



▼ Introduction

Les dictionnaires sont des structures de données permettant de stocker des valeurs reliées sous forme de clé/valeur "Key/Value" et la possibilité de chercher une valeur à travers la clé.

▼ Dictionnaire simple

```
countries_population = {  
    'china': 1_439_323_776,  
    'india': 1_380_004_385,  
    'usa': 331_002_651,  
    'indonesia': 273_523_615  
}
```

```
print(countries_population['usa'])
```

```
331002651
```

▼ Utilisation des dictionnaires

Un dictionnaire en Python est une collection de clé-valeur "key-value". Chaque clé est connectée à la valeur. Une clé peut être utilisée pour accéder à la valeur correspondante. La valeur peut être un nombre, une chaîne de caractères "string", une liste ou un autre dictionnaire. En Python, un dictionnaire est entouré d'accolades {} avec une série de paires de clé-valeur.

Accéder aux valeurs

```
countries_population['china']
```

```
countries_population['china']
```

```
1439323776
```

Ajouter des paires clé-valeur

```
countries_population['pakistan'] = 220_892_340
countries_population['brazil'] = 212_559_417
```

```
countries_population
```

```
{}
```

Dictionnaire vide

Parfois, il est obligatoire de commencer avec un dictionnaire vide.

```
temperatures = {}
```

```
temperatures['france'] = 33
temperatures['sweden'] = 20
temperatures['norway'] = 21
```

```
temperatures
```

```
{'france': 33, 'norway': 21, 'sweden': 20}
```

Double-cliquez (ou appuyez sur Entrée) pour modifier

```
del countries_population['china']
```

```
countries_population
```

```
{'india': 1380004385, 'indonesia': 273523615, 'usa': 331002651}
```

Pour vider le dictionnaire:

```
countries_population.clear()
```

```
countries_population
```

```
{}
```

La fonction **copy** permet de faire une copie profonde du dictionnaire.

```
copy_countries_population = countries_population.copy()
```

```
copy_countries_population
```

```
{'china': 1439323776,
 'india': 1380004385,
 'indonesia': 273523615,
 'usa': 331002651}
```

La fonction **get** permet d'obtenir la valeur pour la clé spécifiée. Le deuxième argument permet de spécifier une valeur alternative/par défaut quand la clé n'existe pas dans le dictionnaire.

```
countries_population.get('china', 0)
```

```
1439323776
```

Et si la clé n'existe pas dans le dictionnaire, la valeur spécifiée pour le 2ème argument est renvoyée.

```
countries_population.get('chiina', 'non existant')
```

```
'non existant'
```

La fonction **items** permet de renvoyer les éléments du dictionnaire sous forme d'une liste de tuples, elle est utilisée pour parcourir les éléments du dictionnaire.

```
countries_population.items()
```

```
dict_items([('china', 1439323776), ('india', 1380004385), ('usa', 331002651), ('indor
```

Opération unpack, c'est l'affectation d'un tuple de valeur à un tuple de variables.

```
a, _, c = 2, 3, 1
```

```
print(a, c)
```

```
2 1
```

```
for k, _ in countries_population.items():
    print(k, )
```

```
china 1439323776
india 1380004385
```

```
usa 331002651
. . .
```

Si on utilise directement la boucle en utilisant le dictionnaire, une erreur est affichée.

```
for (k, v) in countries_population:
    print(k, v)
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-29-407709ba7b03> in <module>()
----> 1 for (k, v) in countries_population:
      2     print(k, v)

ValueError: too many values to unpack (expected 2)
```

SEARCH STACK OVERFLOW

La fonction **keys** permet de retourner la liste des clés d'un dictionnaire.

```
countries_population.keys()

dict_keys(['china', 'india', 'usa', 'indonesia'])
```

La fonction **values** permet de renvoyer la liste des valeurs d'un dictionnaire.

```
countries_population.values()

dict_values([1439323776, 1380004385, 331002651, 273523615])
```

Pour avoir par exemple la valeur moyenne des populations:

```
sum(countries_population.values())/len(countries_population.values())

855963606.75
```

La fonction **pop** permet de supprimer et de renvoyer la valeur supprimée d'un élément du dictionnaire.

```
countries_population

{'china': 1439323776, 'indonesia': 273523615, 'usa': 331002651}

population = countries_population.pop('china')

population

1439323776
```

La fonction **update** permet de mettre à jour la valeur pour la clé précisée en utilisant {} pour spécifier l'élément en question.

```
countries_population.update({'usa': 0})
```

```
countries_population
```

```
{'india': 1380004385, 'usa': 0}
```

Parcourir un dictionnaire en utilisant une liste de compréhension

```
[(k, v) for (k, v) in countries_population.items()]
```

```
[('china', 1439323776),  
 ('india', 1380004385),  
 ('usa', 331002651),  
 ('indonesia', 273523615)]
```

★

▼ Imbrication de listes et de dictionnaires

On peut utiliser une liste de dictionnaires.

```
countries = [  
    {'name': 'france', 'population': 43948948, 'gdp': 93938388},  
    {'name': 'belgique', 'population': 33399409, 'gdp': 9338838}  
]
```

Comme on peut utiliser un dictionnaire de listes.

```
employees = {'dupont': ['engie', 'bnp paribas', 'socgen'], 'alex': ['orange']}
```

