

29/10/2025

## maketrans & translate

```
intab = "aeiou"
outtab = "12345"
my = "this is a string function!"
transtab = my.maketrans(intab,outtab)
print(transtab)
print(my.translate(transtab))

{97: 49, 101: 50, 105: 51, 111: 52, 117: 53}
th3s 3s 1 str3ng f5nct34n!
```

## REMOVING PUNCTUATION

```
import string

string.punctuation

'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

mystring = "!hi wha?t is t[h]e we?ather li$ke"
newstring =
mystring.translate(str.maketrans("", "", string.punctuation))
print(newstring)

hi what is the weather like

mystring = "!hi wha?t is t[h]e we?ather li$ke"
newstring =
mystring.translate(str.maketrans("a", "Z", string.punctuation))
print(newstring)

hi whZt is the weZther like

a = newstring.maketrans("a", "Z", string.punctuation)
print(a)

{97: 90, 33: None, 34: None, 35: None, 36: None, 37: None, 38: None,
39: None, 40: None, 41: None, 42: None, 43: None, 44: None, 45: None,
46: None, 47: None, 58: None, 59: None, 60: None, 61: None, 62: None,
63: None, 64: None, 91: None, 92: None, 93: None, 94: None, 95: None,
96: None, 123: None, 124: None, 125: None, 126: None}
```

# tuple

- an immutable , ordered sequence which supports indexing,slicing & iterating

```
t = ()
type(t)

tuple

t = (1,2)
type(t)

tuple

a = (1,2,3)
b = (4,5,6)
tuple(zip(a,b))

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

a = (1,2,3)
b = (4,5,6,7)
tuple(zip(a,b))

((1, 4), (2, 5), (3, 6))
```

## tuple packing

```
a = 1,2,3,4
a

(1, 2, 3, 4)
```

## tuple unpacking

```
a,b,c,d = a
a,b,c,d
b
print(a,b,c,d)

1 2 3 4
```

## type conversion

```
a = [1,2,3,4]
print(a)
```

```

b = tuple(a)
print(b)

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

a = tuple("python")
print(a)

('p', 'y', 't', 'h', 'o', 'n')

a,b,*c = 1,2,3,4,5 # A WILL take 1 , b will take 2 & *c will take
3,4,5 as a list

print(a,b,c)

1 2 [3, 4, 5]

```

## indexing & slicing

```

t = (4,5,6,7,8,9)
t[-1]

9

t[-1:-5:-1]

(9, 8, 7, 6)

```

## + and -

```

t1 = (1,2,3)
t2 = (4,5,6)
t1+t2

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

t1*3

(1, 2, 3, 1, 2, 3, 1, 2, 3)

t1*t2 #can't multiply sequence by non-int of type 'tuple'

-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-28-1b4b44balda1> in <module>
----> 1 t1*t2

```

```
TypeError: can't multiply sequence by non-int of type 'tuple'
```

## membership operator

```
t =(7,8,9)
7 in t
```

```
True
```

```
6 in t
```

```
False
```

```
6 not in t
```

```
True
```

## iteration

```
for i in t:
    print(i)
```

```
7
8
9
```

```
for i, j in enumerate(t):
    print(i,j)
```

```
0 7
1 8
2 9
```

```
for i in enumerate(t):
    print(i)
```

```
(0, 7)
(1, 8)
(2, 9)
```

## universal method:

```
t = (4,6,2,7,9,3)
print(len(t))
print(min(t))
```

```

print(max(t))
print(sorted(t))
print(sorted(t,reverse = True))

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

print(sum(t))

31

```

## tuple method

### 1)count

```

t = (1,2,3,3,3,5,6,7)
t.count(3)

```

3

```

t = (1,2,3,3,3,5,6,7)
t.count(3,3)

```

```

-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-38-993fa3f5faf4> in <module>
      1 t = (1,2,3,3,3,5,6,7)
----> 2 t.count(3,3)

```

TypeError: count() takes exactly one argument (2 given)

```

t = (1,2,3,3,3,5,6,7)
t.count(100)

```

0

### 2)index

```

t = (4,5,6,7,6)
t.index(6,3,1000)

```

## comparison

```
(1,2,3)==(1,2,3,4)
```

```
False
```

```
(1,2,3,4) == (1,2,3,4)
```

```
True
```

```
(1,2,3,4)<(1,2,3,4)
```

```
False
```

```
(1,2,3,4)>(1,2,3,4)
```

```
False
```

```
(1,2,3,4)>(1,2,3,4,5)
```

```
False
```

```
(1,2,3,4)>(1,1,5,4)
```

```
True
```

```
(1,2,"a") > (1,2,"A")
```

```
True
```

## Generator

```
t = (1,2,3,4,5)
new = tuple(x*2 for x in t) #multiplication
print(new)
```

```
(2, 4, 6, 8, 10)
```

```
t = (1,2,3,4,5)
new = tuple(x**2 for x in t)#square
print(new)
```

```
(1, 4, 9, 16, 25)
```

# zip tuples

```
t1 = (1,2,3)
t2 = (4,5,6)
new = tuple(x+y for x,y in zip(t1,t2))
print(new)
```

```
(5, 7, 9)
```

```
t = (4,5,1,3,8,7,9)
# reversed(t)
```

```
tuple(reversed(t))
```

```
(9, 7, 8, 3, 1, 5, 4)
```

```
list(reversed(t))
```

```
[9, 7, 8, 3, 1, 5, 4]
```

```
t = (1,2,3)
```

```
del t[0] #'tuple' object doesn't support item deletion
```

```
-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-77-46b1c19aff9f> in <module>
      1 t = (1,2,3)
----> 2 del t[0]
```

```
TypeError: 'tuple' object doesn't support item deletion
```

1) WAP to create a fun named length to find length of the string without using len method

```
def length():
    count = 0
    for i in s:
        count+=1
    print("length: ",count)
s = input("enter string: ")
length()
```

```
enter string: dhruvi
length: 6
```

2) WAP to swap the cases of the string without using swapcase method

```
#### input: -mR, eD #### o/p: Mr,Ed
```

```

old = input("enter any string :")
new=""
for i in old:
    if i.islower():
        new = new + i.upper()
    elif i.isupper:
        new = new + i.lower()
    else:
        new+=1
print(new)

```

```

enter any string :DhruVi
dHRUVi

```

wap to extract the username from the email address .....

- i/p:abc@gmail.com
- o/p:abc

```

email = input("enter an email : ")
email.split("@")[0]

```

```

enter an email : anc@gmail.com

```

```

'anc'

```

```

email = input("enter an email : ")
pos = email.index("@")
email[:pos]

```

```

enter an email : asd@gmail.com

```

```

'asd'

```

WAP to count the frequency of the character in a given string

- i/p: www.google.com -search for - w
- o/p:3

```

s = input("enter a string : ")
search = input("what do you want to search : ")
count=0
for i in s:
    if i == search:
        count+=1
print(count)

```

```

enter a string : www.google.com
what do you want to search : o
3

```