

Exercices sur les dictionnaires

Exercice 1 : Predict, run, investigate.

Sans exécuter, qu'affichent les codes suivants ? Vérifier en exécutant le code.

```
# Code 1
fruits = {'pommes': 4, 'melons': 3, 'poires': 6, 'clémentines': 16}
for k in fruits.keys():
    print(k)
```

```
# Code 2
fruits = {'pommes': 4, 'melons': 3, 'poires': 6, 'clémentines': 16}
for k, v in fruits.items():
    print(k, "->", v)
```

```
# Code 3
fruits = {'pommes': 4, 'melons': 3, 'poires': 6, 'clémentines': 16}
for v in fruits.values():
    print(v)
```

Exercice 2 : Modify.

1) Modifier le code 2 de l'exercice 1 pour afficher uniquement le nom des fruits dont la quantité est strictement plus grande que 3.

Exercice 3 : Make.

1) On considère le dictionnaire suivant : `notes = {"Alice": 14, "Bob": 9, "Charlie": 16}`

- 1) Donner l'instruction permettant d'afficher la note de Bob.
- 2) Donner l'instruction permettant d'ajouter la note de Diane : 12.
- 3) Donner l'instruction permettant de modifier la note d'Alice : 15.

2) Voici deux dictionnaires :

```
athletes = {"Mike": (1.75, 68), "John": (1.89, 93), "Kate": (1.67, 62)}
sportifs = {"Mike": {"taille": 1.75, "poids": 68}, "John": {"taille": 1.89, "poids": 93}, "Kate": {"taille": 1.67, "poids": 62}}
```

- 1) De quel type sont les clés des deux dictionnaires `athletes` et `sportifs` ?
- 2) De quels types sont les valeurs de ces deux dictionnaires ?
- 3) Quelle instruction permet d'accéder à la taille de Kate dans le dictionnaire `athletes` ?
- 4) Quelle instruction permet d'accéder à la taille de Kate dans le dictionnaire `sportifs` ?

3) On dispose du dictionnaire suivant :

```
capitales = {"France": "Paris", "Italie": "Rome", "Espagne": "Madrid"}
```

- 1) Avec une boucle, afficher chaque pays et sa capitale.
- 2) Avec une boucle, afficher uniquement les capitales.
- 3) Avec une boucle, compter le nombre d'éléments du dictionnaire. (interdiction d'utiliser la fonction len)

4) On dispose d'un dictionnaire associant un élève à sa note.

```
notes = {"Alice": 14, "Bob": 9, "Charlie": 16, "Diane": 12}
```

Écrire un programme qui calcule et affiche la moyenne des notes.

5) Le Scrabble est un jeu de société où l'on doit former des mots avec tirage aléatoire de lettres, chaque lettre valant un certain nombre de points. Le dictionnaire scrabble contient cette association entre une lettre et son nombre de points.

```
scrabble = {'A': 1, 'B': 3, 'C': 3, 'D': 2, 'E': 1, 'F': 4, 'G': 2, 'H': 4, 'I': 1, 'J': 8, 'K': 10, 'L': 1, 'M': 2, 'N': 1, 'O': 1, 'P': 3, 'Q': 8, 'R': 1, 'S': 1, 'T': 1, 'U': 1, 'V': 4, 'W': 10, 'X': 10, 'Y': 10, 'Z': 10}
```

Écrivez une fonction nb_points(mot) qui prend en paramètre une chaîne de caractères majuscules mot et qui renvoie le nombre de points au scrabble de mot.

```
print(nb_points('ARBRE')) # affiche 7  
print(nb_points('XYLOPHONE')) # affiche 32
```

6) On dispose du dictionnaire :

```
annuaire = {"Alice": "0601020304", "Bob": "0611223344"}
```

On souhaite écrire un programme qui demande un nom à l'utilisateur puis affiche son numéro s'il est présent dans l'annuaire ou affiche un message d'erreur sinon.

- 1) Écrire une première version sans parcours.
- 2) Écrire une version avec un parcours.

7) Écrire une fonction comptage_occ_mot qui prend en paramètre une liste de mots et construit un dictionnaire indiquant le nombre d'occurrences de chaque mot.

```
l = ["chat", "chien", "chat", "oiseau", "chien", "chat"]  
comptage_occ_mot(l) # affiche {"chat": 3, "chien": 2, "oiseau": 1}
```

8) Écrire une fonction comptage_occ_lettre qui prend en paramètre un mot et construit un dictionnaire indiquant le nombre d'occurrences de chaque lettre.

```
comptage_occ_lettre("cool") # affiche {'c': 1, 'o': 2, 'l': 1}
```

9) On dispose d'un dictionnaire associant un élève à sa note.

```
notes = {"Alice": 14, "Bob": 9, "Charlie": 16, "Diane": 12}
```

Écrire un programme qui affiche uniquement les noms d'élèves ayant la moyenne.

10) Écrire une fonction `ajouter_note` prenant en paramètres un dictionnaire, un nom et une note et qui :

- ajoute l'élève au dictionnaire s'il n'existe pas

- met à jour sa note dans le dictionnaire s'il existe déjà

```
notes = {"Alice": 14, "Bob": 9, "Charlie": 16, "Diane": 12}
ajouter_note(notes, "Alice", 16)
ajouter_note(notes, "Emilia", 20)
print(notes)
# {'Alice': 16, 'Bob': 9, 'Charlie': 16, 'Diane': 12, 'Emilia': 20}
```

11) On dispose d'un dictionnaire associant un élève à sa note.

```
notes = {'Alice': 16, 'Bob': 9, 'Charlie': 16, 'Diane': 12, 'Emilia': 20}
```

Écrire un programme qui affiche :

- la meilleure note et le nom de l'élève correspondant

- la note la plus basse

12) On dispose du dictionnaire suivant :

```
classe = {
    "Alice": {"maths": 14, "nsi": 15},
    "Bob": {"maths": 15, "nsi": 17},
    "Charlie": {"maths": 10, "nsi": 14}
}
```

1) Quelle instruction permet d'afficher la note de NSI d'Alice ?

2) Calculer la moyenne de chaque élève et afficher le nom de l'élève avec la meilleure moyenne.