Numérique et science informatique

Lycée Hoche

année scolaire 2025-2026

Contents

1 Introduction

Un troisième type de types construits est présenté ici , le type **dict** pour les **dictionnaires**. La principale différence avec les listes est qu'un dictionnaire n'est pas ordonné. Un élément n'est pas repéré par un indice entier mais par une "clé".

2 Dictionnaires

2.1 Définition

Dictionnaire

Les éléments d'une liste sont repérés par des indices 0, 1, 2, ...

Dans un dictionnaire, objet de type **dict**, les indices sont remplacés par des objets du type *str*, *float*, *tuple* .
On les appelle des **clés** et à chaque clé correspond une **valeur**.

Ces clés ne sont pas ordonnées.

2.2 Création

Les éléments d'un dictionnaire sont des couples clé-valeur. Un dictionnaire est créé avec des accolades, les différents couples étant séparés par des virgules. La clé et la valeur correspondante d'un élément sont séparées par deux-points.

Exemple :

```
dico = {"A": 0,"B": 1,"C": 2,"D": 3}
```

• On peut construire un dictionnaire en compréhension comme avec les listes

```
>>>{x: 3*x for x in (2, 4, 6)}
{2: 6, 4: 12, 6: 18}
```

ou

```
>>>d = {chr(65+i): i for i in range(26)}
>>> d{'A':0, 'B': 1,'C': 2,'D': 3,'E': 4,'F': 5,'G': 6,'H': 7,'I': 8,'J': 9,
'K':10, 'L':11, 'M': 12,'N': 13,'O': 14,'P': 15,'Q': 16,'R': 17,'S': 18,
'T':19, 'U':20,'V': 21,'W': 22,'X': 23,'Y': 24,'Z': 25}
```

• On peut convertir une liste de listes à deux éléments en dictionnaire avec la **fonction dict**.

```
>>> liste = [['A', 0], ['B', 1], ['C', 2]]
>>> d =dict(liste)
>>> d
{'A':0, 'B': 1,'C': 2}}
```

2.3 Utilisation

Accès aux éléments

• Pour accéder aux clés ou aux valeurs, nous avons les méthodes **keys** et **values**

```
>>>d = {'A': 0,'B': 1,'C': 2}
>>>d.keys()
dict_keys(['A','B','C'])
>>>d.values()
dict_values([0,1, 2])
```

• Pour l'accès à l'ensemble des couples clés-valeurs, nous utilisons la méthode **items**

```
>>>d.items()
dict_items([('A',0), ('B', 1), ('C', 2)])
```

Remarque: les couples clés-valeurs obtenus sont du type tuple.

• Nous pouvons tester l'appartenance à un dictionnaire avec le mot clé **in**

```
>>> "A"in d                 # teste si "A" est une clé
True

>>>3 in d.values()          # teste si 3 est une valeur
False

>>>('C', 2)in d.items()     # teste si ('C', 2) est un couple clé-valeur
True
```

• On en déduit trois manières de parcourir un dictionnaire.

Il est possible d'utiliser les clés, les valeurs, ou les couples clés-valeurs.
Voici un exemple avec les clés

```
>>>for key in d:
...print(key)

A
B
C
```

Pour afficher les valeurs du dictionnaire:

```
>>>for valeur in d.values():
    print (valeur)

0
1
2
```

• Lorsque vous faites une boucle sur un dictionnaire, la clé et la valeur associée peuvent être récupérées en même temps en utilisant la méthode items() :

```
>>>animaux = {'chat': 'félin', 'serpent': 'reptile}

for k, v in animaux.items():
    print(k, v)


chat felin
serpent reptile
```

• L'accès à une valeur particulière s'obtient en précisant la clé. Par exemple,d["A"] a la valeur 0.

Modification dans un dictionnaire

• Nous pouvons modifier une valeur par affectation comme d["A"] = 1. Et l'instruction d["D"] = 3 ne provoque pas d'erreur: une nouvelle clé est créée.

```
>>>d = {'A': 0,'B': 1,'C': 2}
>>>d['A']= 1
>>>d
d = {'A': 1,'B': 1,'C': 2}
```

Attention : une instruction comme v = d["E"] provoque une erreur car la clé "E" n"existe pas. La méthode **get** permet de gérer ce genre de problème.

```
>>> v = d.get("A")
>>> v
0
>>> v = d.get("E")
>>> print(v)
None
```

Nombre d'éléments

la fonction ***len*** renvoie le nombre d'éléments d'un dictionnaire, sa longueur.

Suppression d'un élément

• Pour supprimer un élément, nous utilisons l'instruction del d[clé].

```
>>> d = {'A': 0,'B': 1,'C': 2,'D': 3}
>>> del d["D"]
>>> d
{'A':0, 'B': 1,'C':2}
```

copie

Les comportements sont similaires à ceux rencontrés avec les listes en particulier si les valeurs sont des listes. Il est donc conseillé d'utiliser la fonction **deepcopy** du module copy pour être certain d'obtenir une "vraie" copie.Les éléments d'un dictionnaire peuvent aussi être des dictionnaires. Voici un exemple

```
vols= {'Lisbonne': {'heure': 21:10,'num':'EJU7674','compagnie':'EASYJET'},
'Vienne':{'heure': 21:25,'num':'OS430','compagnie':'AUSTRIAN AIRLINES'},
'Londres':{'heure': 21:55,'num','BA357''compagnie':'BRITISH AIRWAYS'}...}
```

L'implémentation d'un dictionnaire optimise le coût en temps de la recherche d'un élément

3 Ressources

https://docs.python.org/fr/3/tutorial/datastructures.html#dictionaries