



Les dictionnaires

Le dictionnaire est un type de données qui se rapproche des listes, mais, contrairement aux listes, qui n'a pas une structure ordonné.

Un dictionnaire est composé de clés auxquelles sont associées une valeur. Aussi bien les clés que les valeurs peuvent être de tous les types .

I Création d'un dictionnaire

Le dictionnaire se crée à l'aide d'accolade (quand les listes sont créées avec des crochets et les tuples avec des parenthèses).

```
>>> dico = {}  
>>> type(dico)  
<class 'dict'>
```

Pour créer une clé, on la nomme entre guillemets, le tout entre crochet. On lui affecte sa valeur à l'aide du signe égal.

```
>>> dico = {}  
>>> dico ["Nom "] = "Chan"  
>>> dico ["Prénom "] = "Jacky"  
>>> dico  
{'Nom ': 'Chan', 'Prénom ': 'Jacky'}
```

On peut aussi le créer directement :

```
>>> dicol = { "Nom " : "Chan", "Prénom" : "Jacky"}  
>>> dicol  
{'Nom ': 'Chan', 'Prénom': 'Jacky'}
```

La syntaxe n'est pas tout à fait la même, la clé n'étant plus entre crochet et l'affectation se faisant avec les deux points.

II Opérations sur les dictionnaires

a) Parcourir un dictionnaire.

Pour parcourir la liste des clés , on utilise la méthode keys

```
>>> for cle in dicol.keys():  
    print(cle)
```

```
Nom  
Prénom  
'
```

A noter que l'instruction for cle in dico1() fonctionne aussi.

Pour parcourir la liste des valeurs, on utilise la méthode values

```
>>> for valeurs in dicol.values():  
    print(valeurs)
```

```
Chan  
Jacky  
'
```

Pour avoir les clés et les valeurs, on utilise la méthode items

```
>>> for cle, valeur in dicol.items():  
    print(cle,valeur)
```

```
Nom Chan  
Prénom Jacky  
'
```

Remarque , on peut vouloir utiliser une mise en page plus appropriée :

La méthode format nous le permet :

```
>>> for cle, valeur in dicol.items():  
    print('le {} est {}'.format(cle, valeur))  
  
le Nom est Chan  
le Prénom est Jacky  
'
```

Les premières accolades sont remplacées par les clés et les secondes par les valeurs.

Pour avoir la valeur d'une clé , on utilise la méthode get

```
>>> dicol.get("Nom ")  
'Kennedy'  
'
```

Attention , on peut utiliser la méthode get avec deux paramètres, le nom de la clé et 0 au cas où la clé n'aurait pas de valeur.

b) Supprimer, ajouter un élément.

Pour ajouter, il suffit de créer une nouvelle clé.

```
>>> dicol["Age"]=25
>>> dicol
{'Nom ': 'Chan', 'Prénom': 'Jacky', 'Age': 25}
```

Pour modifier la valeur d'une clé, on procède comme pour une variable :

```
>>> dicol["Nom "]='Kennedy'
>>> dicol
{'Nom ': 'Kennedy', 'Prénom': 'Jacky', 'Age': 25}
```

Pour supprimer une clé, on utilise la méthode del (ou la méthode pop si on veut la conserver la valeur associée à la clé supprimée en mémoire)

```
>>> del(dicol["Age"])
>>> dicol
{'Nom ': 'Kennedy', 'Prénom': 'Jacky'}
```

Enfin, pour copier un dictionnaire de façon indépendante, on utilise la méthode copy, sinon la modification d'un dictionnaire modifiera l'autre. Cette remarque s'applique aussi aux listes.

Exercice

A partir d'une phrase entrée par l'utilisateur, créer un dictionnaire qui ait les lettres en clés et leur occurrence en valeurs.

Transformer ce dictionnaire en liste (ou tuple) et afficher le résultat par ordre alphabétique.

Prolongation : Afficher ce résultat sous forme d'histogramme avec le module matplotlib.