TP1. Fonctions : Dictionnaires – Listes – Tuples

Introduction

Dans ce TP nous manipulerons les **chaînes de caractères**, les **Dictionnaires** et des **Listes** ainsi que les **fonctions.**

Créez sur votre **p :** dans le répertoire **Python**, un sous-répertoire de travail nommé **RES208**, puis un sous-répertoire nommé **TP1** dans lequel vous sauvegarderez vos scripts Python.

Avertissement

Pour que votre code soit valide, il vous faut respecter le nom de la fonction, les paramètres (ou arguments) qui lui sont associés et le type du résultat.
IMPORTANT : dans le programme, on séparera bien la Définition des fonctions du Programme principal par des commentaires **# Définition de la fonction** … puis **# Programme Principal** ….

1. Gestion des personnes d'une entreprise

Dans ce programme nous utiliserons des **conteneurs**

Attention : Pour ce programme**,** on pourra **utiliser** le **for element in liste**.

Un **employé** d'une entreprise est caractérisé par son **prénom**, son **nom** et son **age**. Pour stocker ces informations, vous pensez au conteneur **Dictionnaire**, qui aura comme **clefs** `"prénom"`, `"nom"`, et `"age"`.

Dans le programme nommé **gestionEntreprise.py** initialiser **3 employés** que vous stockerez dans les variables **individu1**, **individu2**, **individu3** qui seront des **Dictionnaires.** Ces instructions sont à écrire dans le programme principal.

On donne les informations sur les individus

	Prénom	Nom	Age
individu1	John	DEUF	30
individu2	Aline	HEA	22
individu3	Tom	EIGERY	18

On va regrouper ces informations dans une **liste** car ce conteneur permet de gérer facilement l'**ajout**, la **suppression**, …. des employés de l'entreprise.

1.1 Dans le programme principal, **définissez** la liste nommée **entreprise** constituée des 3 individus du précédent tableau.

1.2 Ecrivez la procédure **afficheMenu( )** qui permet d'afficher le menu ci-dessous :

****** Menu de la Gestion d'entreprise ******
1 : Lister les personnes
2 : Ajouter une personne
3 : Enlever une personne
f : pour quitter le programme

1.3 Définissez la procédure **listerLesPersonnes(uneEntreprise)** qui permet d'afficher toutes les informations (prénom, nom, âge) des employés de l'entreprise. Bien réfléchir au **type de boucle** à utiliser.

1.4 Définissez la fonction **estPresent(unTel, uneEntreprise)** qui retourne les deux valeurs suivantes dans un **Tuple**:

- un **booléen** qui est vrai si l'employé **unTel** est effectivement présent dans la variable **uneEntreprise**, et faux dans le cas contraire.
- un **indice** qui correspond à la place (indice) de l'employé dans la liste : cet indice sera nécessaire lors de l'appel **enleverUnePersonne( )**

Remarque : on ne testera que le nom et le prénom, et bien réfléchir au type de boucle à utiliser si la liste comptait 500 individus ou plus

1.5 Définissez la procédure **ajouterUnePersonne(uneEntreprise)** qui ajoute l'employé à la variable **uneEntreprise** en fin de la liste. Les informations concernant cet employé seront rentrées par l'utilisateur. Bien sûr, on n'ajoutera cette personne que si elle n'est pas dans l'entreprise. On fera un test avec une personne dans l'entreprise et une autre personne comme par exemple l'individu4

individu4 = {"prénom":"Alain", "nom":"PROVISTE","age":25}

1.6 Définissez la procédure **enleverUnePersonne(uneEntreprise)** qui supprime toutes les informations relatives à la personne qui quitte l'entreprise. Il faudra bien sûr tester si l'employé est présent dans l'entreprise, et donc connaître sa position (indice) dans la liste.

Testez toutes les fonctionnalités de votre programme, en listant, ajoutant ou supprimant des employés.

2. Listes : recherche d'un max, éliminer les doublons, …

On donne la liste suivante : liste = [**14**, 7, **6**, 19, **2**, **6**, **3**, 7, **3**, 8, **2**, **14**, 10, **2**, 8]

Bien réfléchir au **type de boucle** à utiliser dans les fonctions ci-dessous :

Créez le programme liste_Doublons.py qui définit la fonction **max_Liste(…)** permettant de rechercher la **valeur maximale d'une liste** d'entiers et de l'afficher dans le programme principal.

Attention : on n'utilisera pas la fonction **max(liste)** qui permet d'avoir directement le résultat. On cherche à écrire des algorithmes, on écrira donc cette fonction.
On pourra utiliser **len(liste)** qui retourne le nombre d'éléments de la liste.

Puis définissez la fonction **sans_Doublon( )** qui retourne une liste dans laquelle n'apparait pas deux fois la même valeur, tout en conservant l'ordre d'apparition de ces valeurs dans la liste. **Bien sûr, il ne faudrait pas utiliser set(liste) !!!**

Remarque : on pourra utiliser le **for element in liste**, et le **if element not in liste** ➔ Version 1
Dans la **Version 2** on cherchera un autre algorithme n'utilisant pas cette facilité liée à Python.

La liste sans doublon retournée devra être égale à listeBis = [14, 7, 6, 19, 2, 3, 8, 10]

Tester votre programme.