DAY-19

SEPTEMBER-30

- To print individual elements of tuple

```
t1 = (23,'Indu',78.9,False,'karan')
for i in t1:
    print(i)
```

o/p:

23

Indu

78.9

False

karan

1. t2 = ('task','pen','paper','bench','table')

output:

task

pos = 0 , val = t

pos = 1 , val = a

pos = 2 , val = s

.....

Code:

```
t2 = ('task','pen','paper','bench','table')
for string in t2:
    print(string)
    for j in enumerate(string):
```

```
    print(f"pos = {j[0]},val = {j[1]}")
```

output:

task

pos = 0,val = t

pos = 1,val = a

pos = 2,val = s

pos = 3,val = k

pen

pos = 0,val = p

pos = 1,val = e

pos = 2,val = n

paper

pos = 0,val = p

pos = 1,val = a

pos = 2,val = p

pos = 3,val = e

pos = 4,val = r

bench

pos = 0,val = b

pos = 1,val = e

pos = 2,val = n

pos = 3,val = c

pos = 4,val = h

table

pos = 0,val = t

pos = 1,val = a

pos = 2,val = b

pos = 3,val = l

pos = 4,val = e

- Tuple is immutable so we cannot perform data manipulation directly but we can change it to list and perform all necessary data manipulation and convert back to tuple.

Example:

temp = list(t1)

temp

o/p: [23, 'Indu', 78.9, False, 'karan']

temp[3]=True

temp

o/p: [23, 'Indu', 78.9, True, 'karan']

t1 = tuple(temp)

t1

o/p:

(23, 'Indu', 78.9, True, 'karan')

t3 = ('priya',[1,2,3,4],78.4,[[[10,20,30]]])

t3

o/p: ('priya', [1, 2, 3, 4], 78.4, [[[10, 20, 30]]])

-To access elements inside the tuple:

t3[2]

78.4

t3[1][1]

2

t3[3][0][0][2]

30

2.tup1 = (10,20)

tup2 = (50,70)

expected o/p:

tup1 = (50,70)

tup2 = (10,20)

Code:

```
tup1 = (10,20)
tup2 = (50,70)
print("before swap\n",tup1,tup2)
tup1,tup2 = tup2,tup1
print("after swap\n",tup1,tup2)
```

output:

```
before swap
 (10, 20) (50, 70)
after swap
 (50, 70) (10, 20)
```

3. Sort a tuple of tuples by 2nd item

tuple1 = (('a',23),('b',37),('c',11),('d',29))

expected output:

(('c',11),('a',23),('d',29),('b',37))

Code:

```
tuple1 = (('a',23),('b',37),('c',11),('d',29))
temp = sorted(list(tuple1),key=lambda x:x[1])
tuple1 = tuple(temp)
tuple1
```

output:

(('c', 11), ('a', 23), ('d', 29), ('b', 37))

Sets:

- A set is an unordered,mutable,collection of unique elements.
- It is defined using set() or {}
- It automatically deletes the duplicated values.
- Heterogenous in nature

```
s1 = {23, 'Indu', 78.9, True, 'karan','Indu', 78.9, True}
s1
```

o/p:

{23, 78.9, 'Indu', True, 'karan'}

- To add elements in the set:
  ```
  s1.add(34)
  s1
  ```
  {23, 34, 78.9, 'Indu', True, 'karan'}
- To delete elements in the set:
  ```
  S1.remove(34)
  ```

S1
o/p:
{23, 78.9, 'Indu', True, 'karan'}

- Pop randomly removes one element from the set.
  s1.pop()
  s1
  o/p:
  {23, 78.9, 'Indu', 'karan'}
- To avoid errors we use discard function
  s1.discard(11)
  s1
  o/p:{23, 78.9, 'Indu', 'karan'}

1. s1 = {4,5,7,22,8}

s2 = {5,7,10,50,6}

- to get common elements :
  s1.intersection(s2)
- to get unique elements:
  s1.union(s2)
- to get unique elements of s1:
  s1.difference(s2)
- to get unique elements of s2:
  s2.difference(s1)
- to get symmetric elements :
  s1.symmetric_difference(s2)

2. Remove all the duplicates from the list

l3 = [1,2,3,3,3,4,4,5,6,7,8,9,9]

code:

```
l3 = [1,2,3,3,3,4,4,5,6,7,8,9,9]

l3 = list(set(l3))

l3
```

o/p: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Dictionary:

- A dictionary is a built-in unordered, mutable, and key-value data structure. It stores data in pairs: each key maps to a value, like a real-world dictionary where a word (key) maps to its meaning (value).
- Key–Value Mapping Data stored as {key: value} pairs.
- Mutable You can add, update, or delete elements after creation.
- Keys must be unique; values can repeat.
- We can define dictionary by using {key:values} or dict()

Example:

```
student = {
   "name": "Indu",
   "age": 22,
   "course": "AI"
}
```

o/p:

```
{'name': 'Indu', 'age': 22, 'course': 'AI'}
```

```
pairs = [("name", "Indu"), ("age", 22)]
student = dict(pairs)
student
```

o/p: {'name': 'Indu', 'age': 22}

Data Manipulation on Dictionaries

A. Add or Update

```
student["grade"] = "A"
student["age"] = 23
```

o/p: {'name': 'Indu', 'age': 23, 'grade': 'A'}

B. Delete

```
del student["course"]
age = student.pop("age")
student.clear()
```

C. Iterate

```
for key in student:
```

```
    print(key, student[key])
```
o/p:
name Indu
age 23
grade A

```
for key, value in student.items():
    print(key, value)
```
o/p:
name Indu
age 23
grade A

D. Merge / Combine
```
extra = {"college": "VIT", "year": 2025}
student.update(extra)
```
o/p:
{'name': 'Indu', 'age': 23, 'grade': 'A', 'college': 'VIT', 'year': 2025}

E. Dictionary Comprehension

Create or transform dictionaries:

```
squares = {x: x**2 for x in range(5)}
squares
```
o/p: {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}