

INFO2013 - Hiver 2020

Travail de Laboratoire 6 - Python : Sets et Dictionnaires (26/02/2020)

Partie 1 : Environnement de développement

- Environnement de développement : Vous pouvez utiliser l'application web Jupyter Notebook, en ligne (voir Laboratoire 3), ou bien repl.it (voir laboratoire 4).
- **repl.it** est plus recommandé ici. (Jupyter est l'alternative en cas où repl.it est hors service).
- Si vous utilisez repl.it, vous pouvez remettre votre travail **par courriel en envoyant les liens repl.it** qui sont générés lorsque vous créez des repl Python (copiez les liens de la barre d'adresse du navigateur web, ex. <https://repl.it/repls/UnluckyTotalDrawings>).

Partie 2 : programmes

Exercices 1 :

Écrire un programme Python qui permet de vérifier si une liste contient des duplicatas. Le programme doit permettre de saisir la liste et afficher un message indiquant s'il y'a des éléments qui se répètent dans la liste.

Exemples :

```
Liste1 : [1, 2, 3, 1, 4, 3] # la liste contient des duplicatas
```

```
Liste2 : ['a', '2', 5, 6] # la liste ne contient pas des
```

duplicatas

Testez votre programme lorsque vous finissez.

Exercice 2 :

Modifier le programme précédent afin qu'il affiche, cette fois, tous les éléments qui se répètent dans la liste.

```
Exemple : Liste : [1, 2, 3, 1, 4, 3, 5] # affiche [1, 3]
```

Exercice 3 :

Écrire un programme qui permet de gérer les résultats d'un tournoi de soccer entre plusieurs équipes. Le programme doit permettre de saisir les noms des équipes, puis saisir les résultats des matchs (3 points pour l'équipe gagnante, 0 pour l'équipe perdante, et 1 point pour les parties nulles) ensuite, afficher les équipes en ordre de leurs scores respectifs.

Les grandes lignes de votre programme peuvent être comme suit :

- Saisir la liste des noms des équipes. (*liste = (input('Donnez les noms des équipes : ')).split()*)
- Créer un dictionnaire à partir de cette liste pour les clés, et 0 pour les valeurs. Utilisez, pour ceci, les listes de compréhension. (*equipes_scores = {x : 0 for x in liste}*)
- Pour chaque deux équipes, demander à l'utilisateur de saisir le nom de l'équipe gagnante. S'il y a eu égalité, l'utilisateur ne saisit rien. Ajouter les points nécessaires aux équipes en mettant à jour les valeurs correspondantes dans le dictionnaire.
- Afficher les équipes en ordre décroissant de leurs scores. (Utilisez la fonction **sorted()** et une liste de compréhension : *tuple_trie = sorted((score, nom) for (nom, score) in equipes.items()) [::-1]*)

Exemple :

Saisir les noms des équipes : **a b c**

Qui a gagné entre a et b? **a** # **a** avait gagné **b**

Qui a gagné entre a et c? # égalité entre **a** et **c**

Qui a gagné entre b et c? **b** # **b** avait gagné **c**

Résultat :

a: 4
b: 3
c: 1

Annexe

Nom(s) et prénom(s)

.....
.....

Code (vous n'avez pas besoin de remettre ce document si vous envoyez les liens repl.it par courriel):