

Chapitre B.2 - Dictionnaires

I. Définition

Un **dictionnaire** (type **dict**) est une collection d'objets **non-ordonnés**. Un dictionnaire est composé de plusieurs éléments, chacun de ces éléments se compose d'une clé et d'une valeur.

Un dictionnaire est donc un ensemble de paires clé: valeur.

II. Créer un dictionnaire

Pour créer un dictionnaire, on utilise des `_____` (`{ }`) dans lesquelles on écrit l'ensemble des couples (`__, _____`) du dictionnaire.

Exemple :

```
mon_dico = {"clé 1": 1, "clé 2": 2}
```

III. Accéder aux éléments d'un dictionnaire

Pour accéder au élément d'un dictionnaire, on écrit le nom du dictionnaire suivie de `_____` (`[]`) dans lesquelles on écrit la `__` correspondant à la valeur recherchée.

Exemple :

```
mon_dico["clé 1"]
```

IV. Mutabilité des dictionnaires

Les dictionnaires sont des valeurs mutables. Cela signifie que l'on peut modifier, ajouter ou supprimer des valeurs dans un dictionnaire.

A. Modifier une valeur

Pour modifier une valeur dans un dictionnaire, il faut utiliser le mécanisme d'affectation.

Exemple :

```
mon_dict = {'a': 1, 'b': 2, 'c': 4}
mon_dict['c'] = 3
```

B. Ajouter une valeur

Pour ajouter une dans un dictionnaire, il faut utiliser la même instruction que pour modifier une valeur comme si la clé existait déjà.

Exemple :

```
mon_dict = {'a': 1, 'b': 2, 'c': 3}
mont_dict['d'] = 4
```

C. Supprimer une valeur

Pour supprimer un couple (clé, valeur) d'un dictionnaire, on utilise la méthode `pop`. (Une méthode est une fonction que l'on peut appeler à partir d'un objet, ici un dictionnaire)

On met la clé que l'on souhaite supprimer en paramètre.

Exemple :

```
mon_dict = {'a': 1, 'b': 2, 'c': 3}
mon_dict.pop('c')
```

Si la clé passée en paramètre n'existe pas, la fonction lève une exception.

Il est possible de fournir un second paramètre à la fonction. Lorsque ce second paramètre est fourni, la fonction renverra la valeur de ce paramètre dans le cas où la clé est inexistante et ne soulève donc pas d'exception.

Exemple :

```
mon_dict = {'a':1, 'b':2, 'c':3}
mon_dict.pop('d')
mon_dict.pop('d',0)
mon_dict.pop('c',0)
```

V. Parcourir les éléments d'un dictionnaire

Exercice :

- créer un dictionnaire `identite` contenant les couples clés, valeurs suivants : (`Prénom`, `Alan`), (`Nom`, `Turing`), (`Profession`, `Mathématicien`)
- Tester les fonctions suivantes :

```
identite.keys()
identite.values()
identite.items()
```

- Que renvoie chacune de ces fonctions ?

Les fonctions présentées dans l'exercice précédent renvoyant des listes, on peut parcourir les dictionnaires en utilisant ces listes.

Exercice : Ecrire une fonction qui affiche l'ensemble des couples clé-valeur d'un dictionnaire sous la forme suivante :

```
(clé1, valeur1)
(clé2, valeur2)
(clé3, valeur3)
```