# Dictionnaire (type dict)

Partie 2 : Données structurées

Quiz \_\_\_\_\_



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.

### Implémentation \_\_\_\_\_

•Initialisation d'un dictionnaire

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 ,

Dictionnaire vide \_\_\_\_\_\_

Il est possible de créer un dictionnaire vide :

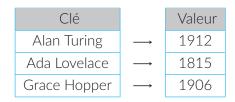
```
# Créer un dictionnaire vide
dico = {}
```

#### • • Dictionnaire non vide \_\_

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

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

On peut représenter les clés et valeurs associées dans ce dictionnaire par ce tableau :



#### •Accéder, ajouter et modifier des éléments \_\_\_\_\_

• • Accéder à un élément \_\_\_

A 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"]

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

• • 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 :

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



- 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.
  - Parcourir les dictionnaires
    - • Parcourir les clés \_\_\_\_\_
- i II existe deux méthodes pour parcourir les clés d'un dictionnaire :
  - En iterant sur le dictionnaire
  - En itérant sur l'objet dict\_keys obtenu avec la méthode keys ()

#### Exemple

Itération sur le dictionnaire :

```
informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
for cle in informaticiens:
    print(cle)
```

#### Exemple

Itération sur l'objet dict\_keys obtenu par l'utilisation de la méthode keys():

```
informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
for cle in informaticiens.keys():
    print(cle)
```

Dans les 2 cas l'affichage sera:

Alan Turing Ada Lovelace Grace Hopper

Parcourir les valeurs

Dans ce cas aussi, il existe deux méthodes pour parcourir les valeurs d'un dictionnaire :

- En utilisant les clés du dictionnaire,
- En itérant sur l'objet dict\_values obtenu avec la méthode values()

#### Exemple

Itération sur le dictionnaire :

```
informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
for cle in informaticiens:
    print(informaticiens[cle])
```

Itération sur l'objet dict\_values :

```
informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
for valeur in informaticiens.values():
    print(valeur)
```

Dans les 2 cas l'affichage sera:

1912

1815

1906

- Encore une fois, il existe deux méthodes pour parcourir les couples clé / valeur d'un dictionnaire :
  - En utilisant les clés du dictionnaire
  - En itérant sur l'objet dict items obtenu avec la méthode items()

#### Exemple

Itération sur le dictionnaire :

```
informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
for cle in informaticiens:
    print(f"{cle} est né.e en {informaticiens[cle]}")
```

Itération sur l'objet dict\_items :

```
informaticiens = {"Alan Turing":1912, "Ada Lovelace":1815, "Grace Hopper":1906}
for cle, valeur in informaticiens.items():
    print(f"{cle} est né.e en {valeur}")
```

Dans les 2 cas l'affichage sera:

```
Alan Turing est né.e en 1912
Ada Lovelace est né.e en 1815
Grace Hopper est né.e en 1906
```

#### •Supprimer un couple clé / valeur \_\_\_\_\_

Pour supprimer un couple clé / valeur d'un dictionnaire, on utilise l'instruction del suivi du nom du dictionnaire et de la clé du couple à supprimer entre crochets.

del nom dictionnaire clé

#### Exemple

Dans cet exemple, nous allons supprimer le couple cle / valeur : "Alan Turing":1912 du dictionnaire informaticiens.

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

L'affichage obtenu sera :

```
{"Ada Lovelace":1815, "Grace Hopper":1906}
```