Python et Unix : TP 1

Préambule :

L'ensemble des exercices traités en TP s'inscrit dans une continuité, vous vous appliquerez à écrire aussi proprement (modularité) que possible les solutions que vous élaborerez. Les TPs suivants seront construits en réutilisant le code que vous développerez au cours des premières séances.

L'ensemble des informations nécessaires à la résolution des exercices est disponible dans les transparents de cours ou sur www.python.org en section documentation pour une information plus pointue.

L'objectif de ce premier TP est de vous familiariser avec le langage python, en introduisant les divers types de base (string, int, list et dict) ainsi que les entrée (raw_input) et sorties (print) en mode console.

Exercice 1 : Prise en main

Lancez l'interprète python en mode interactif avec la commande "python". Déclarez une variable entière, une variable string, une variable list. Manipulez ces variables, évaluez les, réaffectez les. Affectez a une variable à la valeur d'une autre (var1 = var2). Modifiez en une et examinez le résultat. Entrez quelques expressions arithmétiques.

Exercice 2 : Premier script

Ecrivez un script dans un fichier portant l'extension .py, en première ligne utilisez *#! /usr/bin/python* et *chmod +x* pour rendre votre script exécutable.

Le but de cet exercice est de vous familiariser avec les types de données python et les entrées sorties. On vous demande de concevoir une petite application d'agenda permettant de stocker des numéros de téléphone.

Question 1: l'Annuaire

L'association d'un nom de personne à son numéro de téléphone se fait naturellement à l'aide du type dictionnaire. Créez un dictionnaire contenant les entrées (toto->123) et (tata->456). Construisez le de deux manières différentes :

1. En l'initialisant avec son contenu
2. En l'initialisant vide et en ajoutant les entrées

Enfin prévoyez un affichage (print) qui ait la forme :
toto : 123
tata : 456

NB: les types de données de python sont directement imprimables à l'aide de print.

Question 2 : Plusieurs numéros par personne

Dans un nouveau fichier, créez une liste qui va permettre de contenir les numéros associés à une personne. Insérez dans la liste les entrées [123, 456 , 789] de plusieurs façons différentes.
A l'aide de l'opérateur *slice* list[a:b], imprimez :

1. Tous les éléments
2. Les deux et troisième éléments
3. Le premier élément
4. Le dernier élément

Question 3 : Interaction utilisateur

A présent nous allons ajouter une interaction avec l'utilisateur : dans un nouveau fichier initialisez un dictionnaire vide, puis à l'aide de *raw_input* écrivez un script qui demande à l'utilisateur un nom et un numéro de téléphone et l'insère dans la table. L'insertion sera ensuite confirmée à l'utilisateur. Dans un premier temps on considérera que les personnes n'ont qu'un numéro de téléphone.
La sortie de votre script aura l'aspect suivant :
Entrez un nom : ***toto***
Entrez un numéro de tél. : ***1234***

Vous avez inséré toto : 1234
A bientot

Question 4 : Boucle d'interaction

Au lieu de se terminer après une insertion, modifiez votre script pour que les questions à l'utilisateur se fassent dans une boucle infinie, permettant d'ajouter plusieurs entrées à l'agenda. La terminaison du programme sera obtenue à l'aide de *ctrl-c.*

Modifiez votre script pour que l'action se termine si l'utilisateur entre une ligne vide (nom ou numéro), permettant une sortie propre.

Question 5 : Dictionnaires et listes

Ajoutons la possibilité d'associer plusieurs numéros à une personne: modifiez votre script pour demander plusieurs numéros de téléphone. Votre programme demandera des numéros de téléphone jusqu'à ce que l'utilisateur entre une ligne vide. Une ligne de "Entrez un nom:" vide terminera le programme.
Les numéros seront stockés dans votre dictionnaire. Notez que python permet d'associer une liste d'objets à une clé dans un dictionnaire.
Attention au passage par copie, les listes sont passées par référence en python :

```
a = [1,2,3]
b = a   #aliasing
a[1] = -1
print b
        [1,-1,3]
```

Pour effectuer une copie de liste il faut utiliser l'opération *b =a[:]*
Imprimez le contenu complet de l'agenda à la terminaison du script.

[i]Question 6 : Définition de fonction

Nous allons améliorer l'interaction utilisateur en lui proposant un menu :
Agenda python
1: Ajouter entrée
2: Afficher agenda complet
3: Supprimer entrée
4: Quitter
Quel est votre choix :

Ecrivez **une fonction** sans arguments qui imprime ce texte et rends la valeur saisie par l'utilisateur. Testez cette fonction en l'appelant une fois et en imprimant le résultat du choix.

Améliorez cette fonction pour rattraper les saisies incorrectes:
Quel est votre choix: zrgd
Choix invalide ! : zrgd. Recommencez :
Agenda Python ....

Question 7 : Intégration

Reprenez le code de la question 5 en le plaçant dans une fonction afin d'implémenter le choix numéro 1.
Ecrivez une fonction qui réalise le choix 2, à l'aide d'une boucle *for*.

Réalisez également le choix 3 dans une fonction, qui prendra cette forme :
*Quelle entrée supprimer ?:**TITI***
Désolé, TITI n'est pas référencé dans l'agenda.
***OU** L'entrée associée à TITI à été supprimée de l'agenda.*

Enfin intégrez les appels à ces fonctions dans une boucle d'interaction infinie, qui s'interrompt (break) si l'utilisateur choisit l'option **4**

[i] Ce sujet est librement inspiré du support proposé par l'université sciences cognitives de Sussex (GB) en intiation à python.