7/11/22

Tuple and Dictionary

*. **Tuple** → It is a data structure which contain sequence of values. A Tuple consists of multiple values in a singe variable separated by commas (,).

i) Tuples are Enclosed within parenthesis ().

ii) Tuple allow duplicate values.

iii) Element in a Tuple are heterogeneous.

iv) Tuples are immutable i.e. we cannot change the element of tuple in a place. This means ~~wh~~ we cannot perform insert, update, and delete operation on them.

v) Tuple having single value, is known as ~~St~~ Singleton Tuple.

vi) If a tuple comprises a single element, the element should be followed by commas (,) to distinguish a tuple.

Examples :- T= (0,1,2,3,4)
T1= ("hello", 1,2,3, 'World')
T2 = ()
T3= (70,)
T4= ((0,1,2),(3,4,5))
↳ Nested Tuple.

Q) Write an output in a given tuple.

```
T = (10, 20, 30, 40, [4, 5, 6])
T[4][1] = 500
print (T)
```

=> (10, 20, 30, 40, [4, 500, 6])

Dictionary

14/11/22 Dictionary is an unordered collection of item. Each ~~value~~ item is a key value pair. It is separated by its value by a colon & items are separated by commas., They are enclosed in curly braces ('{}').

Dictionary are mutable, i.e. we can add new items & change the value of existing items.
The value of a dictionary can be of any type but keys must be an ~~inn~~ immutable data types such as strings, nos. or tuples.

Examples :-

d={} ⇒ Empty Dictionary

d= {'R':'Rainy', 's':'summer', 'w':'winter'}

↑ Key ↑ value

Methods to create a dictionary :-

i) Create an empty dictionary by two curly braces through a variable name.

```
O/P → >>> d= { }
③     >>> type (d)
      <class 'dict'>
```

ii) Creating a dictionary by using string data type for both the key value pair.

```
O/P → d2= {'a':'anuj', 'b':'bajaj', 'c':'casual'}
      print (d2)
```

iii) To create a dictionary by ~~the user~~ built in function {dict}(). It is used to create a dictionary with no items

O|P→

```
D= dict()
print (D)
```

→ { }

iv) Create a copy of dictionary by passing a dictionary as an arguement to dict() function.

```
d = {1:2, 2:3}
D = dict (d)
print (D)
```

O|P→ {1:2, 2:3}

v) To add an item to a dictionary, we can use [] for accessing and initialising dictionary values.

```
D= { }
D[1] = "one"
D [2] = "Two"
D [3] = "Three"
print (D)
```

O|P→ {1:'one', 2:"Two', 3:"Three"}

15/11/22

#. Accessing the Dictionary Elements

To access dictionary element, the square brackets [] along with the key are used to obtain its value If we attempt to access the data item which are not present in the given dictionary, we get an error.

Ex:-
```
d = {'Name': 'anuj', 2: 'Ravi'}
print (d[2])
print (d['anuj'])
print (d[5])
```

O/p ->
```
Ravi
Error
Error.
```

#. Traversing a Dictionary

Traversing means accessing each element in a dictionary by using loop.

'in' operator are used with for loop to access each element

Ex:-
```
d= {'Name':'anuj', 2:'Ravi'}
for i in d:
        print (i)
        print (d[i])
```

Appending values to Dictionary.

We can add new element to the existing dictionary, join two dictionary into one. If we want to add a single element in a dictionary, then we have to use dictionary name along with the key value.

Syntax

```
dict_name [Key]= value.
```

#. Updating element in Dictionary

Updating means modifying existing key : value pair or merging another dictionary with the another one.

Two dictionaries can be merged into one by ~~a~~ using update method.
~~If~~ It merges the key and value of one dictionary with another and overwrites the value of the same key

Syntax

```
        dict_1. update (dict_2)
```

Ex:-

```
d = {'Name' : 'anuj', 2: 'Aman'}
d1 = { 'Tilak': 2 , 'Name' : 'Suman' }

d. update (d1)
print (d)
```

O/P=) {'Name' : 'Suman', 2: 'Aman', 'Tilak': 2}

#. 'in' and 'not in' Membership Operator.

This operator checks whether the given key is present in the dictionary or not. It returns true if the key element is present otherwise it returns false.

Note* Function such as min(), max() & sum(), only applies to the keys in a dictionary.

#. Removing an item from a Dictionary

We can remove an item from existing dictionary using del() or pop() function.

i) del() → It deletes the value by taking key as a input.

Syntax.

```
del. dict_name [Key]
```

If key is not present in the given dict., it will generate an error.

If you want to delete the entire dictionary, then use the del () command along with the dictionary name.

→
```
d= {1:'A', 2:'B'}
del d
```

ii) pop () → This method will not only delete the item but also returns the specific key with the deleted values.

Syntax :- dict_name.pop (Key)

Common Dictionary Functions & Methods.

i) len () → It returns the number of key value pair from the given dictionary

Syntax ⇒ len (d).

ii) keys () → It returns a list of key values in a given dictionary.

Syntax dict_name.keys()

iii) values() → It returns a list of values from key : value pair in a given dictionary.

Syntax dict_name.values()

15/11/20

iv) clear () → It removes all the element from the particular dictionary.

Syntax d. clear ()

v) get() → The get method returns a value for a given key. If key is not available, it returns a value – none.

Syntax

d.get (key, default = none)

```
>>> d = {1:2, 2:3, 3:4}
>>> d.get (2)
>>> 3.

>>> d. get (5)
>>> none

>>> d.get (4, 'not there')
>>> not there.
```

vi) **items()** → It returns the content of a dictionary as a list of tuples, having key-value pair.

Syntax D.items()

```
>>> d= {1:2, 2:3, 3:4}
>>> d. items ()
>>> [(1,2) ,(2,3), (3,4)]
```

vii) copy()

vii) copy() → It cannot copy a dictionary by using '=' operator. We can use the function copy to create a new dictionary and copy the values to its.

Syntax dict_new = old_dict.copy()

```
>>> d = {1:2, 2:3, 3:4}
>>> d1 = {}
>>> d1 = d.copy()
```

viii) popitem() → It removes last item from the dictionary and returns the deleted item.

Syntax D.popitem()

```
>>> d={1:2,2:3,3:4}
>>> s= d.popitem()
>>> print(s)
>>>
```

ix) max() and min() → The method max() returns key having maximum value whereas min() returns the key having minimum value.

x) sorted() → It sorts the element of a dictionary by its key or value.

```
>>> D={'student1':80,'student2':78,
       'student3':76}
    L1= sorted (D)
    print (L1)
O/P → {'student1', 'student2', 'student3'
```

```
>>> L2 = dict (sorted (D.items()))
    print (L2)
```

~~o/p→~~

```
O/P → {'student1':80, 'student2':78, 'student3':
```

```
>>> L3= sorted (D.values())
    print(L3)
```

```
O/P → {76,78,80}
```

17/11/22

Output - Based Questions :-

1)
```
>>> d={1:10 ,2:20, 3:30 ,4:40}
>>> d.items ()
>>> d. values ()
>>> d. keys ()
```

=>
```
[(1,10), (2,20), (3,30) , (4,40)]
[10, 20, 30 ,40]
[ 1, 2, 3, 4]
```

2)
```
>>> d= {1:10, 2:20, 3:30}
>>> d1= {4:40 , 6:60}
>>> d1. update (d)
>>> print (d1)
```

=>
```
 4:40, 6:60
{^,1:10, 2:20 , 3:30}
```

3)
```
d= {1:10 ,2:30 ,3:30}
del d [3]
# pop d [2]
print (d)
```

=>
```
[2,30]
{1: 10 }
```

4)
```
d= {1:10, 2:20, 3:30, 4:40}
len(d)
d.get(6,60)
d[3]=60
print(d)
d.clear()
print(d)
```

⇒
```
 60
{1:10, 2:20, 3:60, 4:40}
 { }
```

Q) Consider the following code and find out the error in it.

a)
```
t=(1,'School, 6, 7)
print (max(t))
```

b)
```
t=(1,'s','h', 5,'p')
t[1]=0
```

c) $T_1 = 7$

$T_2 = (1,2, 3)$

$T_3 = T_1 + T_2$

d)
```
T1 = (1,2,3)
T2 = (4,5,6)
T3 = (8,9)
a,b,c = T1, T2
```

e)
```
T1 = (1,2,3)
T4 = T1 *(2, )
```

f)
```
tup1 =("5") * '3'
print (tup1)
```

Ans a) Max() only works in same data type.

Ans b) Tuples are immutable.

Ans c) '+' operator do not work with integer and tuple. Only works with two tuples.

Ans d) For tuple assignment, left hand side and right hand side should be same. T_3 is not there in R.H.S.

Ans e) '*' operator requires a tuple and an integer value.

Ans f) '*' operator only requires integer value to operate. Here 3 is a string.

Q) WAP to remove the element 3 from the given tuple and ~~displty~~ display it.

~~T = (1,2,3,4,5) ; K=[ ]~~
~~for i in T :~~
~~if~~

```
→ T = (1,2,3,4,5)                    [1,2,3,4,5]
  L= List (T)
  L.pop (2)
  T1 = tuple(L)
  print (T1)
```

Q) Consider the given tuple t1 and answer the following questions.

t1 = (10, 20, 'book', 30, 9.5, "items", [12, 13], (3,4), 30, 5, 30)

11 a) len (t1)

b) t1 [-8:-4]

c) t1. index (20)

8 d) t1. index (30, 7, 10)

e) t1 [:1] * 5

f) t1 [:6]

g) t1 [5 :]

h) t1 . index (30)

i) t1. count (30)

j) any (t1) True

Q) WAP to input a full name and display a name with its initial and title.

Ex!- Ravi Kumar Sahu
R.K. Sahu.

⇒
```
S= input ("Enter your full name = ")
t= S.title()
u= t.split()
T= u.pop(-1)
I = " "
I2= " "
for j in u:
    I = I+j
for k in I:
    if k.isupper():
        I2 = I2 + k
        I2 = I2 + "."
Final = I2+T
print (Final)
```

Ans1a) 11

b) (30, 9.5, "items", [12, 13])

c) 1

d) 8

e) 10, 10, 10, 10, 10

f) 10, 20, 'book', 30, 9.5, "items"

g) "items", [12, 13], (3, 4), 30, 5, 80

h) 3

i) 3

j) True.