

TP2 sur les dictionnaires

En cas de difficultés, pensez à relire la leçon avant de demander de l'aide au professeur.

I. EXERCICES D'ÉCHAUFFEMENT

. EXERCICE 1 : ACCÈS, MODIFICATION ET PARCOURS

Trois amis jouent à un jeu. Mathéo a 30 points, Youssef a 50 points et Sophie a 40 points.

1. Créer un dictionnaire `sc` ayant pour clés les noms des joueurs et ayant pour valeurs leurs scores
2. Quelle instruction permet d'ajouter 10 points à Mathéo ?
3. Quelle instruction permet d'afficher le nouveau score de Mathéo ?
4. Recopier et compléter la fonction `gagnant` qui reçoit ce type de dictionnaire et renvoie le nom du gagnant.

```
def gagnant( scores : dict ) :  
    premier = None  
    for joueur in ..... :  
        if premier == None : premier = .....  
        elif scores[ joueur ] > scores[ premier ] : premier = .....  
    return premier
```

5. Vérifier que `gagnant(sc)` renvoie bien Youssef.
6. Tester cette fonction avec un autre dictionnaire de scores.

. EXERCICE 2 : CRÉATION PAR COMPRÉHENSION

1. Que fait cette ligne ? `t = [chr(65 + randint(0,25)) for i in range(7)]`
.....
.....
2. Créer un dictionnaire `table_7` ayant pour clés les nombres entiers de 1 à 10 et pour valeurs leurs produits par 7. Afficher ce dictionnaire pour vérifier son contenu.
3. Quelle instruction permet d'accéder à la valeur associée à la clé 5 ?
4. Quelle instruction permet de vérifier si 36 fait partie des valeurs de ce dictionnaire ?
.....

II. EXERCICES INSPIRÉS DU LIVRE « APPRENDRE LA PROGRAMMATION PAR LE JEU » DE VINCENT MAILLE, ÉDITION CRÉATIONS NUMÉRIQUES

. EXERCICE 3 : ADAPTATION DE L'EXERCICE B1 PAGE 49

Le **Scrabble** (marque déposée) est un jeu de société et un jeu de lettres où l'objectif est de cumuler des points, sur la base de tirages aléatoires de lettres, en créant des mots sur une grille carrée, dont certaines cases sont primées.

Extrait de Wikipedia

1. Créer une fonction `lettres` qui reçoit en entrée une chaîne de caractère et renvoie un dictionnaire dont les clés sont les lettres présentes dans cette chaîne de caractère et dont les valeurs sont leurs occurrences.

2. Vérifier que `lettres("Bonjour")` renvoie bien le dictionnaire ci-dessous.
`{'B' : 1 ; 'o' : 2 ; 'n' : 1 ; 'j' : 1 ; 'u' : 1 ; 'r' : 1 }`
Rappel : Il n'y a pas de notion d'ordre dans un dictionnaire.
3. Quel dictionnaire est renvoyé par l'appel `lettres("Programmation")` ?
.....
4. Créer une fonction `tirage()` qui choisit sept lettres au hasard dans l'alphabet et renvoie un dictionnaire dont les clés sont les différentes lettres choisies et dont les valeurs sont le nombre de fois où chacune a été choisie.
Aide : Utiliser la question 1 de l'exercice 2 ainsi que la fonction `lettres`.
5. Traduire en Python l'algorithme suivant :

Faire un appel à la fonction <code>tirage</code> et stocker la valeur renvoyée dans la variable <code>chevalet</code> Afficher le contenu de la variable <code>chevalet</code> Initialiser une variable <code>gagner</code> à <code>False</code> et une variable <code>essai</code> à 0. Tant que <code>gagner</code> vaut <code>False</code> et que <code>essai</code> est inférieur à 3 Demander à l'utilisateur de proposer un mot Ajouter 1 au contenu de la variable <code>essai</code> Vérifier si ce mot peut être écrit avec le contenu du <code>chevalet</code> Si oui, affecter <code>True</code> à la variable <code>gagner</code> .
--
6. Tester votre programme

. EXERCICE 4 : ADAPTATION DE L'EXERCICE B6 PAGE 53

On s'intéresse à une partie d'échecs. Il ne reste plus que quatre pièces noires : un premier fou en (1 ; 5), un deuxième fou en (8 ; 5), une tour en (4 ; 2) et le roi en (4 ; 8). On souhaite vérifier si un déplacement donné est valide.

1. Créer un dictionnaire `plateau` ayant pour clés les positions des pièces et pour valeurs leurs types.
2. Écrire une fonction `deplacement` qui reçoit deux cases sous forme d'un couple et renvoie un booléen indiquant si ce déplacement est autorisé ou non. S'il n'y a pas de pièce sur la case de départ, la fonction renverra `False`. Sinon elle vérifiera si le déplacement est compatible avec la pièce présente sur la case de départ.
Aide : La tour ne peut se déplacer que horizontalement ou verticalement, d'autant de cases qu'elle le souhaite. Les fous ne peuvent se déplacer qu'en diagonale, d'autant de cases qu'ils le souhaitent. Le roi ne peut se déplacer que sur l'une des huit cases adjacentes à la sienne.
3. Vérifier que `deplacement((7,3), (8,3))` renvoie `False`
Vérifier que `deplacement((1,5), (2,4))` renvoie `True`
4. Proposer deux autres tests