

Les dictionnaires en Python

Romain Tavenard

1 Organisation de votre code

Pour ce TD, vous créerez un nouveau fichier `td6.py` dans le répertoire que vous avez créé à la première séance. Dans ce fichier, votre code sera organisé de la manière suivante :

```
def ma_premiere_fonction_qui_a_un_meilleur_nom_que_celui_la(n):  
    [...]  
  
def mon_autre_fonction(truc, machin):  
    [...]  
  
# Tests  
print(ma_premiere_fonction_qui_a_un_meilleur_nom_que_celui_la(3))  
# [Sortie] 3
```

Notamment, vous définirez vos fonctions en début de fichier et les appels seront listés en fin de fichier. De cette manière, vous pourrez, d’une question à l’autre, réutiliser les fonctions déjà codées au besoin.

2 Manipulations de base des dictionnaires

2.1 Construction de dictionnaire

Il existe deux façons de déclarer un dictionnaire dont les clés sont des chaînes de caractères, dont voici une illustration :

```
d = dict(a=5, b=7, c=1, d=7, e=7)  
d = {"a": 5, "b": 7, "c": 1, "d": 7, "e": 7}
```

Si, par contre, on souhaite utiliser pour clé autre chose qu’une chaîne de caractère, on utilisera la seconde notation :

```
d2 = {1: 12, 1000: 7}
```

La copie de dictionnaire pose exactement, en Python, les mêmes problèmes que la copie de liste. On utilisera la fonction `dict` pour dupliquer le contenu d'un dictionnaire.

1. Écrivez une fonction qui prend en entrée un dictionnaire et retourne la liste de ses éléments sous forme de paires comme retournée par la méthode `items`, triés par clé.
2. Écrivez une fonction qui prend en entrée une chaîne de caractères et retourne un dictionnaire indiquant, pour chaque mot, sa fréquence dans la chaîne fournie.

3 Exercices de synthèse

3. On dispose d'un dictionnaire associant à des noms de commerciaux d'une société le nombre de ventes qu'ils ont réalisées. Par exemple :

```
ventes={"Dupont":14, "Hervy":19, "Geoffroy":15, "Layec":21}
```

- a. Écrivez une fonction qui prend en entrée un tel dictionnaire et renvoie le nombre total de ventes dans la société.
 - b. Écrivez une fonction qui prend en entrée un tel dictionnaire et renvoie le nom du vendeur ayant réalisé le plus de ventes. Si plusieurs vendeurs sont *ex-aequo* sur ce critère, la fonction devra retourner le nom de l'un d'entre eux.
4. Écrivez une fonction qui prend en entrée une chaîne de caractères comprenant, sur chaque ligne, trois champs séparés par des caractères ';' (un numéro d'étudiant, un nom et un prénom) et retourne un dictionnaire dont les clés sont les numéros d'étudiants lus et les valeurs sont, pour chaque numéro d'étudiant, une chaîne correspondant à la concaténation des prénom et nom de la personne. On pourra tester la fonction avec la chaîne suivante :

```
chaine_etudiants = """213615200;BESNIER;JEAN
213565488;DUPOND;MARC
214665555;DURAND;JULIE"""
```

5. Écrivez une fonction qui prend en entrée un dictionnaire associant à un nom une liste de notes et qui retourne la liste des noms des personnes qui ont la moyenne la plus élevée (s'il y a des *ex-aequo*, cette liste contiendra plusieurs éléments, sinon, elle n'en contiendra qu'un) et la moyenne correspondante. On pourra utiliser le dictionnaire suivant pour tester la fonction ainsi écrite :

```
notes = {"Romain": [1, 5, 12], "Jean": [15, 16, 19], "Robert": [5, 12], "Michel": []}
```