

## TP Informatique 30

### Exercice 1

Un *dictionnaire* est un objet de type `dict` qui réalise une association entre des *clés* et des *valeurs*.

1. Saisir `dico={"chien":"dog","chat":"cat"}` puis afficher `dico` et son type.
2. Tester les instruction `dico["chien"]`, `dico["chat"]`, `dico["oiseau"]`. Qu'observe-t-on ?
3. Tester les instructions `"chien" in dico` et `"bird" in dico`. Qu'observe-t-on ?
4. Saisir `dico["oiseau"]="bird"`, puis afficher `dico`, `dico.keys()` et `dico.values()`.
5. Générer la liste des clés de `dico`
6. Avec l'instruction `for cle in dico:` qui permet le parcours des clés de `dico`, afficher les clés dont la valeur contient la consonne "d".

### Exercice 2

Récupérez le fichier `scrabble.py`, et complétez la fonction `points`.

Cette fonction doit prendre en argument un mot en lettres majuscules, et renvoyer le nombre de points obtenus au scrabble avec ce mot.

### Exercice 3

On effectuera les importations suivantes :

```
import numpy as np, numpy.random as rd
```

Dans ce qui suit, les fonctions devront impérativement utiliser des dictionnaires. Pour `n` entier non nul, l'instruction `rd.randint(1,n+1)` effectue un tirage aléatoire selon une loi uniforme dans  $\llbracket 1; n \rrbracket$ . On pourra tester les fonctions des deux premières questions sur une liste générée aléatoirement par `list(rd.randint(1,5,10))` par exemple (liste de dix nombres tirés au hasard entre 1 et 4).

1. Écrire une fonction `elt(L)` d'argument `L` une liste d'entiers qui renvoie la liste des éléments de `L` sans doublon. Par exemple, l'appel `elt([2, 1, 2, 3, 1])` renvoie `[2, 1, 3]`.
2. Écrire une fonction `doublon(L)` d'argument `L` une liste d'entiers qui renvoie `True` si un élément est en doublon et `False` sinon.
3. Écrire une fonction `kpremiers(n,k)` d'arguments `n`, `k` des entiers et qui renvoie les `k` premiers éléments distincts obtenus au cours de tirages successifs selon la loi uniforme dans  $\llbracket 1; n \rrbracket$ .

## Exercice 4

Dans cet exercice, on cherche à déterminer la langue dans laquelle est écrit un texte en s'appuyant sur la fréquence d'apparition des lettres d'un alphabet dans le texte considéré. On complètera le fichier `rech_langue.py` sur le site de la classe.

1. Écrire une fonction `frequence(texte)` d'argument `texte` une chaîne de caractères qui renvoie un dictionnaire dont les clés sont les lettres de `texte` avec pour valeur la fréquence d'apparition de chaque lettre. Par exemple, l'appel `frequence("langage")` renvoie

'l': 0.14285714285714285, 'a': 0.2857142857142857, 'n': 0.14285714285714285, 'g': 0.2857142857142857, 'e': 0.14285714285714285

puisque la chaîne "langage" est de taille égale à 7 et  $\frac{1}{7} \simeq 0.1428$ .

2. Soit A un alphabet constitué par une chaîne de caractères. On définit la distance entre deux dictionnaires de fréquences d'apparition de lettres  $D_1$  et  $D_2$  par

$$d(D_1, D_2) = \sum_{x \in A} |D_1[x] - D_2[x]|$$

avec la convention  $\forall i \in \{1, 2\} \quad D_i[x] = 0 \quad \text{si } x \notin D_i$

Écrire une fonction `dist(D1, D2)` d'arguments `D1`, `D2` des dictionnaires contenant des fréquences d'apparition de lettres et qui renvoie le calcul de la distance entre  $D_1$  et  $D_2$ . L'alphabet considéré sera l'alphabet français.

3. Décommenter la zone de code mettant en œuvre la détermination de la langue d'un texte puis exécuter le code qui réalise cette détermination sur quatre textes écrits dans des langues différentes.

## Exercice 5

Dans ce qui suit, les fonctions devront impérativement utiliser des dictionnaires. On pourra les tester sur une liste générée aléatoirement par `list(rd.randint(1, 5, 10))` par exemple (liste de dix nombres tirés au hasard entre 1 et 4).

1. Écrire une fonction `freqmax(L)` d'argument `L` une liste d'entiers qui renvoie un élément de `L` et son nombre d'occurrences tel que ce nombre est maximal. Par exemple, l'appel de `freqmax([1, 1, 2, 2, 1])` renvoie (1,3).
2. Écrire une fonction `freqmax_occ(L)` d'argument `L` une liste d'entiers qui renvoie un élément de `L` dont le nombre d'occurrences est maximal et la liste des indices de ses occurrences. Par exemple, l'appel de `freqmax_occ([1, 1, 2, 2, 1])` renvoie (1, [0, 1, 4]).

## Exercice 6

Ouvrir le fichier `dico_cplx.py`, l'exécuter puis proposer diverses conjectures de complexité sur les manipulations de dictionnaires.