

Les dictionnaires

Plan

1. Création d'un dictionnaire
2. Accéder à une valeur du dictionnaire
3. Modifier une valeur du dictionnaire
4. Ajouter une nouvelle valeur à un dictionnaire
5. Supprimer un élément du dictionnaire
6. Parcourir les éléments d'un dictionnaire
7. Parcourir les valeurs d'un dictionnaire
8. Parcourir les clefs d'un dictionnaire
9. Vérifier l'existence d'une clé dans un dictionnaire

Définition

- Un **dictionnaire** en Python va permettre de rassembler plusieurs éléments mais ceux-ci seront identifiés par une **clé et pas un indice**.
- On peut faire l'analogie avec un dictionnaire de français où on accède à une définition avec un mot.
- Contrairement aux listes qui sont délimitées par des crochets, on utilise des **accolades** pour les dictionnaires.

Création d'un dictionnaire Python

- Pour créer un nouveau dictionnaire, nous allons devoir utiliser un couple d'accolades `{ }` et définir les paires **clef : valeur** à l'intérieur des accolades comme ceci :
- Syntaxe:

```
NomDictionnaire= {clé1: valeur 1, clé2: valeur2 , clé3 : valeur3 ,.....}
```

- Exemple :

```
monDictio = {"Nom" : "girafe", "Taille": 5.0 , "Poids" : 1100}
```

Le type des éléments dans un dictionnaire

- La **clé** est toujours une chaîne de caractère
- La **valeur** peut être un des types qu'on utilise : entier, réel, chaîne, liste, ou même un dictionnaire ...
- *Exemple:*

```
D = { "nom": "mohamed", "age": 20 , "notes": [12,15,11] }
```

Accéder à une valeur du dictionnaire

- Comme on accède à une **valeur d'une liste** grâce à son **indice**. On accède à une **valeur** du **dictionnaire** grâce à sa **clé** correspondante.
- Exemple:

```
print ( monDictio["Nom"] )  
print ( monDictio["Taille"] )  
print ( monDictio["Poids"] )
```

- Résultat:

```
girafe"  
5.0  
1100
```

Modifier une valeur du dictionnaire

- Pour modifier une valeur du dictionnaire on le fait grâce à **sa clé**.
- Exemple:

```
monDictio["Nom"]="Lion"  
print(monDictio)
```

- Résultat:

```
{'Nom': 'Lion', 'Poids': 1100, 'Taille': 5.0}
```

Ajouter une valeur au dictionnaire

- Pour ajouter une valeur:

```
monDictio["Genre"]="herbivore"  
print(monDictio)
```

- Résultat:

```
{'Genre': 'herbivore', 'Nom': 'Lion', 'Poids': 1100, 'Taille': 5.0}
```


Remplir un dictionnaire par l'utilisateur

- Pour cela on doit créer un dictionnaire vide puis on va affecter des valeurs en précisant leur clés.
- Exemple:

```
nombre_de_pneus = {}  
nombre_de_pneus["voiture"] = 4  
nombre_de_pneus["vélo"] = 2
```

Supprimer un élément du dictionnaire

- **pop(clé):** permet de supprimer un élément par sa clé
- **del NomDictio[clé]:** supprimer un élément par sa clé
- Exemples:

```
d={"nom":"alami" , "prenom":"mohamed"}  
#d.pop("nom")  
del d["nom"]  
print(d)
```

```
#résultat:  
{"prenom": "mohamed"}
```

Vérifier l'existence d'une clé ou dans le dictionnaire

- On peut vérifier l'existence d'une clé ou d'une valeur par le mot clé **in**:
- Exemple:

```
d={"nom":"alami" , "prenom":"mohamed"}  
  
if "nom" in d:  
    print("oui")  
else:  
    print("non")
```

La taille d'un dictionnaire

- On peut récupérer la taille d'un dictionnaire (c'est-à-dire le **nombre de couple** qu'il contient) à l'aide de la fonction **len()** déjà vue pour les listes.
- Exemple:

```
D={"nom": "mohamed", "age": 20, "notes":[12,15,11] }  
L=len(D)  
print("le nombre des éléments du dictionnaire :",L)
```

Parcourir les clés d'un dictionnaire

- La méthode `.keys()` permet de récupérer l'ensemble des clés dans un dictionnaire.
- Exemple:

```
Animal = {'nom':'singe', 'poids':70, 'taille':1.75}
```

```
for c in Animal.keys():  
    print(c)
```

#Résultat:

```
nom  
poids  
taille
```

Parcourir les valeurs d'un dictionnaire

- La méthode **.values()** permet de récupérer l'ensemble des clés dans un dictionnaire.
- Exemple:

```
Animal = {'nom':'singe', 'poids':70, 'taille':1.75}
```

```
for v in Animal.values():  
    print(v)
```

#résultat:

singe

70

1.75

parcourir un dictionnaire

- On peut utiliser la méthode `items()` qui permet de récupérer un élément du dictionnaire soit une **paire clé: valeur**.
- Exemple:

```
Animal = {'nom':'singe', 'poids':70, 'taille':1.75}
```

```
for el in Animal.items():  
    print(el)
```

#résultat:

('nom', 'singe')

('poids', 70)

('taille', 1.75)

parcourir un dictionnaire

- On peut utiliser la méthode `items()` qui permet de récupérer un élément du dictionnaire soit une **paire clé: valeur**.

- Exemple:

```
nombre_de_roues = {"voiture": 4, "vélo": 2, "tricycle": 3}

for cl, val in nombre_de_roues.items():
    print(cl, val)
```

- Résultat:

```
voiture 4
vélo 2
tricycle 3
```


Parcourir un dictionnaire

- Autre façon :

```
Animal = {'nom':'singe', 'poids':70, 'taille':1.75}

for cl in Animal:
    print(cl, " :",Animal[cl])
```

- Cl ici est considéré comme une clé (key)
- Résultat:

```
>>>
nom : singe
poids : 70
taille : 1.75
```

Liste de dictionnaires

- Exemple:

```
lst = [{'a':0,'b':1,'c':2},  
       {'d':3,'c':4,'b':5},  
       {'a':5,'d':6,'e':1}]  
  
print(lst[1])  
print(lst[1]['c'])
```

Résultat:

```
{'d': 3, 'c': 4, 'b': 5}  
4
```