

# Les dictionnaires en Python

27 janvier 2025

Un dictionnaire est une structure de type `dict` qui rassemble, comme les listes, un certain nombre d'éléments. Ces éléments sont identifiables par une clé, et à chaque clé est associée une valeur.

Une *entrée* d'un dictionnaire est une couple (*clé : valeur*). Un dictionnaire rassemble un certain nombre d'entrées. Les clés sont en général des chaînes de caractères et les valeurs sont de type quelconque, selon l'usage que l'on en fait.

*Exemple de dictionnaire qui associe à chaque mois le nombre de jours qu'il contient :*

```
dico = {"janvier" : 31, "février": 28, "mars" : 31, "avril" : 30}
```

les clés sont les mois et le nombre de jours sont les valeurs. Les valeurs, comme les clés peuvent être de type quelconque.

## I. Création, accès et modification des éléments d'un dictionnaire

### 1. Création d'un dictionnaire

On peut initialiser un dictionnaire comme ci-dessus :

```
dico = {"janvier" : 31, "février": 28, "mars" : 31, "avril" : 30}
```

On peut aussi un créer un vide puis le remplir de la manière suivante :

---

```
dico = { }    ou    dico = dict()
dico['janvier'] = 31
dico['fevrier'] = 28
dico['mars'] = 31
```

---

### 2. Accès aux éléments et modification d'un dictionnaire

- l'instruction `val = dico['janvier']` affecte à `val` la valeur 31 associée à la clé `janvier`.
- Pour ajouter un élément à un dictionnaire il suffit d'ajouter une nouvelle clé et de l'associer à une valeur en écrivant par exemple `dico['mai'] = 31`.
- En écrivant `dico['mai'] = 12` on écrase la valeur précédente associée à la clé préexistante.
- On peut supprimer un couple (*clé : valeur*) en écrivant par exemple `del dico['mai']`
- Pour savoir si un élément fait partie d'un dictionnaire, on peut écrire `'Mars' in dico`, qui vaut `True`. De plus `'juin' in dico` renvoie `False`.
- pour afficher les éléments du dictionnaire il suffit d'écrire `print(dico)` ou simplement `dico` dans la console.

## II. Parcours d'un dictionnaire

On peut parcourir un dictionnaire à l'aide d'une boucle `for` de la manière suivante :

La boucle suivante parcourt les clés des éléments d'un dictionnaire et affiche les valeurs associées :

---

```
for mois in dico:                ou bien   for mois in dico.keys():
    print(dico['mois'], end='')        print(dico['mois'], end='')
```

---

Le code ci-dessus affiche 31 28 31 30

La boucle suivante parcourt les valeurs des éléments d'un dictionnaire et affiche celles-ci :

---

```
for v in dico.values():
    print(v, end='')
```

---

Le code ci-dessus affiche 31 28 31 30, comme le précédent.

La méthode `items()` appliquée à un dictionnaire renvoie un tuple (clé, valeur) et peut être utilisée au sein d'une boucle :

---

```
for c,v in dico.items():
    print(c,v)
```

---

Le code ci-dessus affiche les tuples :

---

```
'janvier', 31
'fevrier', 28
'mars', 31
'avril', 30
```

---

*remarque* : Il n'existe pas de méthode permettant de retrouver une clé à partir d'une valeur donnée. En effet, plusieurs clés différentes peuvent être associées à une même valeur.

Toutefois, on peut toujours créer un code qui renvoie la liste des clés correspondant à une valeur donnée.

### Exercice 0

1. Créer le dictionnaire des mois jusqu'en décembre avec le nombre de jours associés.
2. Afficher tous les mois qui ont 30 jours.