**1. Create a tuple with single item 23**

***Sample Output***

Create a tuple with Item = 23

Type = tuple

t=(23,)

print("Type=",type(t))

**Output:**

Type= <class 'tuple'>

**2. Unpack the tuple into 5 variables**

***Sample Output***

Tuple = (11, 22, 333, 44, 55)

11  
22  
333  
44  
55

t=(11,22,333,44,55)

(a,b,c,d,e)=t

print(a,b,c,d,e,sep='\n')

**Output:**

11

22

333

44

55

**3. Swap two tuples in Python**

***Sample Output***

Before Swap A : 10

Before Swap B : 20

After Swap A : 20

After Swap B : 10

t1=(1,2,3)

t2=('a','b','c')

print("Before swapping t1=",t1,"and t2=",t2)

t1,t2=t2,t1

print("After swapping t1=",t1,"and t2=",t2)

**Output:**

Before swapping t1= (1, 2, 3) and t2= ('a', 'b', 'c')

After swapping t1= ('a', 'b', 'c') and t2= (1, 2, 3)

**4. Copy specific elements from one tuple to a new tuple**

***Sample Output***

(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

(30, 40, 50, 60, 70, 80)

t=(10,20,30,40,50,60,70,80,90,100)

t1=t[2:8]

print(t)

print(t1)

**Output:**

(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

(30, 40, 50, 60, 70, 80)

**5. Modify the tuple**

***Sample Output***

( 10, 20, 30, 40, 50 )

( 10, 20, 33, 40, 50 )

t=(10,20,30,40,50)

print(t)

l=list(t)

l[2]=33

t=tuple(l)

print(t)

**Output:**

(10, 20, 30, 40, 50)

(10, 20, 33, 40, 50)

**6. Sort a tuple of tuples by 2nd item**

***Sample Output***

( ('a', 53), ('b', 37), ('c', 23), ('d', 1), ('e', 18) )

( ('d', 1), ('e', 18), ('c', 23), ('b', 37), ('a', 53) )

t=( ('a', 53), ('b', 37), ('c', 23), ('d', 1), ('e', 18) )

tup=list(t)

print(t)

lst = len(tup)

for i in range(0, lst):

for j in range(0, lst-i-1):

if (tup[j][1] > tup[j + 1][1]):

temp = tup[j]

tup[j] = tup[j + 1]

tup[j + 1] = temp

t=tuple(tup)

print(t)

**Output:**

(('a', 53), ('b', 37), ('c', 23), ('d', 1), ('e', 18))

(('d', 1), ('e', 18), ('c', 23), ('b', 37), ('a', 53))

**7. Counts the number of occurrences of item 30 from a tuple**

***Sample Output***

(30, 50, 10, 30, 70, 50, 30)

Number of Counts 30 : 3

t=(30, 50, 10, 30, 70, 50, 30)

print(t)

print("Number of Counts 30 :",t.count(30))

**Output:**

(30, 50, 10, 30, 70, 50, 30)

Number of Counts 30 : 3

**8. Write a Python program to compute element-wise sum of given tuples**

***Sample Output***

A = (2, 5, 8)

B = (6, 5, 1)

C = (1, 4, 7)

D = (3, 7, 2)

Sum of Elements = (12, 21, 18)

A = (2, 5, 8)

B = (6, 5, 1)

C = (1, 4, 7)

D = (3, 7, 2)

print("A",A)

print("B",B)

print("C",C)

print("D",D)

s=[]

for i in range(3):

sum=A[i]+B[i]+C[i]+D[i]

s.append(sum)

print("Sum of Elements = ",tuple(s))

**Output:**

A = (2, 5, 8)

B = (6, 5, 1)

C = (1, 4, 7)

D = (3, 7, 2)

Sum of Elements = (12, 21, 18)

**9. Write a Python program to sort a tuple by its float element**

***Sample Output***

[ ('Ram', '89.20'), ('Siva', '76.45'), ('Pooja', '84.40'), ('Tara', '68.43'), ('Jeeva', '91.40') ]

[ ('Jeeva', '91.40'), ('Ram', '89.20'), ('Pooja', '84.40'), ('Siva', '76.45'), ('Tara', '68.43') ]

t=[ ('Ram', '89.20'), ('Siva', '76.45'), ('Pooja', '84.40'), ('Tara', '68.43'), ('Jeeva', '91.40') ]

t.sort(key = lambda x: float(x[1]), reverse = True)

print(t)

**Output:**

[ ('Ram', '89.20'), ('Siva', '76.45'), ('Pooja', '84.40'), ('Tara', '68.43'), ('Jeeva', '91.40') ]

[ ('Jeeva', '91.40'), ('Ram', '89.20'), ('Pooja', '84.40'), ('Siva', '76.45'), ('Tara', '68.43') ]

**10. Write a Python program to replace last value of tuples in a list**

***Sample Output***

[(5, 2, 3), (4, 7, 6), (8, 9, 6)]

Replace value = 10

[(5, 2, 10),(4, 7, 10),(8, 9, 10)]

l=[(5, 2, 3), (4, 7, 6), (8, 9, 6)]

print(l)

l[-1]=(8,9,10)

print(l)

**Output:**

[(5, 2, 3), (4, 7, 6), (8, 9, 6)]

[(5, 2, 3), (4, 7, 6), (8, 9, 10)]

**11. Write a Python program to Extract tuples having K digit elements**

***Sample Output***

[ (47, 23), (3, 78), (22, 53), (121, 45), (7,) ]

K = 2

[ (47, 23), (22, 53) ]

l=[ (47, 23), (3, 78), (22, 53), (121, 45), (7,) ]

k=2

l1=[]

for i in l:

if 10<=i[0]<=100 and 10<=i[1]<=100 :

l1.append(i)

print(l1)

**Output:**

[ (47, 23), (22, 53) ]

**12. Write a Python program to Extract Symmetric Tuples**

***Sample Output***

[ (18, 23), (2, 9), (7, 6), (9, 2), (10, 2), (23, 18) ]

{ (2, 9), (18, 23) }

l = [ (18, 23), (2, 9), (7, 6), (9, 2), (10, 2), (23, 18) ]

l1=[]

for i in l:

t=i[::-1]

if t in l and i not in l1:

l1.append(t)

print(l1)

**Output:**

[(23, 18), (9, 2)]

**13. Write a Python program to Sort Tuples by their Maximum element**

***Sample Output***

[ (4, 5, 5, 7), (1, 3, 7, 4), (19, 4, 5, 3), (1, 2) ]

[ (19, 4, 5, 3), (4, 5, 5, 7), (1, 3, 7, 4), (1, 2) ]

t=[ (4, 5, 5, 7), (1, 3, 7, 4), (19, 4, 5, 3), (1, 2) ]

for i in range(0,len(t)):

for j in range(0,len(t)-i):

if(max(t[i]) > max(t[j])):

temp=t[i]

t[i]=t[j]

t[j]=temp

print(t)

**Output:**

[(19, 4, 5, 3), (1, 3, 7, 4), (1, 2), (4, 5, 5, 7)]

**14. Write a Python program to Remove nested records from tuple**

***Sample Output***

(10, 20, (30,), 40, (50, 60), 70)

Removal of Nested Tuple : (10, 20, 40, 70)

t=(10, 20, (30,), 40, (50, 60), 70)

l=[]

tt=()

for i in t:

if type(i) != type(tt):

l.append(i)

t=tuple(l)

print(t)

**Output:**

(10, 20, 40, 70)

**15. Write a Python program to Elements Frequency in Mixed Nested Tuple**

***Sample Output***

(10, 20, (20, 10), 30, (40, 50, 60), 60, 70)

Elements Frequency : {10: 2, 20: 2, 30: 1, 40: 1, 50: 1, 60: 2, 70: 1}

t=(10, 20, (20, 10), 30, (40, 50, 60), 60, 70)

t1=()

l=[]

for i in t:

if type(i)== type(t1):

for j in i:

l.append(j)

else:

l.append(i)

res=[]

for i in l:

if i not in res:

res.append(i)

print(i,":",l.count(i),end=" ")

print(",",end="")

**Output:**

10 : 2 ,20 : 2 ,30 : 1 ,40 : 1 ,50 : 1 ,60 : 2 ,70 : 1 ,

**16. Write a Python program to get unique elements in nested tuple**

***Sample Output***

[ (1, 3, 5), (4, 5, 7), (1, 2, 6), (10, 9), (10,) ]

Unique Element in Nested Tuples : [1, 3, 5, 4, 7, 2, 6, 10, 9]

l=[ (1, 3, 5), (4, 5, 7), (1, 2, 6), (10, 9), (10,) ]

l=list((sum(l,())))

res=[]

for i in l:

if l.count(i) == 1 and i not in res:

res.append(i)

print(res)

**Output:**

[3, 4, 7, 2, 6, 9]

**17. Write a Python program to Concatenate tuples to nested tuples**

***Sample Output***

Tuple 1 : ((18, 23, 2, 9),)

Tuple 2 : ((10, 3, 11),)

Tuples after Concatenating : ((18, 23, 2, 9), (10, 3, 11))

t1=((18, 23, 2, 9),)

t2=((10, 3, 11),)

t3= t1+t2

print(t3)

**Output:**

((18, 23, 2, 9), (10, 3, 11))

**18. Write a Python program to Sort by Frequency of second element in Tuple List**

***Sample Output***

[ (2, 7), (3, 7), (2, 5), (8, 7), (6, 5), (9, 8) ]

Sorted List of tuples : [ (2, 7), (3, 7), (8, 7), (2, 5), (6, 5), (9, 8) ]

l=[ (2, 7), (3, 7), (2, 5), (8, 7), (6, 5), (9, 8) ]

l1=sorted(l,key=lambda a: l.count(a),reverse=True)

print(l1)

**Output:**

[(2, 7), (3, 7), (2, 5), (8, 7), (6, 5), (9, 8)]

**19. Write a Python program to Sort lists in tuple**

***Sample Output***

( [10, 50, 60], [80, 20, 30], [70, 100, 40], (90,) )

Tuple after sorting list : ( [10, 50, 60], [20, 30, 80], [40, 70, 100], [90] )

**20. Write a Python program to Order Tuples using external List**

***Sample Output***

[ ('B', 68), ('D', 70), ('A', 67), ('C', 69) ]

Ordered Tuple List : [ ('A', 67), ('B', 68), ('C', 69), ('D', 70) ]

**21. Write a Python program to Filter Tuples by Kth element from List**

***Sample Output***

[ ('B', 68), ('D', 70), ('A', 67), ('C', 69) ]

List of kth elements to filter = [67 , 70 , 71, 75]

Kth element index = 1

Filtered tuples : [('D', 70), ('A', 67)]

**22. Write a Python program to Closest Pair to Kth index element in Tuple**

***Sample Output***

[ (23, 18), (9, 2), (2, 3), (9, 18), (23, 2) ]

Given Tuple = (20, 2)

Kth Element Index = 1

Nearest Tuple : (23, 18)

**23. Write a Python program to Tuple List intersection (Order irrespective)**

***Sample Output***

List 1 : [ (3, 4), (5, 6), (9, 10), (4, 5) ]

List 2 : [ (5, 4), (3, 4), (6, 5), (9, 11) ]

Intersection : { (4, 5), (5, 6), (3, 4) }

**24. Write a Python program to Intersection in Tuple Records Data**

***Sample Output***

list 1 : [ ('A', 65), ('D', 68), ('B', 66) ]

list 2 : [ ('D', 68), ('C', 67), ('A', 65) ]

Intersection of data records : [ ('A', 65), ('D', 68) ]

**25. Write a Python program to Unique Tuple Frequency (Order Irrespective)**

***Sample Output***

[ (3, 4), (1, 2), (4, 3), (5, 6) ]

Unique tuples Frequency : [(1, 2), (3, 4), (5, 6)]

Unique tuples Frequency Count : 3

**26. Write a Python program to Skew Nested Tuple Summation**

***Sample Output***

(1, (2, (3, (4, (5, None)))))

Summation of 1st positions : 15

**27. Write a Python program to Convert Binary tuple to Integer**

***Sample Output***

(1, 0, 1, 0)

Decimal number is : 10

**28. Write a Python program to Tuple XOR operation**

***Sample Output***

Tuple 1 : (10, 4, 6, 9)

Tuple 2 : (5, 2, 3, 3)

XOR tuple : (15, 6, 5, 10)

**29. Write a Python program to AND operation between Tuples**

***Sample Output***

Tuple 1 : (10, 4, 6, 9)

Tuple 2 : (5, 2, 3, 3)

AND operation Between Tuple : (0, 0, 2, 1)

**30. Write a Python program to Elementwise AND in tuples**

***Sample Output***

Tuple 1 : (10, 4, 6, 9)

Tuple 2 : (5, 2, 3, 3)

Elementwise AND Tuple : (0, 0, 2, 1)