Les dictionnaires

Notions sur les dictionnaires

- Un dictionnaire Python est un ensemble ordonné de paires clé-valeur. Les éléments du dictionnaire sont des paires clé-valeur séparées par des virgules.
- Les clés du dictionnaire Python sont immuables et doivent être uniques, tandis que les valeurs sont mutables et peuvent être modifiées ou modifiées à tout moment. C'est exactement comme dans un dictionnaire linguistique : le mot (clé) est immuable, mais sa définition (valeur) est modifiable.
- Les dictionnaires constituent un type d'objet Python qui ressemble aux listes (ils sont modifiables comme elles), mais ce ne sont pas des séquences; ce sont des paires d'information, et, à ce titre, doivent être considérés comme des objets différents.
- Les éléments que l'on enregistre ne sont pas disposés dans un ordre particulier. En revanche, on peut accéder à n'importe lequel d'entre eux à l'aide d'un index spécifique qui est la clé, souvent alphabétique ou numérique.
- Comme dans une liste, les éléments mémorisés dans un dictionnaire peuvent être de n'importe quel type (valeurs numériques, chaînes de caractères, listes,... et même dictionnaires).
- Les dictionnaires se révèlent très pratiques lorsqu'on doit manipuler des structures complexes à décrire et que les listes présentent leurs limites.
- Les dictionnaires sont des collections non ordonnées d'objets, c'est-à-dire qu'il n'y a pas de notion d'ordre (i.e. pas d'indice, alors qu'il y a un indice dans les listes). On accède aux valeurs d'un dictionnaire par des clés.

Création d'un dictionnaire

Création en combinant des listes

Entrée [1]:
```
Liste_cle=["clé1","clé2","clé3","clé4","clé5"]
Liste_valeur=["valeur1","valeur2","valeur3","valeur4","valeur5"]
dico={} # ou dico=dict()
for i in range (len(Liste_cle)):
    dico[Liste_cle[i]]=Liste_valeur[i]
print(dico)
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4', 'clé5': 'valeur5'}
```

Création directe

Entrée [2]:
```
dico2={"clé6":"valeur6","clé7":"valeur7"}
print(dico2)
```

```
{'clé6': 'valeur6', 'clé7': 'valeur7'}
```

Chercher/manipuler des données dans un dictionnaire

Chercher la valeur d'une clé : dico.get(str)

Entrée [3]:
```
print(dico)
a=dico.get("clé1") ; print(a)
b=dico.get("clé6") ; print(b)
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4', 'clé5': 'valeur5'}
valeur1
None
```

Vérifier la présence d'une clé dans un dictionnaire python

Entrée [4]:
```
c="clé2" in dico ; print(c)
val="valeur2" in dico ; print(val) # On ne peut pas vérifier directement la présence d'une valeur
```

```
True
False
```

Supprimer une entrée dans un dictionnaire python

Entrée [5]:
```
print(dico)
del dico["clé5"] ; print(dico)
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4', 'clé5': 'valeur5'}
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4'}
```

Récupérer les clés d'un dictionnaire python : dico.keys() [2 façons]

Entrée [6]:
```
L_cles=[]
for k in dico.keys() : # Par une boucle
    print(k,type(k))
    L_cles.append(k)
print(L_cles)
```

```
clé1 <class 'str'>
clé2 <class 'str'>
clé3 <class 'str'>
clé4 <class 'str'>
['clé1', 'clé2', 'clé3', 'clé4']
```

Entrée [7]:
```
print(dico.keys())       # écriture directe dans le shell
print(type(dico.keys()))
```

```
dict_keys(['clé1', 'clé2', 'clé3', 'clé4'])
<class 'dict_keys'>
```

Entrée [8]:
```
# Un dictionnaire n'est pas une liste !
print(dico)
print(dico.keys())
print(dico.values())
print(dico[0])
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4'}
dict_keys(['clé1', 'clé2', 'clé3', 'clé4'])
dict_values(['valeur1', 'valeur2', 'valeur3', 'valeur4'])

---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/var/folders/28/btygf73s0x9_xl6rnmgv3vsc0000gn/T/ipykernel_97473/644786465.py in <module>
      3 print(dico.keys())
      4 print(dico.values())
----> 5 print(dico[0])

KeyError: 0
```

Récupérer les valeurs d'un dictionnaire python : dico.values() [2 façons]

Entrée [9]:
```
L_valeurs=[]
for k in dico.values() : # Par une boucle
    print(k,type(k))
    L_valeurs.append(k)
print(L_valeurs)
```

```
valeur1 <class 'str'>
valeur2 <class 'str'>
valeur3 <class 'str'>
valeur4 <class 'str'>
['valeur1', 'valeur2', 'valeur3', 'valeur4']
```

Entrée [10]:
```
print(dico.values())       # écriture directe dans le shell
print(type(dico.values()))
```

```
dict_values(['valeur1', 'valeur2', 'valeur3', 'valeur4'])
<class 'dict_values'>
```

Récupérer les clés et valeurs d'un dictionnaire python : dico.items() [2 façons]

Entrée [11]:
```
L_totale=[]
for i,j in dico.items() : # Par une boucle
    print(i,j)
    L_totale.append((i,j)) # on crée une liste de tuple
print(L_totale)
```

```
clé1 valeur1
clé2 valeur2
clé3 valeur3
clé4 valeur4
[('clé1', 'valeur1'), ('clé2', 'valeur2'), ('clé3', 'valeur3'), ('clé4', 'valeur4')]
```

Entrée [12]:
```
print(L_totale[0]);print(type(L_totale[0]))
print(L_totale[0][0]) ; print(type(L_totale[0][0]))
print(L_totale[0][1]) ; print(type(L_totale[0][1]))
```

```
('clé1', 'valeur1')
<class 'tuple'>
clé1
<class 'str'>
valeur1
<class 'str'>
```

Entrée [13]:
```
L_totale_2=[]
for i,j in dico.items() : # Par une boucle
    print(i,j)
    L_totale_2.append(i) # on crée une liste
    L_totale_2.append(j) # de chaînes de caractères
print(L_totale_2)
```

```
clé1 valeur1
clé2 valeur2
clé3 valeur3
clé4 valeur4
['clé1', 'valeur1', 'clé2', 'valeur2', 'clé3', 'valeur3', 'clé4', 'valeur4']
```

Créer une copie indépendante, ou non, d'un dictionnaire python

Copie non indépendante

Entrée [14]:
```
# copie non indépendante
e=dico ; print(e)
dico["clé4"]="valeur4_Modifiée" ; print(e) # on modifie dico, et e est affecté
print(id(dico))
print(id(e))
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4'}
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4_Modifiée'}
4485389120
4485389120
```

Copie indépendante

Entrée [15]:
```
# copie indépendante
dico["clé4"]="valeur4" ; print(dico)
print(id(dico))
e=dico.copy() # e est copiée avec un autre adressage mémoire
print(id(e))
dico["clé4"]="valeur4_Modifiée" ; print(dico) ; print(id(dico)) ;  print(e)
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4'}
4485389120
4502900864
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4_Modifiée'}
4485389120
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4'}
```

Rappel : on a la même logique avec les listes pour les copies indépendantes

Entrée [16]:
```
# copie dépendante
L1=[1,2,3,4]
L2=L1
print(L2 is L1) ; print(L2 == L1) ; print(id(L1))
L1.append(5)
print(L1,L2)
print(L2 is L1) ; print(L2 == L1) ; print(id(L1))
print("******************")
# copie indépendante
L3=L1.copy()
print(L3 is L1) ; print(L3 == L1) ; print(id(L1)) ; print(id(L3))
L1.append(6)
print(L1,L3)
print(L3 is L1) ; print(L3 == L1) ; print(id(L1)) ; print(id(L3))
```

```
True
True
4502948352
[1, 2, 3, 4, 5] [1, 2, 3, 4, 5]
True
True
4502948352
******************
False
True
4502948352
4502948096
[1, 2, 3, 4, 5, 6] [1, 2, 3, 4, 5]
False
False
4502948352
4502948096
```

Fusionner des dictionnaires python

Entrée [17]:
```
dico.update(dico2) ; print(dico)
```

```
{'clé1': 'valeur1', 'clé2': 'valeur2', 'clé3': 'valeur3', 'clé4': 'valeur4_Modifiée', 'clé6': 'valeur6', 'clé7': 'valeur7'}
```

Renvoyer la clé associée à une valeur

Entrée [18]:
```
def trouve_la_cle(dictionnaire,v):
    """ cherche dans le dictionnaire si la valeur v existe. Si oui, renvoie la clé associée"""
    for key, val in dictionnaire.items():
        if v == val:
            return key
    return "la clé associée à "+str(v)+" n'existe pas"
```

Entrée [19]:
```
color_dict ={"blue":1, "green":2, "red":3, "orange":4}
print(trouve_la_cle(color_dict,1))
print(trouve_la_cle(color_dict,2))
print(trouve_la_cle(color_dict,5))
```

```
blue
green
la clé associée à 5 n'existe pas
```