

find()

```
In [3]: # find()
# return index of first occurrence of the given subString
# if it is not available then we will get (-1)
s='Learning Python is very easy'
print(s.find('a'))
print(s.find('s'))
print(s.find('x'))
print(s.find('Python'))
print(s.find('python'))
```

2
17
-1
9
-1

```
In [9]: # s.find(substring,begit,end)
print(s.find('a',7,26))
```

25

count()

```
In [22]: s='abcddddababcdxyzxyza '
print(s.count('a'))
print(s.count('d'))
print(s.count('abcd'))
print(s.count('p'))
print(s.count(' '))
```

3
6
2
0
2

```
In [23]: #s.count(substring,begit,end)
print(s.count('a',8,15))
```

1

replace()

```
In [96]: #to replace old string with new string
# s.replace(old string,new string, number)
s='Learning java is easy'
x=s.replace('a',' bbbb ',1)
print(x)
```

Le bbbb rning java is easy

split()

```
In [51]: #split(separator)-> we can split the given string according to specified separator by
#default separator is space.
# the return type of split() method is list
#split()
s='Hello           worldd'

l=s.split()
print(l)
```

```

l=s.split('l') # if character between letter then it give number of letter -1 empty string
print(l)

l=s.split('d')
print(l)

l=s.split('H')
print(l)

['Hello', 'world']
['He', '', 'o          wor', 'dd']
['Hello           worl', '', '']
[ '', 'ello           world']

```

In [50]:

```

s='29-10-2025'
l=s.split('-')
print(l)

```

['29', '10', '2025']

translate() with maketrans() function

In [53]:

```

import string
print(string.punctuation)
print(len(string.punctuation))

! "#$%&'()*+,-./:;<=>?@[\\]^_`{|}~
32

```

maketrans()

In []:

```

# make translate table
# mapping of character to their replacement or to name for deletion
# maketrans(from_chars,tochars,delete_chars)

```

translate()

In []:

```

# Applies to translation table created by maketrans()
# return new String= with character replaced or deleted according to table

```

In [67]:

```

import string
s='py$@th!!on'
l=s.maketrans("", "", string.punctuation)
print(l)
# ! "#$%&'()*+,-./:;<=>?@[\\]^_`{|}~ = 32
print()

x=s.translate(l)
print(x)

l=s.maketrans("", "", "@$")
print(l)
x=s.translate(l)
print(x)

print()

l=s.maketrans("n", "m", "$")
print(l)
x=s.translate(l)
print(x)

```

```
print()

{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}

python
{64: None, 36: None}
pyth!on

{110: 109, 36: None}
py@th!om
```

```
In [69]: t='Hello Sem'
x='mSa'
y='eJo'
table=t.maketrans(x,y)
print(table)
print(t.translate(table))
```

```
{109: 101, 83: 74, 97: 111}
Hello Gee
```

```
In [70]: t='Hello Sem'
x='mSa'
y='eJoa'# x and y both length must be same
table=t.maketrans(x,y)
```

```
-----  
ValueError                                     Traceback (most recent call last)
<ipython-input-70-ea21acef3b23> in <module>
      2 x='mSa'
      3 y='eJoa'
----> 4 table=t.maketrans(x,y)
```

ValueError: the first two maketrans arguments must have equal length

```
In [72]: #Q wpp replace each special symbol with # in following string
s='/*John is @developer & musician!!'

t=s.maketrans(string.punctuation,32*'#')
print(s.translate(t))

##John is #developer # musician##
##John is #developer # musician##
```

```
In [90]: s='/*John is @developer & musician!!'
for i in s:
    if not i.isalnum() and i!=' ':
        s=s.replace(i,'#')
print(s)
#-----
s='/*John is @developer & musician!!'

l=string.punctuation
for i in s:
    if i in l:
        s=s.replace(i,'#')
print(s)
#-----
s='/*John is @developer & musician!!'
for i in string.punctuation:
    s=s.replace(i,'#')
print(s)
```

```
##John is #developer # musician##
##John is #developer # musician##
##John is #developer # musician##
```

```
In [99]: # wpp to remove i'th char from string
i=int(input("Enter number:"))
s='Hello world'
print(s.replace(s[i], '', 1))
# or
print(s[:i]+s[i+1:])
```

Enter number:10
Hello worl
Hello worl

```
In [105...]: # wpp find count of all occurrences of substring in a give string by ignoring case
s='Welcome to USA. usa is awesome. Usa is good. Usain bolt is American'.lower()
print(s.count('usa'))
```

4

```
In [116...]: # WPP to display all position of substring in a given string
s='aaaabcdabcacdb'
sub='a'
a=0
c=0
for i in s:
    if i==sub:
        c=c+1
        print(a)
    a=a+1
else:
    print('count=',c)
```

0
1
2
3
7
10
13
count= 7

```
In [131...]: #WPP to merge char of two string into single string by taking char alternetively
x='abcaaa'
y='123'
c=' '
for i in range(min(len(x),len(y))):
    c=c+x[i]+y[i]
else:
    if len(x)==min(len(x),len(y)):
        c=c+y[i+1:]
    else:
        c=c+x[i+1:]

print(c)
```

a1b2c3aaa

```
In [4]: #index of substring in main String
s='abcdefghijklmnopqrstuvwxyzabcxyzabc'
sub='abc'
pos=-1
flag=False
n=len(s)
while True:
```

```

pos=s.find(sub,pos+1,n)
if pos== -1:
    break
print(pos)
flag=True
if not flag:
    print('not found')

```

0
3
12
18

In [16]:

```

s='a9b3c2d0'
a=''
for i in range(0,len(s),2):
    a=a+s[i]*int(s[i+1])
print(a)

```

aaaaaaaaabbbcc

In [48]:

```

s='a20bi10cz5'
n=''
l=''
a=''
i=0
for i in s:
    if i.isalpha():
        if len(n)!=0:
            a=a+l*int(n)
            l = ''
            n = ''
        l=l+i

    else:
        n=n+i
if len(n)!=0:
    a=a+l*int(n)
print(a)

```

aaaaaaaaaaaaaaaaaaaaabibibibibibibibiczczczc

In [50]:

```

#WPP to velidity of password
#min 8 char
#alpha
#atleast one upper
#atleast one digit
#atleast special char from [_,@,$]
u=0
d=0
s=0
f=True
p=input('Enter password:')
if len(p)<8:
    print('Entre velid pass')
else:
    for i in p:
        if i.isupper():
            u=u+1
        elif i.isdigit():
            d=d+1
        elif i not in '_@$':
            f=False

```

```

        break
    elif i==' ':
        print('Enter valid pass')
        f=False
    if f and u!=0 and d!=0 and s!=0:
        print('valid')
    else:
        print('Not valid')

```

Enter password:Tejas@1234#
Not valid

In [49]:

```
#WPP to sift decimal digit n placies to the left wraping if sift if greter then numb
n='12345'
s=int(input('Enter number:'))
if s>len(n):
    print(n[::-1])
else:
    print(n[s:]+n[:s])
```

Enter number:3
45123

Tuple

In [60]:

```
t=('aaa')
print(t)#3
print(len(t))
print(type(t))
t=('aaa',)#1
print(t)
print(len(t))
print(type(t))
```

aaa
3
<class 'str'>
(‘aaa’,)
1
<class 'tuple'>

In [52]:

```
t=()
print(type(t))
```

<class 'tuple'>

Accessing element of tuple

- 1.by using index
- 2.by usingslicing opration

In [66]:

```
t=(10,20,30,40,50)
print(t[2])
print(t[2:4])
print(t[2:100])
print(t[2:])
print(t[:2])
```

30
(30, 40)
(30, 40, 50)
(30, 40, 50)
(10, 30, 50)

Methetical opereter for tuple

```
In [68]: # +
t1=(10,20,30)
t2=(30,40,50)
t=t1+t2
print(t)
# *
t=(10,20,30)
t1=t*3
print(t1)
```

```
(10, 20, 30, 30, 40, 50)
(10, 20, 30, 10, 20, 30, 10, 20, 30)
```

Functions of tuple

```
In [1]: #len()
t=(10,20,30)
print(len(t))
```

```
3
```

```
In [2]: #count()
t=(1,2,2,3,3,3,1,1,2,4,5)
print(t.count(1))
print(t.count(2))
print(t.count(6))
```

```
3
3
0
```

```
In [4]: #index()-returns the index of first occurrence of given element if specified element
#           Value Error
t=(10,20,10,10,20)
print(t.index(10))
print(t.index(20))
print(t.index(30))
```

```
0
1
```

ValueError Traceback (most recent call last)
<ipython-input-4-340f67b4714a> in <module>
 4 print(t.index(10))
 5 print(t.index(20))
----> 6 print(t.index(30))

ValueError: tuple.index(x): x not in tuple

```
In [5]: s='hello world' #index() work for string but find() cannot work for tuple
print(s.index('w'))
```

```
6
```

```
In [6]: t=(10,20,10,10,20)
print(t.index(10,1,5))
```

```
2
```

```
In [8]: #sorted()
t=(10,30,40,20,50)
t1=sorted(t)
print(t1)*****output is List*****
print(tuple(t1))
```

```
[10, 20, 30, 40, 50]
(10, 20, 30, 40, 50)
```

```
In [17]: s='LJET'
s1=sorted(s)
print(s1)*****output is List*****
print(''.join(s1))

['E', 'I', 'J', 'L', 'T']
EIJLT
```

```
In [19]: t=(10,30,20,10,40)
t1=sorted(t,reverse=True)
print(t1)

[40, 30, 20, 10, 10]
```

```
In [20]: # min() & max()
t=(10,50,40,20,30)
print(min(t))
print(max(t))
```

```
10
50
```

```
In [21]: #Tuple packing and unpacking
a=10
b=20
c=30
d=40
t=a,b,c,d
print(t)

(10, 20, 30, 40)
```

```
In [2]: t=(10,20,30,40)
a,b,c,d=t
print(a,b,c,d)
```

```
10 20 30 40
```

```
In [4]: a,a,a,a=t
print(a,a,a)
_,_,_,_=t
print(_,_,_)
```

```
40 40 40
40 40 40
```

```
In [8]: #reversed()- it return object
s='python'
print(list(reversed(s)))
t=(20,30,10,40,50)
print(list(reversed(t)))
r=range(5,9)
print(list(reversed(r)))
l=[1,2,3,4,5]
print(list(reversed(l)))
print(reversed(l))
```

```
['n', 'o', 'h', 't', 'y', 'p']
[50, 40, 10, 30, 20]
[8, 7, 6, 5]
[5, 4, 3, 2, 1]
<list_reverseiterator object at 0x000001E7CDE2AB20>
```

```
In [11]: #enumerate()
#If you pass a string to enumerate(), the output will show you the index and value for
#enumerate(itrable,start=0)
s1='LJIET'
obj1=enumerate(s1)
print(obj1)
print(list(obj1))#List of tuples
obj1=enumerate(s1,2)
print(list(obj1))

<enumerate object at 0x000001E7CD7516C0>
[(0, 'L'), (1, 'J'), (2, 'I'), (3, 'E'), (4, 'T')]
[(2, 'L'), (3, 'J'), (4, 'I'), (5, 'E'), (6, 'T')]
```

```
In [19]: for i in enumerate(s1):
    print(i) # tuple
print()
for i,j in enumerate(s1):
    print(i,j,sep=' - ')
print()
for i,j in enumerate(s1,5):
    print(i,j)
```

```
(0, 'L')
(1, 'J')
(2, 'I')
(3, 'E')
(4, 'T')

0 - L
1 - J
2 - I
3 - E
4 - T

5 L
6 J
7 I
8 E
9 T
```

```
In [22]: l=['eat','sleep','repeat']
for i in enumerate(l):
    print(i)
for i,j in enumerate(l):
    print(i,j)
```

```
(0, 'eat')
(1, 'sleep')
(2, 'repeat')
0 eat
1 sleep
2 repeat
```

```
In [44]: # even ,odd,even_sum,odd_sum,even,odd-max,min
t=(1,2,4,6,7,8,13,14,16)
even=()
esum=0
odd=()
osum=0
for i in t:
    if i%2==0:
        even+=i,
        esum+=i
    else:
        odd+=i,
```

```
osum+=i
print('even',even)
print('odd',odd)
print('e_sum',esum)
print('o_sum',osum)
print('e max,e min',max(even),min(even))
print('o max,o min',max(odd),min(odd))
```

```
even (2, 4, 6, 8, 14, 16)
odd (1, 7, 13)
e_sum 50
o_sum 21
e max,e min 16 2
o max,o min 13 1
```

In []:

In []: