

Les dictionnaires

M. Tellene

1 Création de l'environnement de travail

Avant de commencer le TP, vous allez créer votre environnement de travail. Pour ce faire vous allez dans votre dossier personnel. Une fois arrivé, créer un dossier « TP_dictionnaire ». C'est dans ce dossier que vous sauvegarderez les exercices de ce TP.

2 Un dictionnaire, c'est quoi?

Il faut voir un dictionnaire Python comme un dictionnaire de mots (ex : Larousse)



En Python, un dictionnaire est une collection non ordonnée (contrairement aux tableaux ou aux chaînes de caractères qui sont ordonnées¹) d'associations *clef*: *valeur* où chaque **clef est unique**.

Cette clef permet d'accéder à la valeur associée.

Exemple :

```
>>> perso = {"nom": "Bond", "prenom": "James", "profession": "agent secret"}
```

Dans cet exemple :

- la clef `nom` est associée à la valeur `Bond`
- la clef `prenom` est associée à la valeur `James`
- la clef `profession` est associée à la valeur `agent secret`

1. Si vous ne comprenez pas pourquoi, appeler le professeur

3 Comment créer et remplir un dictionnaire?

Il est possible de créer et remplir un dictionnaire de plusieurs manières différentes.

3.1 Instruction `dict()`

On peut initialiser un dictionnaire vide grâce à l'instruction `dict()`. On remplit ensuite ce dictionnaire en définissant les clefs et les valeurs associées à ces clefs.

Exemple :

1. la variable `series` est initialisée comme un dictionnaire vide
2. dans `series`, on associe la valeur `HBO` à la clef `Game of Thrones` et la valeur `BBC One` à la clé `Peaky Blinders`
3. on affiche le dictionnaire rempli

```
1 >>> series = dict()
2
3 >>> series["Game of Thrones"] = "HBO"
4
5 >>> series["Peaky Blinders"] = "BBC One"
6
7 >>> series
8 {'Game of Thrones': 'HBO', 'Peaky Blinders': 'BBC One'}
```

3.2 Accolades vides

On peut aussi initialiser un dictionnaire vide grâce à une paire d'accolades `{}`. On remplit ensuite ce dictionnaire en définissant les clefs et les valeurs associées à ces clés.

Même exemple que précédemment (mais en changeant l'initialisation du dictionnaire) :

1. la variable `series` est initialisée comme un dictionnaire vide
2. dans `series`, on associe la valeur `HBO` à la clef `Game of Thrones` et la valeur `BBC One` à la clé `Peaky Blinders`
3. on affiche le dictionnaire rempli

```
1 >>> series = {}
2
3 >>> series["Game of Thrones"] = "HBO"
4
5 >>> series["Peaky Blinders"] = "BBC One"
6
7 >>> series
8 {'Game of Thrones': 'HBO', 'Peaky Blinders': 'BBC One'}
```

3.3 Définition par extension

On peut également définir un dictionnaire en donnant directement des valeurs aux clefs dans les accolades.

Dans l'exemple ci-dessous, on crée et remplit le dictionnaire en même temps :

```
1 >>> series = {"Game of Thrones": "HBO", "Peaky Blinders": "BBC One"}
```

4 Comment récupérer des données dans un dictionnaire

Étant donné qu'un dictionnaire n'est pas ordonné, il n'est pas possible d'extraire les éléments comme on le ferait avec un tableau ou une chaîne de caractères. Ainsi le code suivant ne marchera pas :

```
1 >>> series = {"Game of Thrones": "HBO", "Peaky Blinders": "BBC One"}
2
3 #je veux récupérer la valeur de "Game of Thrones", comme c'est le premier élé
  ment de mon dictionnaire, j'écris alors :
4 >>> series[0]
```

A l'exécution de ce code, quelle erreur est produite?

.....

L'erreur indique qu'il y a une erreur de clef : la clef 0 n'existe pas dans notre dictionnaire. Ainsi si l'on veut récupérer la valeur de la clef `Game of Thrones`, il faudra écrire :

```
1 >>> series["Game of Thrones"]
```

Attention : il est possible de récupérer la ou les valeurs d'une clef en entrant la clef mais l'inverse n'est pas vrai. Il n'est donc pas possible de récupérer la clef de la valeur `HBO` en écrivant :

```
1 >>> series["HBO"]
```

5 Notion de clef et de valeurs

Dans un dictionnaire :

- une valeur peut être présente plusieurs fois
- une clef n'est présente qu'une seule fois

5.1 Clefs d'un dictionnaire

La clef correspond à l'indice dans un tableau ou un tuple : c'est le nom de l'élément dans le dictionnaire. **Deux associations ne peuvent donc pas avoir la même clef :**

```
1 >>> valise = {"chapeau": 3, "pantalon": 6, "chemise": 6, "chapeau": 2}
2
3 >>> valise
4 {'chapeau': 2, 'pantalon': 6, 'chemise': 5}
```

Lorsqu'on utilise une seconde fois la même clef, on redéfinit la valeur associée à cette clef :

```
1 >>> dico = {}
2
3 >>> d["a"] = 42
4
5 >>> d["b"] = 24
6
7 >>> d
8 {'a': 42, 'b': 24}
9
10 >>> d["a"] = 0
11
12 >>> d
13 {'a': 0, 'b': 24}
```

5.2 Valeurs d'un dictionnaire

Les exemples précédents ont montré qu'il est possible de modifier un dictionnaire (on dit qu'un dictionnaire est muable). On peut modifier les valeurs associées aux clefs d'un dictionnaire : celui-ci reste le même objet.

```
1 >>> series = {"Game of Thrones": "HBO", "Peaky Blinders": "BBC One"}
2
3 # permet de récupérer l'identité de series
4 >>> id(series)
5 139702873427648
6
7 # on modifie la valeur de la clef "Peaky Blinders"
8 >>> series["Peaky Blinders"] = "BBC Two"
9
10 >>> series
11 {'Game of Thrones': 'HBO', 'Peaky Blinders': 'BBC Two'}
12
13 # on peut voir que l'identité de series est toujours la même suite au
14   changement de valeur
15 >>> id(series)
16 139702873427648
```

5.3 Quelques exercices

Créer un fichier **dictionnaire.py**, c'est dans ce fichier que vous écrirez toutes les fonctions demandées.

EXERCICE 1

Écrire une fonction ajout qui prend un argument un dictionnaire, un clef et une valeur. La fonction devra ajouter l'association *clef*: *valeur* dans le dictionnaire et renvoyer ce dernier.

EXERCICE 2

Écrire une fonction dico_voyelles qui ne prend pas d'argument. Cette fonction devra créer et renvoyer un dictionnaire avec en clef les voyelles de l'alphabet et en valeur 0 pour chacune des clefs.

```
1 >>> dico_voyelles()
2 {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0, 'y': 0}
```

EXERCICE 3

Écrire une fonction compte_voyelles qui prend argument une chaîne de caractères. **En utilisant la fonction précédente** dico_voyelles, cette fonction doit compter le nombre d'occurrences de chaque voyelle de la chaîne passée en argument.

Exemple :

```
1 >>> compte_voyelle("salut")
2 {'a': 1, 'e': 0, 'i': 0, 'o': 0, 'u': 1, 'y': 0}
3
4 >>> compte_voyelle("voyelle")
5 {'a': 0, 'e': 2, 'i': 0, 'o': 1, 'u': 0, 'y': 1}
```

6 Comment parcourir un dictionnaire?

Rappel : Un dictionnaire n'est pas une structure ordonnée. Le parcours d'un dictionnaire ne se fera pas nécessairement dans l'ordre dans lequel les associations *clef*: *valeur* ont été entrées, ni dans un quelconque autre ordre.

6.1 Parcours par les clefs

Pour parcourir un dictionnaire par les clefs, il faut utiliser la fonction `keys()`.

Exemple :

```
1 series = {  
2     "Game of Thrones": "HBO",  
3     "Peaky Blinders": "BBC One",  
4     "Big Bang Theory": "CBS "  
5 }  
6  
7 for clef in series.keys():  
8     print(clef, ":", series[clef])
```

Exécuter le code suivant, quel est le résultat produit?

.....
.....
.....

6.2 Parcours par les valeurs

Pour parcourir un dictionnaire par les valeurs, il faut utiliser la fonction `values()`.

Exemple :

```
1 series = {  
2     "Game of Thrones": "HBO",  
3     "Peaky Blinders": "BBC One",  
4     "Big Bang Theory": "CBS "  
5 }  
6  
7 for value in series.values():  
8     print(value)
```

Exécuter le code suivant, quel est le résultat produit?

.....
.....
.....

6.3 Parcours par les clefs et les valeurs

On peut obtenir les associations *clef*: *valeur* avec la fonction `items()` :

Exemple :

```
1 series = {
2     "Game of Thrones": "HBO",
3     "Peaky Blinders": "BBC One",
4     "Big Bang Theory": "CBS "
5 }
6
7 for (clef, valeur) in dico.items():
8     print(clef, ":", valeur)
```

Exécuter le code suivant, quel est le résultat produit?

.....

.....

.....

7 Exercices

EXERCICE 4

Le Scrabble TM, selon Wikipédia, est un jeu de société 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 primées². Les valeurs et les lettres peuvent être rangées dans un dictionnaire :

```
1 scrabble = {"A": 1, "B": 3, "C": 3, "D": 2, "E": 1, "F": 4, "G": 2, "H": 4,
2             "I": 1, "J": 8, "K": 10, "L": 1, "M": 2, "N": 1, "O": 1,
3             "P": 3, "Q": 8, "R": 1, "S": 1, "T": 1, "U": 1, "V": 4,
4             "W": 10, "X": 10, "Y": 10, "Z": 10}
```

Écrire une fonction `points_mot` qui prend un argument une chaîne de caractères, cette chaîne symbolisera un mot (attention la chaîne doit uniquement être composée de caractères majuscules). La fonction devra calculer et renvoyer le nombre de points que rapporterait ce mot au Scrabble TM.

Exemple :

```
1 >>> points_mots("PYTHON")
2 20
3
4 >>> points_mots("LYCEE")
5 16
```

EXERCICE 5

Écrire une fonction `meilleure_lettre` qui prend en argument un tableau de lettre (toujours en majuscule). La fonction devra renvoyer la lettre valant le plus de points au Scrabble TM. En cas d'égalité, ce sera la lettre ayant l'indice le plus petit, dans le tableau donné en argument, qui sera renvoyée.

Exemples :

```
1 >>> meilleure_lettres(["L", "Y", "C", "E"])
2 'Y'
3
4 >>> meilleure_lettres(["A", "B", "C", "D"])
5 'B'
6 # B et C ont le même nombre de point, on renvoie B car il est avant C dans le
   tableau donné en argument
```

2. <https://fr.wikipedia.org/wiki/Scrabble>

EXERCICE 6

On considère le dictionnaire suivant :

```
1 series = {
2     "Game of Thrones": "HBO", "Peaky Blinders": "BBC One", "Merlin": "BBC One",
3     "Big Bang Theory": "CBS", "Sherlock": "BBC One",
4     "NCIS: Los Angeles": "CBS", "Doctor Who": "BBC One", "Chernobyl": "HBO",
5     "Les Sopranos": "HBO", "The Witcher": "Netflix", "MacGyver": "CBS",
6     "La Casa de papel": "Netflix", "Kingdom": "Netflix", "Arcane": "Netflix",
7     "Scorpion": "CBS", "Euphoria": "HBO", "Les 100": "Netflix",
8     "Band of Brothers": "HBO", "Les Mystères de l'Ouest": "CBS",
9     "Umbrella Academy": "Netflix", "The Office": "BBC One"
10 }
```

Écrire une fonction `affichage` qui prend en argument une chaîne de caractères. La fonction devra afficher les clefs qui ont pour valeur la chaîne passée en argument.

Exemple :

```
1 >>> affichage("HBO")
2 Game of Thrones
3 Chernobyl
4 Les Sopranos
5 Euphoria
6 Band of Brothers
```

EXERCICE 7

Considérons le dictionnaire suivant :

```
1 valise = {
2     "lunette de soleil": 1,
3     "bob": 1,
4     "maillot de bain": 3,
5     "serviette": 2,
6     "chaussure": 2,
7     "chemise": 5,
8     "short": 4
9 }
```

Écrire une fonction `total` qui ne prend pas d'argument. La fonction devra calculer et renvoyer :

- le nombre d'éléments différents dans la valise
- le nombre total d'éléments dans la valise

La fonction devrait renvoyer le résultat suivant :

```
1 >>> total()
2 (7, 18)
```

EXERCICE 8

Écrire une fonction `nb_occurrences` qui prend en argument une chaîne de caractères. La fonction devra renvoyer un dictionnaire ayant en clef les différentes lettres de la chaîne et en valeur le nombre de fois qu'elles apparaissent. **Important** : attention à l'espace (la touche). Une chaîne de caractères peut contenir des espaces mais non ne voulons pas qu'ils apparaissent dans notre dictionnaire

Exemple :

```
1 >>> nb_occurrences("bonjour")
2 {'b': 1, 'o': 2, 'n': 1, 'j': 1, 'u': 1, 'r': 1}
3
4 >>> nb_occurrences("allez les bleus")
5 {'a': 1, 'l': 4, 'e': 3, 'z': 1, 's': 2, 'b': 1, 'u': 1}
```

EXERCICE 9

En vous servant de la fonction précédente, écrire une fonction `lettre_maximum` qui prend en argument une chaîne de caractères (ayant la longueur d'un texte). La fonction devra renvoyer la lettre la plus présente dans la chaîne de caractère.

Exemple :

```
1 >>> lettre_maximum("""Vous savez, moi je ne crois pas qu'il y ait de bonne ou
de mauvaise situation. Moi, si je devais résumer ma vie aujourd'hui avec
vous, je dirais que c'est d'abord des rencontres. Des gens qui m'ont tendu
la main, peut-être à un moment où je ne pouvais pas, où j'étais seul chez
moi. Et c'est assez curieux de se dire que les hasards, les rencontres
forment une destinée... Parce que quand on a le goût de la chose, quand on a
le goût de la chose bien faite, le beau geste, parfois on ne trouve pas l'
interlocuteur en face je dirais, le miroir qui vous aide à avancer. Alors ça
n'est pas mon cas, comme je disais la, puisque moi au contraire, j'ai pu :
et je dis merci à la vie, je lui dis merci, je chante la vie, je danse la
vie... je ne suis qu'amour ! Et finalement, quand beaucoup de gens aujourd'
hui me disent "Mais comment fais-tu pour avoir cette humanité ?", et bien je
leur réponds très simplement, je leur dis que c'est ce goût de l'amour ce
goût donc qui m'a poussé aujourd'hui à entreprendre une construction mé
canique, mais demain qui sait ? Peut-être simplement à me mettre au service
de la communauté, à faire le don, le don de soi...""")
2
3 La lettre qui apparaît le plus est la lettre 'e'
```