

Algorithmique et Programmation

Dictionnaires

Dictionnaire

Un dictionnaire est une autre structure de données qui permet de **stocker** plusieurs valeurs y compris des tuples, des listes ou même encore d'autres dictionnaires.

L'utilité des dictionnaires, c'est qu'ils nous permettent de stocker des valeurs en utilisant nos propres indices, ainsi il suffit de fournir l'indice pour accéder à l'information stockée avec cet indice.

Création d'un dictionnaire

Un dictionnaire, appelé aussi un « tableau associatif », est une séquence de paires « clé : valeur » séparées par virgules et mises entre accolades.

dict1 = {'C451236' : 'Anabih issa', 'E985477' : 'Amal Abbassi' }
dict2 = {'a': 1, 'b':2, 'c':3, 'd':4}
dict3 = {'name': 'Ouassit Youssef'}
dict4 = { } # dictionnaire vide

Les deux points « : » doit être utiliser comme séparateur entre la clé la valeur correspondante.

Accès aux valeurs d'un dictionnaire

Comme toutes les autres séquences pour accéder aux valeurs d'un dictionnaire, vous devez utiliser les crochets :

D = {"name": "zara", "age": 7, "class": "first"}			
D ["name"]	D ["age"]	D ["class"]	D ["state"]
'zara'	7	'firts'	KeyError

Une Erreur « **KeyError** » est générée si vous avez fourni une clé qui n'existe pas, comme dans « state ».

Dictionnaires sont modifiables « mutables »

Vous pouvez mettre à jour un dictionnaire par ajout de nouvelle valeur, modifier des valeurs existantes ou supprimer des valeurs.

AJOUT DE VALEUR
D = {"name": "zara", "age": 7, "class": "first"}
D["school"] = "High School"
La clé "school" n'existe pas dans le dictionnaire « D » donc elle va être ajoutée avec la valeur « High School »

MODIFICATION DE VALEUR
D = {"name": "zara", "age": 7, "class": "first"}
D["age"] = 8

La valeur de la clé « age » va être modifiée en « 8 »

SUPPRESSION DE VALEUR

D = {"name": "zara", "age": 7, "class": "first"}

del D['age'] # supprimer la paire de clé 'age'

D.clear() # vider le dictionnaire

del D # supprimer le dictionnaire tout entier

Propriétés de clés d'un dictionnaire

Les valeurs d'un dictionnaire n'ont aucune restriction et donc elles peuvent être de n'importe quel type. Ce n'est pas le cas pour les clés.

Il y'a deux importants points à se rappeler pour les clés d'un dictionnaire :

- Une clé ne peut pas référencer plus qu'une valeur.
- La clé doit être « immutable » c.-à-d. ça peut être de n'importe quel type sauf (liste, dictionnaire).

Operations de bases sur les dictionnaires

D = {'C451236' : 'Anabih issa', 'E985477' : 'Amal Abbassi' }

Expression	Résultat	Description
len(D)	2	Nombre de valeurs dans un dictionnaire
'C451236' in D	True	Teste d'appartenance d'une clé
D.has_key('C451236')	True	Teste d'appartenance d'une clé
for cle in D : print(D[cle])	'Anabih issa' 'Amal Abbassi'	Iteration par boucle for

Manipulation des dictionnaires (méthodes)

Python inclut des méthodes de manipulation des dictionnaires :

D = {"name": "zara", "age": 7, "class": "first"}

D.clear()	Vide le dictionnaire 'D'
D.copy()	Retourne une copie du dictionnaire 'D'
D.fromkeys(keys, value)	Crée et retourne un dictionnaire à partir de la liste keys et la valeur 'value'
D.get(key, default)	Return la valeur de la clé key si ça existe dans le dictionnaire si non la fonction retourne la valeur default
D.items()	Retourne la liste de paires (clé, valeur)
D.keys()	Retourne la liste des clés du dictionnaire 'D'
D.update(D2)	Met à jour le dictionnaire 'D' par le dictionnaire 'D2'
D.values()	Retourne la liste de valeurs stockées dans le dictionnaire 'D'