

Travaux Pratiques

Initiation à la programmation avec Python (v3)

Feuille n.4

Boucles, Interactions avec l'utilisateur, Dictionnaires

Copyright (C) 2015-2019 Jean-Vincent Loddo
Licence Creative Commons Paternité - Partage à l'Identique 3.0 non transposé.

Construire un programme **indépendant** (*shebang*, `chmod +x`) pour chaque exercice. Ceci n'empêche pas de tester des bouts de code dans l'interpréteur, avant de les intégrer dans le programme. Toutes les questions et les résultats présentés par les programmes doivent être clairement compréhensibles à l'utilisateur.

1 Jeu du nombre caché

Un joueur *Secret* tire au hasard un nombre entier entre 0 et 100. Un joueur *Détective* essaye de deviner ce nombre. À chaque tour de jeu, *Détective* tente sa chance avec un nombre x . Le joueur *Secret* répond alors de trois façons possibles :

- “tu as deviné, c’est bien x le nombre caché”, si le nombre caché est égal à x
- “plus petit”, si le nombre caché est inférieur à x
- “plus grand”, si le nombre caché est supérieur à x

Dans le premier cas, le jeu se termine, dans les deux autres le jeu continue (en boucle).

- (a) Écrire un programme `detective.py` qui permet à l'utilisateur de jouer dans le rôle de *Détective* (celui qui doit deviner) contre l'ordinateur qui joue dans le rôle de *Secret* (celui qui cache le nombre).
(b) Modifier le programme pour qu'il affiche, à la fin de la partie, le **score** de l'utilisateur, c'est-à-dire le nombre de tentatives effectuées par l'utilisateur-détective pour arriver à la solution
- Écrire un programme `secret.py` qui permet à l'utilisateur de jouer dans le rôle de *Secret* contre l'ordinateur qui joue dans le rôle de *Détective*. À la fin du match, le programme affichera son propre **score**. Pour que le programme joue de façon intelligente, il faudra qu'il réalise une recherche appelée *dichotomique*. Sur la page web <https://fr.wikipedia.org/wiki/Dichotomie> vous trouverez une explication de ce type de recherche avec justement le jeu du nombre caché (Paul dans le rôle de *Secret*, Julie dans le rôle de *Détective*).
- Écrire un programme `le_plus_rapide.py` où chacun des joueurs (utilisateur, ordinateur) essaye de deviner le nombre caché par l'autre, donc chacun joue à tour de rôle comme *Secret* et *Détective*. Gagne le match celui qui fait le meilleur score, c'est-à-dire celui qui devine en premier.

2 Dictionnaires

Pour s'exercer avec la *structure de données* des dictionnaires, nous allons réaliser de *vrais* dictionnaires, ce qui nous permettra au moins de comprendre pourquoi ils s'appellent ainsi...

- (a) Écrire un programme `traducteur_couleurs_fr_en.py` qui demande à l'utilisateur un mot représentant une couleur en français ("blanc", "noir", "rouge", "bleu", "marron", "vert", "gris", "rose", "jaune", "orange", "violet") et affiche l'équivalent en anglais. Exemple d'exécution :

```
Couleur en français ? rose
Traduction anglaise : pink
```

- (b) Modifier le programme pour qu'il affiche un message d'accueil et pour qu'il **répète** la traduction (les actions question-réponse) tant que l'utilisateur ne saisi pas une chaîne vide "" ou la chaîne **"quitter"**. Avant de sortir, le programme affichera aussi un message de salutations ou remerciement.

```
Bienvenue dans ce formidable outil de traduction français-anglais.  
Couleur en français ? ...  
Traduction anglaise : ...  
...  
Couleur en français ? quitter  
Merci d'avoir utiliser ce service et à bientôt.
```

2. Écrire un programme `traducteur_couleurs_en_fr.py` faisant l'inverse, c'est-à-dire traduisant une couleur de l'anglais vers le français, avec un message d'accueil, la répétition du service tant que l'utilisateur ne souhaite pas l'interrompre, et avec un message de sortie (comme pour 1.b).

```
Bienvenue dans ce formidable outil de traduction anglais-français.  
Couleur en anglais ? green  
Traduction française : vert  
...
```

3. Écrire un programme `traducteur_couleurs_auto_fr_en.py` faisant comme les précédents (en boucle) mais détectant aussi automatiquement la langue source (en essayant par le français en premier).

```
Bienvenue dans ce formidable outil de traduction français-anglais ou vice-versa.  
Couleur en français ou anglais ? green  
Mot anglais dont la traduction française est : vert  
Couleur en français ou anglais ? jaune  
Mot français dont la traduction anglaise est : yellow  
...
```

4. Modifier le programme `traducteur_couleurs_auto_fr_en.py` de façon qu'il permette en plus à l'utilisateur de "remplir" (dynamiquement) le dictionnaire (la base de données) lorsque le mot saisi est inconnu¹. Exemple :

```
Couleur en français ou anglais ? ivoire  
Désolé, je ne connais pas. À quelle langue appartient-il ? français  
Traduction anglaise ? ivory  
Merci d'avoir ajouter cette traduction. On reprend...  
Couleur en français ou anglais ? ivory  
Mot anglais dont la traduction française est : ivoire  
...
```

1. Vous trouverez des exemples à l'adresse https://fr.wikipedia.org/wiki/Liste_de_noms_de_couleur

3 Tri fusion

Consigne : Créez un fichier `exercice_3.py` dans lequel vous écrirez vos fonctions. Attention à bien respecter le nom des fonctions.

1. En utilisant la fonction `randint` du module `random`, écrire une fonction `random_list(l, n)` qui renvoie une liste de taille `l` dont les éléments sont des entiers générés de manière aléatoire entre 0 et `n`.
2. Écrire une fonction `split(L, n)` qui coupe `L` en deux listes `L1` et `L2` de sorte que `L1` soit de longueur `n` et qui renvoie ces deux listes. Par exemple

`split([4,9,11,6,2,11], 4) = ([4,9,11,6], [2,11]).`

3. Écrire une fonction `fusion(L1, L2)` qui prend en entrée deux listes triées et qui renvoie l'union triées des deux listes. Par exemple

`fusion([1,3,9], [2,3,5,7]) = [1,2,3,3,5,7,9].`

4. Écrire une fonction `tri_fusion(L)` qui prend en entrée une liste et qui la trie en utilisant les fonctions `split` et `fusion`. Schématiquement, la fonction `tri_fusion` fonctionne ainsi: elle commence par séparer `L` en deux moitiés. Ensuite, elle tri chacune des moitiés en s'appelant elle-même (c'est donc une fonction récursive). Finalement, elle fusionne chacune des moitiés triées pour obtenir la liste `L` triée.