Date 8/3/21

CHAPTER → 7

Sequence Data Type : Dictionary

Introduction – बड़ी मात्रा में डेटा स्टोर करने के लिए इसका प्रयोग होता है। इसमें data 'key : Value' pair के रूप में होता है जहाँ key (immutable) होती है तथा Value पर (mutable) होती है।

key तथा Value को (:) से Separate (अलग) किया जाता है।

dictionary के डेटा को Comma से अलग किया जाता है तथा यह curly bracket के अंदर लिखी जाती है।

NOTE → Key : Value की एक unique id होती है।

{} → Empty dictionary

{1: 'aaa', 2: 'bbb', 3: 'ccc'} → Dictional of names

Creating a Dictionary :- डिक्शनरी बनाने के दो तरीके हैं –

1. By assigning Values
2. Dynamically via Keyboard

1. **By assigning Value** :- Dictionary बनाने के लिए Key : Value pair को { } में लिखा जाता है और इसे एक नाम से assign कर दिया जाता है।

Syntax: <dictionary name> = {<key:<Value>, <key>: <Value>,

Example: players = {1: 'Sachin', 2: 'Dhoni, 3: 'Rohit'}

2. **Dynamically via keyboard :-**

Input Function का प्रयोग करके dynamic dictionary बनायी जाती है।

Syntax: <key variable> = input ("<any message>")
<Value Variable> = input ("<any message>")

Example: k = input ("Enter dict key")
v = input (" " " Value")
dict [k] = v

Program- एक प्रोग्राम लिखे जिसमें 3 शहरों का तापमान इनपुट करके उनका उपयोग कर एक डिक्शनरी बनाएं। शहर का नाम key और तापमान को value के रूप में लें।

```
dict = {}
for a in range (3):
    k = input ("Enter city name")
    v = input ("Enter temp of ")
    dict [k] = v
print (dict)
```

Accessing the Dictionary –

दो प्रकार से dictionary access की जाती है –

1. **Accessing whole dictionary** → पूरी डिक्शनरी को एक साथ एक्सेस करने के लिए इसके नाम का प्रयोग किया जाता है।

Ex:
```
marks = {1:38, 2:41, 3:29}
print (marks)
```

NOTE → * के साथ डिक्शनरी प्रिन्ट करते समय बिना punctuation marks के डिक्शनरी की key प्रिन्ट होती हैं।

Date: Page:

2. Accessing individual element :- डिक्शनरी में डेटा key:Value pair के रूप में स्टोर होता है। का प्रयोग करके उसकी Value को key एक्सेस किया जाता है।

Syntax: <dictionary name> [<key>]

Ex. Marks = {1:38, 2:41, 3:29}
print (marks[2])

Program → पाँच छात्रों के अंको की डिक्शनरी बनाने हेतू एक प्रोग्राम लिखें और डिक्शनरी को प्रिंट करे। किसी एक छात्र के अंक को प्रिन्ट करने के लिए यूजर से उनका रोल नम्बर इनपुट लेकर अंक प्रिन्ट करें।

```
marks = {1:39, 2:46, 3:30, 4:37, 5:42}
print (marks)
While (True):
    print ("Roll numbers is", marks)
    k = input ("Enter Roll no.")
    print ("Marks of roll no.", k, "is", marks[k])
```

Traversing the Dictionary :-

डिक्शनरी के element को access करने के लिए for loop में डिक्शनरी के नाम का प्रयोग किया जाता है। यह process traversing है।

Syntax: for <control variable> in <dictionary name>:
    body of loop

Example: Marks = {1:39, 2:46, 3:30, 4:37, 5:42}
for a in marks:
    print (marks[a])

Dictionary Operations :-

1. **Adding element to dictionary** –

Assignment method का प्रयोग करके डिक्शनरी में element जोड़े जाते हैं।

Syntax : <dictionary name> [<key>] = <Value>

Example : Cricket = {1:'Virat', 2 : 'Dhoni'}
Cricket [3] = 'Sachin'
print (Cricket)

2. **Creating dictionary using key & Value separately :-**

zip Function तथा dict Function का प्रयोग करके key तथा Value की Separately input के द्वारा dictionary बनायी जाती हैं।

Syntax : <dictionary name> = dict (zip ((<list of keys>), (<list of value>)))

Example : Cricket = dict (zip ((1,2,3), ('Sachin', 'Dhoni', 'Virat')))
print (Cricket)

3. **Creating dictionary using key : Value pair as list :-**

इस method में दो elements की एक लिस्ट dict की function के argument के रूप में pass जाती हैं। इस फंक्शन में एक से अधिक argument हो सकते हैं।

Syntax : <dictionary name> = dict([[<key value pair>],
[<key value pair>],.....])

Example: Cricket = dict([[1,"Sachin"],[2,"Rohit"],[3,"Dhoni"]])
print (cricket)

Program: क्रिकेटर के नाम से एक डिक्शनरी बनाने के लिए एक प्रोग्राम लिखे जिसमें क्रिकेटर के नाम के रूप में और इंडेक्स नं० के रूप में का उपयोग करें। एलिमेंट की value संख्या यूजर्स पर निर्भर key होनी चाहिए। एलिमेंट्स को प्रिंट भी करे।

```
c=1
Cricket = {}
ch=1
while (ch==1):
    v= input ("Enter player name")
    cricket[c] = v
    c = c+1
    ch= input ("Do you want to add any more data
                yes=1 and no=0")
print (cricket)
```

4. Updating dictionary element :—

डिक्शनरी में पहले से उपस्थित element अपडेट किये जा सकते हैं। यह नये element जोड़ने के समान है।

Example: dict = {1:'a', 2:'b', 3:'c'}
update - dict[1] = 'd'

new elem. - dict[4] = 'e'

5. **Deleting dictionary element :—**

डिक्शनरी से element हटाने के दो तरीके हैं -

1. del statement
2. pop function

Syntax-1 del < dictionary name > [< key >]

Example del cricket [2]

NOTE - दी गई key ~~अगर~~ यदि dictionary में नही है तब एक error message print होगा।

Syntax-2 < dictionary name >. pop (< key >)

Example Cricket. pop (2)

NOTE - यह फंक्शन element delete करने के बाद उसकी Value को प्रिन्ट भी करता है।

6. **Checking existing key :—**

Membership Operator 'in' तथा 'not in' का प्रयोग करके डिक्शनरी में key की उपस्थिति Check की जा सकती है।

Syntax : <key> in < dictionary name >
<key> not in < dictionary name >

Example: ~~if rno in~~ players = { 1 : "Tendulkar", 2 : "Sachin", 3 : "kapil" }

```
k = 3
if k in players:
    print ("Player is", players [k])
else:
    print ("Player is not exist")
```

Nested dictionary :—

एक डिक्शनरी के अंदर अन्य डिक्शनरी nested dictionary कहलाती है।

Internal dictionary Outer dictionary की Value होती है।

Syntax: <dictionary name> = {<key> : {<key> : <Value>, <key> : <Value>
{<key> : <Value> . . . . }. . }

Example:
```
Student = {1:{'name':"sohan", 'mark':32}, 2:{'name':"Dinesh
'mark':40}, 3:{'name':"mohan", 'mark':42}}
print(student[1]['name'])
```
Output: Sohan

Functions of Dictionary :-

1. लेन (len()) :— इसका प्रयोग डिक्शनरी की key को गिनने के लिए किया जाता है।

Syntax: len(<dictionary name>)

Example:
```
dic = {1:"abc", 2:"def", 3:"ghi"}
print(len(dic))
```
Output: 3

2. क्लीयर (clear()) :— डिक्शनरी से सभी element को हटाने के लिए इस function का प्रयोग किया जाता है।

Syntax: <dictionary name>.clear()

Example: dic.clear()

3. गेट (get()) :- इस फंक्शन का प्रयोग दी गई key की Value प्राप्त करने के लिए किया जाता है। यह Function दो argument input करता है।

1. key 2. Message

Syntax: <dictionary name>.get(<key>,<message>)
Example: V= dic.get(4,"key does not exist")
print(V)

4. हैश कुंजी (has_key()) :- यह फंक्शन दी गई dictionary में key की present चैक करता है। तथा Boolean Value return करता है।

Syntax: <dictionary name>.has_key(<key>)

Example: V = dic.has_key(3)
print(V)

Output True

5. आइटम्स (items()):- यह फंक्शन डिक्शनरी की सभी key:Value pair की सूची Tuple return करता है।

Syntax: <Dictionary name>.items()
Example: dic.items()

Output dict_items([(1,'abc'),(2,'def'),(3,'ghi')])

6. कुंजीयां (keys()) :- यह फंक्शन डिक्शनरी की सभी key को return करता है।

NOTE → key लिस्ट के रूप में return होती है।

Syntax: <dictionary name>.keys()
Example: dic.keys()
dict_keys([1,2,3])

(7.) वैल्यू (Values()):- यह फंक्शन डिक्शनरी की सभी Value को return करता है।
Value लिस्ट के रूप में return होती है।

Syntax: <dictionary name>.values()
Example: dic.value()
dict_values(['abc','def','ghi'])

(8.) अपडेट (Update()):- यह फंक्शन एक डिक्शनरी में अन्य डिक्शनरी को जोड़ देता है।

Syntax: <original Dictionary>.update(<new dictionary>)
Example: Cricket.update(players)

Players dictionary के elements को cricket dictionary में जोड़ दिया जायेगा।