

Les Structures de Données Python

TP2 - Les sets et les dictionnaires

Halim Djerroud

révision 0.1

Partie 1 :

Exercice 1

1. Construire un dictionnaire « Semestre2 » des 6 premiers mois de l'année en cours avec comme valeurs le nombre de jours respectif.
2. Afficher la liste des mois et celle des jours.
3. Afficher la liste des mois et celle des jours.
4. Faire un test si l'année est bissextile et mettre à jour le nombre de jour pour le mois de février.
5. Construire le dictionnaire « Semestre1 » qui commence du 4 septembre pour une durée de 6 mois avec comme valeurs le nombre de jours respectifs.
6. Ajouter dans un dictionnaire « AnnéeScolaire » les éléments du « Semestre1 » et ceux du « Semestre 2 »
7. Créer le dictionnaire « VacancesScolaire » qui associe à chaque mois le nombre de jours de vacances selon le calendrier des vacances scolaires de l'année 2021-2022
8. Créer manuellement le dictionnaire « weekend » qui associe à chaque mois le nombre de jours de weekend
9. Créer le dictionnaire « JoursTravail » qui associe à chaque mois le nombre de jours de travail effectif.

Exercice 2

Écrire puis tester une fonction qui combine deux dictionnaires en un seul en additionnant les valeurs ayant les mêmes clefs. Exemple :

```
Entrée :  
d1 = {'a': 100, 'b': 200, 'c':300}  
d2 = {'a': 300, 'b': 200, 'd':400}  
Sortie :  
s{'a': 400, 'b': 400, 'd': 400, 'c': 300}
```

Exercice 3

Considérons le dictionnaire imbriqué suivant qui est composé de clés de type entier et de valeurs qui sont à leur tour de type dictionnaire :

```
personnes = {1 : {'nom': 'John', 'age': '27', 'genre': 'Homme'},  
             2 : {'nom': 'Marie', 'age': '22', 'genre': 'Femme'},  
             }
```

1. Ajoutez deux nouvelles personnes au dictionnaire « personnes »
2. Modifiez le nom de la personne ayant pour clé 1 de John à Jonathan
3. Supprimez la personne ayant pour clé 2.
4. Affichez toutes les paires de clés - valeurs du dictionnaire « personnes » à l'aide d'une boucle for. Assurez-vous de bien formater votre affichage.

Exercice 3

Imaginez que vous travaillez actuellement dans une entreprise informatique et que vous recevez la liste d'employés suivante.

Nom	Age	Département
Jean Macke	38	Vente
Lisa Crawford	29	Management
Sujan Patel	33	RH

Votre responsable vous demande d'utiliser Python pour stocker ces données en vue d'une utilisation ultérieure par l'entreprise.

1. Utiliser une liste de dictionnaires pour stocker ces 3 employés.
2. Proposer à l'utilisateur un menu lui permettant :
 - (a) D'afficher l'ensemble des employés de la liste sous la forme suivante :

```

Nom : Jean Mackee
Âge : 38
Département : Vente
-----
Nom : Lisa Crawford
Âge : 29
Département : Management
-----
Nom : Sujan Patel
Âge : 33
Département : RH
-----

```

- (b) D'afficher les informations détaillées d'un employé dont le nom est donné par l'utilisateur (penser à utiliser une fonction)
- (c) D'afficher l'ensemble des employés ayant un âge inférieur à une valeur a donnée par l'utilisateur.

Exercice 4

Voici une base de données reportant le nombre de ventes réalisées par des employés dans quatre différentes régions géographiques.

	Nord	Sud	Est	Ouest
Léa	4568	1540	6548	4154
Marie	848	1254	8878	4110
Julie	4188	4042	798	4487
Marc	1544	4785	743	1548

1. En utilisant un dictionnaire imbriqué (dictionnaire de dictionnaire), représenter les informations du tableau ci-dessus.
2. Écrire une fonction qui à partir d'un nom d'employé et d'une région, elle retourne le nombre de ventes réalisées.
3. Ajouter une fonction qui permet à partir d'un nom d'employé et d'une région de modifier le nombre de ventes associé.
4. Ajouter une fonction qui permet d'afficher toutes les ventes réalisées dans une région géographique donnée.

Partie2

Exercice 1

Le but ici est de créer un moyen d'indexation des mots d'un livre ou d'un document. Pour cela, nous souhaitons implémenter un dictionnaire **Index** où les clés représentent les mots et les valeurs seront la liste des numéros de pages dans lesquelles figurent les mots. Les numéros de pages doivent figurer de façon unique pour chaque mot.

1. Écrire une fonction `Add` qui prend en paramètre un mot et un numéro de page et qui permet de l'ajouter à l'index lorsque cela est possible.
2. Écrire une fonction `remove` qui prend en paramètres un mot et qui permet de le supprimer (ainsi que toutes les pages dans lesquelles il figure) de l'index.
3. Écrire une fonction `search` qui prend en paramètre un mot et qui permet de retourner toutes les pages dans lesquelles il figure.
4. Écrire une fonction `display` qui affiche le contenu de l'index trié par ordre alphabétique et dont les numéros de pages sont triés dans l'ordre croissant. Exemple d'affichage :

```
Liste : 8, 50  
Python : 12,25  
Tuple : 30
```

5. Tester les 4 fonctions précédentes dans un programme principal.

Exercice 2

À partir de l'index précédent, nous souhaitons générer un index inverse **reverse_index** qui permet d'associer à chaque numéro de page la liste des mots qu'elle contient.

1. Écrire une fonction `create_reverse_index` qui prend en paramètre un index et qui permet de créer un **reverse_index**.
2. Écrire une fonction `display_reverse_index` qui permet d'afficher le contenu de **reverse_index** par ordre croissant sur les numéros de page, l'ensemble de mots pour chaque page doit être affiché par ordre alphabétique. Exemple d'affichage de cette méthode :

```
10 : Python  
20 : Python tuple  
30 : dictionnaire  
40 : Langage Python
```