= dico_min_max["max"] :
            dico_min_max["max"] = note
    return dico_min_max
```

Exercice 2: Un élève de première décide d'utiliser les dictionnaires pour stocker la liste de ses notes par matière.

1. En partant d'un dictionnaire vide, donner les procédures pour obtenir un dictionnaire contenant des clés relatives aux matières sans notes.

Correction

```
mon_dico = {}

mon_dico["Maths"] = []
mon_dico["NSI"] = []
mon_dico["Langue_1"] = []
mon_dico["Langue_2"] = []
mon_dico["Français"] = []
mon_dico["Histoire"] = []
mon_dico["EPS"] = []
```

2. Écrire la fonction `inclusion_note` qui prend en argument une matière et une note puis alimente la liste des notes de la matière. Si la matière n'existe pas, la clé est créée, avec sa note.

Correction

```
def inclusion_note(matiere, note) :
    global mon_dico
    if matiere in mon_dico :
        mon_dico[matiere].append(note)
    else :
        mon_dico[matiere] = [note]
```

3. Écrire la fonction `moyenne_dico` qui ne prend pas d'argument et renvoie le dictionnaire des moyennes par matières. Si la matière n'a pas de note, matière n'apparaît pas dans le dictionnaire.

Correction

```
def moyenne(liste) :
    somme = 0
    for i in liste :
        somme += i
    if len(liste) == 0 :
        return "pas_de_note"
    else :
        return somme/len(liste)

def moyenne_dico() :
    global mon_dico
    dico_moy = {}
    for k in mon_dico.keys() :
        if mon_dico[k] != [] :
            dico_moy[k] = moyenne(mon_dico[k])
    return dico_moy
```

Exercice 3 :

1. Ecrire une fonction `comptage_lettre` qui prend en argument un mot, et renvoie le dictionnaire associant à chaque lettres, le nombre d'occurrence de cette lettre dans le mot.

Exemples :

```
>>> comptage_lettre("perlimpimpin")
{'e': 1, 'i': 3, 'l': 1, 'm': 2, 'n': 1, 'p': 3, 'r': 1}
```

Correction

```
def comptage_lettre(mot) :
    dico = {}
    for l in mot :
        if l in dico :
            dico[l] += 1
        else :
            dico[l] = 1
    return dico
```

2. Ecrire la fonction `construction_mot` d'un mot, qui à l'inverse prend le dictionnaire précédent, puis construit un mot avec le même nombre de lettres données par le dictionnaire.

```
>>> construction_mot({'e': 1, 'i': 3, 'l': 1, 'm': 2, 'n': 1, 'p': 3, 'r': 1})
'eiilmnpppr'
```

Correction

```
def construction_mot(dico) :
    mot = ""
    for i in dico.items() :
        (lettre, nombre) = i
        while nombre != 0 :
            mot += lettre

            nombre -= 1
    return mot
```
