

Dictionnaire (type dict)

Partie 1 : Tableaux & Dictionnaires



Code : <https://wooclap.com>

XEIOYT

Objectifs :

- Connaître l'interface et l'implémentation des dictionnaires,
- Créer un dictionnaire vide ou non,
- Accéder à une valeur à partir de sa clé,
- Ajouter un couple clé / valeur,
- Parcourir les clés, les valeurs d'un dictionnaire.

Définition

Définition

Dictionnaire

En informatique, un dictionnaire est une structure de données représentant une séquence finie qui associe une clé et une valeur.

Interface

Dans l'interface d'un dictionnaire, nous pouvons citer les opérations suivantes :

1. Créer un dictionnaire vide ou non,
2. Mesurer un dictionnaire : combien de valeurs comporte-t-il ?
3. Accéder à une valeur à partir de sa clé.
4. Ajouter un couple clé / valeur .
5. Supprimer un couple clé / valeur .
6. Itérer sur les clés, les valeurs et les couples clés / valeurs

Implémentation

• Initialisation d'un dictionnaire

•• Représentation

En python, les dictionnaires sont contenus dans des accolades { ... }

Les clés sont séparées des valeurs par deux points : et les couples clés / valeurs sont séparés par des virgules ,

```
{ clé1 : valeur1 , clé2 : valeur2 }
```



Tous les types de variables ne peuvent pas être utilisés comme clé.

Exemple

Par exemple, le code suivant :

```
1 ma_liste = ["a", "b"]
2 my_dic = {ma_liste: "valeur"}
```

lèvera l'erreur suivante :

```
1 TypeError: unhashable type: 'list'
```

Propriété

Les types immutables peuvent être utilisés alors que les types mutables ne le peuvent pas

Exemples non exhaustifs :

—

—



La possibilité d'utiliser ou non un type comme clé provient de la présence ou non de la méthode `__hash__` sur le type en question. (cf. voir la partie *Principe de l'implémentation*)

● Utilisation

i Un dictionnaire pourrait être vu comme une table d'association entre les clés et les valeurs.

Exemple

```
1 {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
```

Ainsi le dictionnaire ci-dessus, pourrait être vu comme une table d'association entre le nom d'un célèbre informaticiens et sa date de naissance.

Clé		Valeur
Alan Turing	→	1912
Ada Lovelace	→	1815
Grace Hopper	→	1906

●● Dictionnaire vide

— Il est possible de créer un dictionnaire vide de 2 façons différentes :

```
1 # Créer un dictionnaire vide
2 dico = {}
3 dico = dict()
```

●● Dictionnaire non vide

— Il est aussi possible de l'initialiser avec des couples clés / valeurs.

```
1 # Créer un dictionnaire à partir de valeurs
2 informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
```

● Accéder, ajouter et modifier des éléments

●● Accéder à un élément

⚠ Contrairement au tableau, pour accéder à une valeur, on n'utilisera pas l'index de la valeur mais la **clé** qui lui est associée.

L'instruction pour accéder à une valeur est construite à partir du nom du dictionnaire et de la clé de la valeur souhaitée entre crochets.

nom_dictionnaire[clé]

Exemple

Le dictionnaire s'appelle `informaticiens` et la valeur `1815` est associée à la clé `"Ada Lovelace"`.

On accède donc à cet élément avec l'instruction :

`informaticiens["Ada Lovelace"]`

```
1 informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
2 print(informaticiens["Ada Lovelace"])
3 1815
```

- Existence d'une clé _____



L'appel d'une clé non existante dans un dictionnaire génère une erreur.

```
1 informaticiens = {"Alan":1912, "Ada":1815, "Grace":1906}
2 print(informaticiens["Bill"])
```

génère l'erreur :

```
1 Traceback (most recent call last):
2 File "tmp.py", line 2, in <module>
3   print(informaticiens["Bill"])
4 KeyError: 'Bill'
```

Pour vérifier au préalable l'existence de la clé, on utilisera l'instruction `in` qui renverra un booléen.

clé `in` dictionnaire

Exemple

```
1 informaticiens = {"Alan":1912, "Ada":1815, "Grace":1906}
2 "Ada" in informaticiens
3 # True
4 "Bill" in informaticiens
5 # False
```

- Ajouter / modifier un couple clé / valeur _____

Pour ajouter un élément, l'instruction est composée du nom du dictionnaire, de la clé entre crochets et de la valeur. L'affectation se fait par le signe `=`.

On a donc :

`nom_dictionnaire[clé] = valeur`

Exemple

Pour ajouter le couple `Ian 1973`, au dictionnaire `informaticiens`, on utilisera l'instruction :

```
1 informaticiens["Ian Murdock"] = 1973
2 print(informaticiens)
3 {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906, "Ian Murdock":1973}
```

i

Il y a 2 cas possibles :

- Si la **clé** est déjà présente dans le dictionnaire, la **valeur** associée est **remplacée** par la nouvelle,
- Si la **clé** n'est pas déjà présente dans le dictionnaire, le couple clé / valeur est ajouté au dictionnaire.