Karthik Nair | 3EA | 00229802021

Tuples

1. Write a program to remove duplicates from a tuple.

```
def remove_duplicates(t):
    return tuple(set(t))

def remove_duplicates_2(t):
    new=()
    for i in t:
        if i not in new:
            new+=(i,)
    return new


print(remove_duplicates(eval(input("Enter a tuple: "))))
print(remove_duplicates_2(eval(input("Enter a tuple: "))))
```

2. Write a Python program to check if a specified element presents in a tuple of tuples.

Original list:

(('Red', 'White', 'Blue'), ('Green', 'Pink', 'Purple'), ('Orange', 'Yellow', 'Lime'))

Check if White presenet in said tuple of tuples!

True

Check if White presenet in said tuple of tuples!

True

Check if Olive presenet in said tuple of tuples!

False

```
def check_for_specific_element(t, element):
    for i in t:
        if element in i:
            return True

tuple0=(('Red', 'White', 'Blue'), ('Green', 'Pink', 'Purple'), ('Orange',
'Yellow', 'Lime'))
```

```
elements=("Olive", "White")

for i in elements:
    print(f"{i} ",end="")
    if not check_for_specific_element(tuple0, i):
        print("not ", end="")
    print("present")
```

3. Write a Python program to convert a given tuple of positive integers into an integer.

Original tuple:

(10, 20, 40, 5, 70)

Convert the said tuple of positive integers into an integer:

102040570

```
def get_number_from_tuple(t):
    try:
        return int("".join(map(str,t)))
    except (TypeError, ValueError):
        return 0

print(get_number_from_tuple((10, 20, 40, 5, 70)))
```

4. Write a Python program to convert a tuple of string values to a tuple of integer values.

Original tuple values:

(('333', '33'), ('1416', '55'))

New tuple values:

((333, 33), (1416, 55))

```
def convert_string_values_to_int(t):
    i_t=()
    for i in t:
        i_t+=tuple(map(int,i)),
    return i_t

print(convert_string_values_to_int((('333', '33'), ('1416', '55'))))
```

5. Write a program to swap two tuples.

```
a=eval(input("Enter a tuple: "))
b=eval(input("Enter another tuple: "))

a,b=b,a

print(a,"\n", b)
```

6. Write a program to reverse a tuple.

```
a=eval(input("Enter a tuple: "))

a=a[-1::-1]

print(a)
```

7. Write a program to copy specific elements from one tuple to a new tuple.

```
def copy_els(indexes, t):
    new=()
    for i in indexes:
        new+=(t[i],)
    return new

a=eval(input("Enter a tuple: "))
i=eval(input("Enter indexes you want to copy: "))

print(copy_els(i, a))
```

8. Write a program to sort a tuple of tuples by 2nd item.

Sample:

(('r',3),('t',1),('e',2),('y',9))

Expected Output: (('t',1),('e',2),('r',3),('y',9))

```
def sort_by_second_item(t):
    s=sorted([i[1] for i in t])
    s_t=()
    for i in s:
        for j in t:
            if j[1]==i:
                s_t+=(j),
    return s_t

print(sort_by_second_item((('r',3),('t',1),('e',2),('y',9))))
```

9. Write a program to check if all the items in the tuple are same.'''

t=(22,22,22,22)

if min(t)==max(t):

print("All items are same!")

else:

print("False")

10. Write a Python program to convert a given list of tuples to a list of lists.

Original list of tuples: [(1, 2), (2, 3), (3, 4)]

```
t=[(1, 2), (2, 3), (3, 4)]

t=[list(i) for i in t]
print(t)
```

11. Write a Python program to compute the sum of all the elements of each tuple stored inside a list of tuples.

Original list of tuples:

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

Sum of all the elements of each tuple stored inside the said list of tuples:

[9, -1, 7, 8]

```
t=[(1, 2, 6), (2, 3, -6), (3, 4), (2, 2, 2, 2)]
t=[sum(i) for i in t]
print(t)
```

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

Original lists:

(1, 2, 3, 4)

(3, 5, 2, 1)

(2, 2, 3, 1)

Element-wise sum of the said tuples:

(6, 9, 8, 6)

```
t1=(1, 2, 3, 4)
t2=(3, 5, 2, 1)
t3=(2, 2, 3, 1)

t4=tuple(map(sum,zip(t1,t2,t3)))
print(t4)

# zip(t1,t2,t3) returns a zip object, which is an iterator of tuples where
# the first item in each passed iterator is paired together,
# and then the second item in each passed iterator are paired together etc.
# ie tuple(zip(t1,t2,t3))=((1, 3, 2), (2, 5, 2), (3, 2, 3), (4, 1, 1))

# map(sum,zip(t1,t2,t3)) returns an iterator of the sums of the tuples
# ie sum function runs on every tuple in the zip object
# => tuple(map(sum,zip(t1,t2,t3)))=(6, 9, 8, 6)
```