LANGAGE PYTHON : DICTIONNAIRES

Un dictionnaire est un objet de **type dict**. Il permet de stocker des données sous la forme **(clé : valeur)**. Une clé est unique et peut être du type entier, flottant, chaîne de caractères ou n-uplet.

Création d'un dictionnaire

```
>>> d1 = {"Math": 14, "Anglais": 12.5, "Français": 13}
{'Math': 14, 'Anglais': 12.5, 'Français': 13}
```

Création d'un dictionnaire : Fonction *dict*

La fonction *dict*, permet de créer un dictionnaire à partir d'une liste ou d'un **couple de valeurs de la forme** *cle = valeur*.

```
>>> liste = [["Math", 14], ["Anglais", 12.5], ["Français", 13], ["Physique", 11]]
>>> d1 = dict(liste)
{'Math': 14, 'Anglais': 12.5, 'Français': 13, 'Physique': 11}
>>> d2 = dict(Histoire = 8, EPS = 18, NSI = 20)
{'Histoire': 8, 'EPS': 18, 'NSI': 20}
```

Création d'un dictionnaire par compréhension

```
>>> d4 = {'a' + str(x) : x**2 for x in range(1, 5)}
{'a1': 1, 'a2': 4, 'a3': 9, 'a4': 16}
```

Accès à l'ensemble des clés d'un dictionnaires : méthode *keys*

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> d.keys()
dict_keys(['Math', 'Anglais', 'Français', 'Physique'])
```

Accès à l'ensemble des valeurs d'un dictionnaires : méthode *values*

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> d.values()
dict_values([14, 12.5, 13, 11])
```

Accès à l'ensemble des couples d'un dictionnaires : méthode *items*

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> d.items()
dict_items([('Math', 14), ('Anglais', 12.5), ('Français', 13), ('Physique', 11)])
```

Accès à une valeur d'un dictionnaire à partir de sa clé

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> d["Anglais"]
12.5
```

Accès à une valeur d'un dictionnaire à partir de sa clé : méthode *get*

La méthode *get* permet de retourner la valeur d'un élément à partir d'une clé existante et retourne *None* si la clé n'existe pas.

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> d.get("Math")
14
>>> d.get("EPS")
None
```

Modification d'un élément d'un dictionnaire

Les dictionnaires sont, comme les listes, des objets **mutables.**

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> d["Anglais"] = 15
>>> d
{'Math': 14, 'Anglais': 15, 'Français': 13, 'Physique': 11}
```

Taille d'un dictionnaire : fonction *len*

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> len(d)
4
```

Liste des clés ou des couples communs à deux dictionnaires : opérateur &

```
>>> d1 = {"Paul": 18, "Léa": 18, "Kévin": 19, "Cloé": 17}
>>> d2 = {"Nathan": 17, "Clara": 16, "Kévin": 19, "Jeanne": 17}
>>> d1.keys() & d2.keys()
{'Kévin'}
>>> d1.items() & d2.items()
{('Kévin', 19)}
```

Liste des clés ou des couples présents dans un dictionnaire et pas dans un autre : opérateur -

```
>>> d1 = {"Paul": 18, "Léa": 18, "Kévin": 19, "Cloé": 17}
>>> d2 = {"Nathan": 17, "Clara": 16, "Kévin": 19, "Jeanne": 17}
>>> d1.keys() - d2.keys()
{'Paul', 'Léa', 'Cloé'}
>>> d1.items()- d2.items()
{('Paul', 18), ('Léa', 18), ('Cloé', 17)}
```

Suppression d'un élément dans un dictionnaire : fonction *del*

```
>>> d = {"Paul": 18, "Léa": 18, "Kévin": 19, "Cloé": 17}
>>> del(d["Paul"])
{'Léa': 18, 'Kévin': 19, 'Cloé': 17}
```

Suppression de tous les éléments d'un dictionnaire : méthode *clear*

```
>>> d = {"Paul": 18, "Léa": 18, "Kévin": 19, "Cloé": 17}
>>> d.clear()
{}
```

Test d'appartenance d'un élément à un dictionnaire : opérateur *in*

```
>>> d = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> "Math" in d
True
>>> "EPS" in d
False
```

Copie d'un dictionnaire : méthode *copy*

```
>>> Paul = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
>>> Pierre = Paul.copy()
```

Copie d'un dictionnaire contenant des tableaux : fonction *deepcopy* (copie profonde)

```
>>> from copy import deepcopy
>>> d1 = {'a': [1, 1], 'b': [1,2], 'c': [1,3]}
>>> d2 = deepcopy(d1)
```

Parcours d'un dictionnaire

Un dictionnaire est un objet **itérable**, c'est-à-dire qu'il possible de le **parcourir** à l'aide d'une boucle **for**.

```
notes = {"Math": 14, "Anglais": 12.5, "Français": 13, "Physique": 11}
for cle, valeur in notes.items() :
    print("Matière : ", cle, " : ", valeur)

Matière :  Math  :  14
Matière :  Anglais  :  12.5
Matière :  Français  :  13
Matière :  Physique  :  11
```