
Licence 1 Informatique – 2022/2023 – S2

Algorithmique et programmation Python

TD 6 : Dictionnaires

Rappel des consignes :

- Créez un dossier pour la séance.
- Regroupez les fichiers Python « .py » dans ce dossier.
- 1 exercice = 1 fichier Python
- Le nom du fichier doit être en relation avec le numéro de l'exercice (« exercice_1.py », etc.)
- Rédigez en parallèle les algorithmes pour chaque exercice en pseudo-code dans un fichier word unique.
- A la fin de l'heure, vous regroupez les fichiers du dossier (word unique et .py pour chaque exercice) dans une archive (zip, 7z, rar, autre...)
- A rendre sur votre espace Moodle.
- N'oubliez pas de commenter abondamment votre code source Python.

Exercice 1 – Manipulation d'un dictionnaire

Ecrire une application qui doit effectuer les tâches suivantes :

- Créer un dictionnaire vide.
- Demander à l'utilisateur d'effectuer une saisie qui correspondrait à une clé :
 - S'il a entré 'STOP', nous sortons de la boucle de saisie.
 - S'il a entré une chaîne qui existe déjà dans les clés du dictionnaire, vous affichez un message d'erreur, et vous continuez à l'itération suivante de la saisie
 - Dans les autres cas, vous demandez une seconde saisie qui correspondrait à une valeur (toujours chaîne de caractères). Vous ajoutez alors le couple « clé – valeur » dans le dictionnaire.
- A la sortie de la boucle de saisie, si le dictionnaire n'est pas vide, vous affichez tour à tour : la liste des clés, la liste des valeurs, la liste des couples « clés – valeurs ».

Remarque : (1) `keys()` permet d'obtenir la liste des clés d'un dictionnaire (ex. si le dictionnaire s'appelle dico, `dico.keys()` fournit la liste) ; (2) l'opérateur « `in` » permet de savoir si un élément est dans une liste ; (3) `values()` permet d'obtenir la liste des valeurs d'un dictionnaire.

Exercice 2 – Liste de mots (chaînes de caractères)

Ecrire une application qui effectue les tâches suivantes :

- Demander à l'utilisateur d'entrer une liste de mots (de chaînes de caractères). Les doublons sont possibles.
- Tous les mots doivent être convertis en majuscule (voir https://www.w3schools.com/python/ref_string_upper.asp).

- c. La saisie de la liste doit continuer tant qu'il n'a pas entré le mot « STOP ».
- d. Afficher les termes et leur nombre d'apparition (par exemple, si l'utilisateur a saisi [« TOTO », « TATA », « TOTO », « TUTU », « TOTO »], votre programme doit afficher TATA = 1, TOTO = 3, TUTU = 1

Remarque : (1) Cet exercice était déjà proposé lors du TD n°4, il s'agit cette fois-ci de le résoudre en passant par une structure dictionnaire de Python. (2) Une stratégie possible consisterait à partir d'un dictionnaire vide. De rajouter une nouvelle saisie en clé si elle n'est pas présente, en lui attribuant la valeur 1 (numérique). Ou, si elle existe déjà, d'incrémenter (augmenter de 1) la valeur correspondante.

3 - Liste de personnes

Ecrire une application qui :

- (1) Demande à l'utilisateur un nombre de personnes à traiter ;
- (2) Pour chaque personne, on lui fait saisir son « nom », sa « taille » (en cm) et son « poids » (en kg). Les informations doivent être stockées dans une structure de type dictionnaire.
- (3) A la fin de la saisie, la liste des personnes est affichée, avec pour chacun d'entre eux : son « nom », sa « taille », son « poids » et son « IMC » ([indice de masse corporelle](#)).
- (4) Indiquez la personne qui présente l'IMC le plus faible

Remarque : une liste de dictionnaires est conseillée pour stocker l'ensemble des informations.