

## Langages de script

### TP n° 4 : Listes d'associations (dictionnaires)

Les exercices optionnels sont marqués avec (★).

#### Exercice 1 : Dictionnaires

1. Écrire une fonction `listeversdico(l)` qui prend une liste et qui produit un dictionnaire correspondant, où les clés sont les indices de la liste. Par exemple `listeversdico(["petit","klein","small"])` retourne `{0: 'petit', 1: 'klein', 2: 'small'}`. Écrire au moins deux versions significativement différentes.

2. Écrire une fonction `mergedico(d1,d2)` qui prend deux dictionnaires et produit un nouveau dictionnaire contenant les deux. Par exemple,

```
d1,d2 = {1:2,2:3},{1:4,4:4}
mergedico(d1,d2)
```

donne `{1: 4, 2: 3, 4: 4}`. Attention `d1` et `d2` doivent être inchangés.

3. Écrire une fonction `chars(w)`, où `w` est une chaîne de caractères, qui renvoie un dictionnaire (table d'association) dont les clés sont les caractères de `w` et la valeur associée à une clé `k`, le nombre d'occurrences de `k` dans `w`. Écrire ensuite une fonction `affichedico(d)` qui affiche pour un dictionnaire `d` chaque couple clé et valeur.

Par exemple : `affichedico(chars("aacababbcccd"))` donne

```
c 3
d 1
b 3
a 4
```

4. Écrire une fonction `merge(d1,d2)` qui prend en argument deux dictionnaires (tables d'association) et retourne un dictionnaire formé des clés communes aux deux dictionnaires et dont les valeurs sont des 2-uplets comprenant les valeurs des deux dictionnaires.

Par exemple :

```
>>> d1 = {'r':0.56, 't':0.78, 'i':0.23, 'u':0.35}
>>> d2 = {'i':5, 'v':89, 'p': 65, 't':21, 'b':55}
>>> merge(d1,d2)
{'i':(0.23,5), 't':(0.78,21)}
```

#### Exercice 2 : Traduction automatique

On souhaite écrire un programme de traduction automatique simpliste. Le lexique de traduction est stocké dans un fichier (des exemples se trouvent sur Didel) dont chaque ligne est de la forme `mot:traduction` (on suppose que `traduction` est un mot unique sans espace).

1. Écrivez une fonction `read_lexic` prenant comme argument le nom d'un fichier et renvoyant un dictionnaire Python dont chaque couple (clé, valeur) correspond à un mot du lexique et à sa traduction. Testez votre fonction sur `dicosimple.txt`. Dans `dico2.txt` un mot peut avoir plusieurs traductions. Qu'est-ce qui se passe dans ce cas ?
2. Écrivez une fonction `translate_word` prenant un mot et un dictionnaire et renvoyant sa traduction selon ce lexique.
3. Écrivez une fonction `add_word` prenant deux mots et un dictionnaire et ajoutant le mot et sa traduction au dictionnaire.
4. Écrivez une fonction `write_lexic` réciproque de la fonction `read_lexic`. Améliorez cette fonction en faisant en sorte que les entrées soient classées par ordre alphabétique.
5. Écrivez un script `traduction` qui demande à l'utilisateur un mot et lui fournit la traduction si elle existe dans le lexique. Si elle n'existe pas, il demande à l'utilisateur de donner la traduction et la sauvegarde dans le dictionnaire. Testez votre script en utilisant les fichiers fournis.
6. Écrivez un script `traductiontext` qui prend en entrée le nom d'un fichier contenant un lexique, le nom d'un fichier contenant un texte (uniquement composé de mots et de blancs, par exemple le fichier `testtraduction.txt` sur Didel) ainsi que le nom d'un fichier de sortie. Ce script crée un nouveau fichier contenant la traduction du texte par rapport au lexique donné. Si un mot ne se trouve pas dans le lexique, il insère à la place `xxx`.

### Exercice 3 : Traduction automatique 2

Un mot peut avoir plusieurs traductions (voir fichier `dico2.txt`) . Pour traiter ce cas, au lieu de stocker dans le dictionnaire pour chaque mot uniquement une traduction, on peut stocker une liste de traduction, par exemple `{'stupid': ['stupide', 'idiot']}`.

1. Réécrivez les fonctions suivantes de l'exercice précédent pour traiter la nouvelle situation :
  - `read_lexic`,
  - `translate_word` (qui renvoie une liste),
  - `add_word` (qui ajoute aussi une traduction supplémentaire pour un mot déjà inclu dans le dictionnaire)
  - `write_lexic`
2. Un texte peut avoir maintenant plusieurs traductions.
  - Écrivez une fonction `traductionmultiple`, qui étant donnée un dictionnaire pour un lexique et une liste de mots, produit une liste de listes de mots traduits. Par exemple, `traductionmultiple({'I': 'je', 'stupid': ['stupide', 'idiot']}, ['I', 'stupid', 'stupid'])` produit `[[ 'je'], [ 'stupide', 'idiot'], [ 'stupide', 'idiot'] ]`.
  - Réécrivez le script `traductiontexte` de sorte qu'il produise autant de fichiers différents qu'il y a de traductions différentes. Si l'utilisateur donne comme nom de fichier de sortie `nom`, le script produit des fichiers `nom0`, `nom1`, etc.

**Indication :** Utilisez la fonction `produit` du TP3.
3. Écrivez une fonction `inverse` qui étant donné un dictionnaire de traduction de la langue A vers B produit un dictionnaire de la langue B vers A.