

# INFO-H-100 - Programmation

## TP 10 — Dictionnaires et fichiers

Lorsque l'exercice demande d'écrire une fonction, écrivez la fonction demandée et testez-la avec plusieurs valeurs pertinentes.

**Ex. 1.** Écrire une fonction qui renvoie un dictionnaire composé du nombre d'occurrences de tous les mots de 2 lettres (AA, AC, AG, ...) qui existent dans une séquence nucléotidique.

Exemple :

```
>>> dico = occurrences_of_2_char("ACCTAGCCATGTAGAATCGCCTAGGCTTTAGCTAGCTCTAGCTAGCTG")
>>> print(dico)
{'AA': 1, 'AC': 1, 'GT': 1, 'AG': 7, 'CC': 3, 'CA': 1, 'CG': 1,
 'TT': 2, 'GG': 1, 'GC': 7, 'AT': 2, 'GA': 1, 'TG': 2, 'CT': 8,
 'TC': 2, 'TA': 7}
```

**Ex. 2.** Écrire une fonction qui affiche un dictionnaire de la façon suivante :

```
Key : GT - Value : 1
Key : CT - Value : 8
...
```

Bonus : afficher les clefs dans l'ordre alphabétique.

**Ex. 3.** Écrire une fonction qui renvoie un dictionnaire composé de chaque mot contenu dans le fichier `dico.txt` disponible sur la page web des TP (la valeur associée à la clef n'a pas d'importance dans cet exercice). Le fichier `dico.txt` est composé de lignes contenant chacune un mot en majuscule. Chaque mot est unique.

Ensuite, écrire une fonction qui vérifie qu'un mot donné en paramètre est bien dans ce dictionnaire.

```
dico = load_dico("dico.txt")
print(is_in_dico("bonjour", dico)) #-> True
```

**Ex. 4.** Écrire une fonction qui renvoie un dictionnaire contenant le nombre d'occurrences de chaque mot du fichier `hamlet.txt` disponible sur la page web des TP.

Avant cela, veuillez à nettoyer le texte, c'est à dire enlever chaque caractère de ponctuation et mettre en majuscule tous les mots. Renseignez-vous dans la documentation pour trouver les fonctions nécessaires à ce travail (module `string`).

Créez ensuite une fonction qui détermine le mot le plus utilisé dans ce texte.

```
dico = occurrences_of_words("hamlet.txt")
print(most_frequent_word(dico)) # -> THE
```

**Ex. 5.** Soit le fichier `address.dat` suivant :

```
LastName : Verhaegen
FirstName : Boris
Office : UB4.131
Phone : 026503766
=====
LastName : Silovy
FirstName : Alain
```

Les entrées sont séparées par une ligne de 20 signes `=`. Elles sont composées de champs de forme `cle : valeur` séparés par des retour à la ligne. Les champs `LastName` et `FirstName` sont obligatoires et les champs `Office` et `Phone` sont optionnels. L'ordre des entrées et des champs n'a pas d'importance.

Écrire un programme qui permet :

- de charger le carnet d'adresse dans une liste de dictionnaires,

- d’afficher le carnet d’adresse
- d’ajouter une entrée dans le carnet d’adresse,
- de supprimer une entrée dans le carnet d’adresse et
- de sauvegarder le carnet d’adresse dans le fichier `address.dat`

# INFO-H-100 - Programmation

## TP 10 — Dictionnaires et fichiers

### Corrections

---

#### Solution de l'exercice 1:

```
def occurrences_of_2_char(sequence):
    dico = {}
    for i in range(len(sequence)-1):
        key = sequence[i]+sequence[i+1]
        dico[key] = dico.get(key,0) + 1
    return dico
```

#### Solution de l'exercice 2:

```
def print_dico(dico):
    for key in dico:
        print("Key : "+str(key)+" - Value : "+str(dico[key]))
```

Version triée :

```
def print_dico(dico):
    for char in sorted(dico):
        print("Key: " + str(char) + " - Value: " + str(dico[char]))
```

ou

```
def print_dico(dico):
    keys = dico.keys()
    keys.sort()
    for key in keys:
        print("Key : "+str(key)+" - Value : "+str(dico[key]))
```

#### Solution de l'exercice 3:

```
def load_dico(filename):
    f = open(filename)
    list_words = f.read().split()
    f.close()
    dico = {}
    for word in list_words:
        dico[word] = None
    return dico

def is_in_dico(word,dico):
    return word.upper() in dico

dico = load_dico("dico.txt")
print(is_in_dico("bonjour",dico))
```

Autre solution avec un set :

```
def load_dico(filename):
    f = open(filename)
    list_words = f.read().split()
    f.close()
    dico = set()
    for word in list_words:
        dico.add(word)
    return dico
```

#### Solution de l'exercice 4:

```
def get_words(text):
    import string
    itab = string.punctuation #les signes de punctuation...
    otab = " " * len(string.punctuation) #a remplacer par des blancs
    tab = text.maketrans(itab, otab)
    text = text.translate(tab)
```

```

    text = text.upper()
    return text.split()

def occurrences_of_words(filename):
    f = open(filename)
    text = f.read()
    f.close()
    word_list = get_words(text)
    dico = {}
    for word in word_list:
        dico[word] = dico.get(word, 0) + 1
    return dico

def most_frequent_word(dico):
    maxi = 0
    max_word = ""
    for word in dico:
        if dico[word] > maxi:
            maxi = dico[word]
            max_word = word
    return (max_word, maxi)

dico = occurrences_of_words("hamlet.txt")
print(most_frequent_word(dico))

```

## Solution de l'exercice 5:

```

def get_entry_dico(entry):
    lines = entry.split("\n")
    dico = {}
    for line in lines:
        if line != '':
            key,value = line.split(" : ")
            dico[key]=value
    return dico

def load_address_file():
    f = open("address.txt")
    t = f.read()
    f.close()
    t = t.strip()
    entries = t.split("\n=====\\n")
    address_list = []
    for entry in entries:
        dico = get_entry_dico(entry)
        address_list.append(dico)
    return address_list

def address_to_string(address):
    s = ""
    for (key,value) in address.items():
        s+=key+" = "+value+" "
    return s

def show_book(address_list):
    for i in range(len(address_list)):
        s = "["+str(i)+"]"+address_to_string(address_list[i])
        print(s)

def action_add(address_list):
    first_name = input("Enter a first name (mandatory) : ")
    last_name = input("Enter a last name (mandatory) : ")
    office = input("Enter office (optional) : ")
    phone = input("Enter phone (optional) : ")
    dico = {"First_name":first_name, "Last_name":last_name}
    if office != "":
        dico["Office"] = office
    if phone != "":
        dico["Phone"] = phone
    address_list.append(dico)

def action_delete(address_list):
    to_del = int(input("Item to delete : "))
    del address_list[to_del]

def action_save(address_list):

```

```
f = open("address.txt", "w")
for i in range(len(address_list)):
    for (key, value) in address_list[i].items():
        f.write(key+" : "+value+"\n")
    if i < len(address_list)-1:
        f.write("=====\n")
f.close()

def menu():
    address_list = load_address_file()
    text = """1) Delete item\n2) Add item\n3) Save\n4) Exit"""
    choice = 0
    while choice != 4:
        show_book(address_list)
        print(text)
        choice = int(input(" > "))
        if choice == 1:
            action_delete(address_list)
        elif choice == 2:
            action_add(address_list)
        elif choice == 3:
            action_save(address_list)

menu()
```