

Strings

- collection of items or sequence of characters or group of characters.
- it is a derived datatype and immutable(unchangable-once defined they cannot be changes)
- Strings are most popular type in python.we are simply create by enclosing characters in quotations(" or """).
- python has a set of buildin methods that you can use on strings.
- all string methods return nwe values,but they do not change the original string.

```
In [1]: 1 a="string"
        2 print(type(a))
```

```
<class 'str'>
```

```
In [2]: 1 print(len(a))
        2 print(min(a))
        3 print(max(a))
        4 print(sorted(a))
        5 print(sum(a))
```

```
6
g
t
['g', 'i', 'n', 'r', 's', 't']
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-2-1eca4d6d1ad2> in <module>
      3 print(max(a))
      4 print(sorted(a))
----> 5 print(sum(a))

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
In [ ]: 1 s="python"
        2 s1="workshop"
        3 print(s+s1)
        4 print(s,s1)
        5 print(s*3)#repetation
```

```
In [ ]: 1 #Slicing (positive indexing-->(left to right))
        2 s="python workshop"
        3 print(len(s))
        4 print(s[0])
        5 print(s[2])
```

```
In [ ]: 1 #slicing(start,end,step)
2 s
3 print(s[0:3])
4 print(s[:])
5 print(s[:5])
6 print(s[7:])
7 print(s[::2])
8 print(s[1::3])
```

```
In [ ]: 1 #Negative indexing(right to left)
2 print(s[-1])
3 print(s[-3])
```

```
In [ ]: 1 s
2 print(s[::-1])
3 print(s[-4:])
4 print(s[-1:-4])
5 print(s[::-1])
```

```
In [3]: 1 print(dir(str),end=" ")
```

```
['_add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '__getnewa
rgs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__l
e__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce
__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__siz
eof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'cou
nt', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'inde
x', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'i
slower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join',
'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rind
ex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startsw
ith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

```
In [4]: 1 s='python' #It converts the first char to uppercase
2 print(s.capitalize())
```

Python

```
In [5]: 1 s='python workshop' #converts the first char of each word to uppercase
2 print(s.title())
```

Python Workshop

```
In [6]: 1 s="Hello hii" #converts the lowercase
2 s.casefold()
```

Out[6]: 'hello hii'

```
In [7]: 1 print(s.lower()) #convert a string into lowercase
        2 print(s.upper()) #convert into uppercase
```

```
hello hii
HELLO HII
```

```
In [8]: 1 s
```

```
Out[8]: 'Hello hii'
```

```
In [9]: 1 s1="python"
        2 print(s1.startswith('p')) #Returns true if a string starts with the specifi
        3 print(s1.endswith('n'))
```

```
True
True
```

```
In [10]: 1 s2='python programming'
         2 print(s2.count('m')) #Returns the number of times a specified value occur
```

```
2
```

```
In [11]: 1 print(s2.index('p')) #Search the str for a specified value and returns the p
```

```
0
```

```
In [12]: 1 print(s2.rindex('p'))#search the str for a specified value and returns the l
```

```
7
```

```
In [13]: 1 s3="python123" #returns true if all char in str are in the alphabet.
        2 print(s3.isalpha())
```

```
False
```

```
In [14]: 1 print(s3.isdigit()) #returns true if all char in str are in digits.
```

```
False
```

```
In [15]: 1 print(s3.isalnum())#returns true if all char in the string alphanumeric.
```

```
True
```

```
In [16]: 1 l='1234' #returns true if all cahr in the string are decimals
        2 print(l.isdecimal())
```

True

```
In [17]: 1 a='_python' #returns true if the str is an identifier
        2 print(a.isidentifier())
```

True

```
In [18]: 1 a1='PyThOn'
        2 print(a1.isupper())
        3 print(a1.islower())
```

False

False

```
In [19]: 1 b='wekwjrjenc' #returns true if all char in string are printable
        2 print(b.isprintable())
```

True

```
In [20]: 1 a='57554489' #returns true if all char in the string are numeric.
        2 print(a.isnumeric())
```

True

```
In [21]: 1 r="      " #returns true if all char in str are spaces
        2 print(r.isspace())
```

True

```
In [22]: 1 a="python workshop"
        2 print(a.find('o')) #search a str for a specified value and returns the posit
```

4

```
In [23]: 1 print(a.rfind('o'))#search a str for a specified value and returns the Last
```

13

```
In [24]: 1 s="    python    " #remove left side spaces
        2 print(s.lstrip())
```

python

In [64]: 1 `print(s.rstrip())` *#remove right side spaces*

python

In [65]: 1 `print(s.strip())` *#both side spaces are remove*

python

In [66]: 1 `s="PyThOn"` *#change the lowercase become uppercase and upper case become Lowercase*
2 `print(s.swapcase())`

pYtHoN

In [67]: 1 `a="python workshop"` *#splits the string at the specified separator, and returns a list*
2 `print(a.split())`

['python', 'workshop']

In [68]: 1 `a="python@workshop"`
2 `print(a.split('@'))`

['python', 'workshop']

In [71]: 1 `a1="p y t h o n"`
2 `print(a1.split())`

['p', 'y', 't', 'h', 'o', 'n']

In [72]: 1 `print(a.split('o'))`

['pyth', 'n@w', 'rksh', 'p']

In [74]: 1 `b='p','y','t','h','o','n'`
2 `print("".join(b))` *#join the elements of an iterable to the end of the string*

python

In [75]: 1 `print("@".join(b))`

p@y@t@h@o@n

In [76]: 1 `c="my name is {name}".format(name="Mango")`
2 `print(c)` *#formats the specified values in the string.*

my name is Mango

```
In [79]: 1 d="ynthon" #returns a str where a specified value is replaced with specifeif
          2 print(d.replace('y','p'))
```

pnthon

```
In [80]: 1 s="python" #fills tehe string with a specified number of 0 values at the beg
          2 print(s.zfill(10))
```

0000python

```
In [86]: 1 s="python workshop" #return a tuple where the str is parted into three parts
          2 print(s.partition('t'))
          3 print(s.rpartition('h'))
```

('py', 't', 'hon workshop')
('python works', 'h', 'op')

```
In [88]: 1 s1="python" #returns a center in given string
          2 print(s1.center(20))
          3 print(s1.center(20,"*"))#it returns the original str with fill the character
```

python
*****python*****

```
In [89]: 1 a='hello\thi\t' #set the tabsize of the str
          2 print(a.expandtabs())
```

hello hi

```
In [91]: 1 s1='hi iam {0},having id {1}'
          2 print(s1.format('kiran','404'))
```

hi iam kiran,having id 404

```
In [95]: 1 #i/p:a p s s d c
          2 #o/p:Apssdc
          3 s='a p s s d c'
          4 a=s.split()
          5 print("".join(a).capitalize())
```

Apssdc

```
In [97]: 1 x="Hello"
          2 x.encode()
          3 print(x)
```

Hello

- 2tasks: 1.s="abcd123@" o/p:no.of alphabets,no.of digits,no.of special char 2.s="python" o/p:even ascivalues char p,t,h,n