

TP6 – Les dictionnaires et les fichiers

Les dictionnaires augmentent encore la puissance de Python en permettant de gérer des collections plus évoluées que les listes. En effet, les dictionnaires peuvent être considérés comme des listes dans lesquelles l'indice, au lieu d'être un entier positif ou nul, peuvent être une clé qui est une chaîne de caractères. Ainsi, chaque élément du dictionnaire est défini par une clé et une valeur. Je vous propose, dans ce qui suit de faire quelques manipulations sur les dictionnaires, en particulier, vous allez créer des dictionnaires les parcourir et les manipuler (accès, ajout mis à jour des éléments).

Manipulations : Tapez les commandes suivantes (en mode commande) :

```
>>> notes = {}
>>> notes['Mohand']=14
>>> notes['Karima']=15
>>> notes['Said']=12
>>> notes['Amel']=18.5
```

Q1 : de quel type est la variable notes ?

Q2 : Pouvons-nous modifier la clé "Mohand" ?

```
>>> print (notes)
{'Said': 12, 'Amel': 18.5, 'Mohand': 14, 'Karima': 15}
>>> notes.pop('eee', 'Attention clé introuvable')
'Attention clé introuvable '
>>> notes.pop('Mohand', 'erreur')
14
>>>
```

Q3 : Que fait la méthode « pop() » ?

Q4 : Pourquoi, dans la seconde commande ci-dessus, python affiche 'Attention clé introuvable' ?

Q5 : Pourquoi, dans la troisième commande ci-dessus, python affiche 14 ?

```
>>> s = 0
>>> for elem in notes.values():
    s = s + elem

>>> m = s / len(notes.values())
>>> print (m)
```

Q6 : Que contient la variable « m » ?

```
>>> for etudiant in notes.keys():
    print(etudiant)
```

Q7 : Que fait la boucle ci-dessus ?

```
>>> for n in notes.values():
    print(n)
```

Q8 : Que fait la boucle ci-dessus ?

```
>>> print (notes.values())
[12, 18.5, 15]
```

Q9 : Que fait la commande ci-dessus ?

```
>>> print (notes.keys())
[12, 18.5, 15]
```

Q10 : Que fait la commande ci-dessus ?

```
>>> notes['bouzidi']=5.5
```

Q11 : Que fait la commande ci-dessus ?

Exercices : Attention, dans l'exercice suivant, vous devez travailler en mode « scripte »

Exercice 1 : tapez le programme suivant :

```
# -*- coding: utf-8 -*-
molecule = {}
molecule['H2O'] = 'eau'
molecule['O2'] = 'Bi-oxygène'
molecule['HCL'] = 'Acide chloridrique'
molecule['H2SO4'] = 'Acide sulfurique'

for formule, description in molecule.items():
    print ("la formule chimique de ", description, " est ", formule)
```

Que fait le programme ci-dessus ?

Exercice 2 : Améliorez le programme ci-dessus en offrant la possibilité de saisir des molécules avec leur description à partir du clavier.

Les fichiers permettent de sauver des données sur supports externes (permanent comme le disque ou la clé USB). Je vous propose de reprendre le programme sur les molécules (ci-

dessus) et de l'améliorer de sorte qu'au lieu de saisir les molécules à partir du clavier, on pourra les lire depuis un fichier.



Je vous rappelle que pour accéder à un fichier physique, vous devez créer une variable « objet-fichier » et lui la fonction **open**('meolecules.txt', 'w'). Vous devez créer le fichier physique en mode commande en écrivant la commande suivante :

```
>>> f = open('meolecules.txt','w')
```

Ensuite, vous pourrez le lire et le modifier dans votre programme (mode scripte) en vous servant du mode mise à jour :

```
f = open('meolecules.txt','r+')
```

Lorsque vous terminer fermez le fichier en utilisant la méthode **close()**