strings

- grp of or sequence of characters are known as strings
- strings are immutable(un changeable)
- represtented with '' , " "

In [1]:
```
a=12
type(a)
```

Out[1]: int

In [2]:
```
s=input('enter a value')
print(s)
type(s)
```

```
enter a value3
3
```

Out[2]: str

In [6]:
```
s=int(input('enter a value'))
print(s)
type(s)
```

```
enter a value43
43
```

Out[6]: int

In [7]:
```
s="hello ece"
len(s)
```

Out[7]: 9

In [12]:
```
s[5]
```

Out[12]: ' '

In [16]:
```
#string slicing
s[1:8]
```

Out[16]: 'ello ec'

In [23]:
```
# to get the last value of the given string
s[-3]
```

Out[23]: 'e'

In [25]:
```
print(dir(str),end='')
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getne
wargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__',
'__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__
reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr_
_', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'c
enter', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'forma
t_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'is
identifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'is
upper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replac
e', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 's
plitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper',
'zfill']
```

In [29]:
```
s='abc120'
s.isalnum()
```

Out[29]: True

In [30]:
```
s.isdigit()
```

Out[30]: False

In [31]:
```
s.isnumeric()
```

Out[31]: False

In [32]:
```
s.isalpha()
```

Out[32]: False

In [33]:
```
s='hello ece'
s.capitalize()
```

Out[33]: 'Hello ece'

In [34]:
```
s.casefold()
```

Out[34]: 'hello ece'

In [37]:
```
s1='HELLO ECE'
s1.casefold()
```

Out[37]: 'hello ece'

In [38]:
```
s1.lower()
```

Out[38]: 'hello ece'

In [44]:
```
s1.center(5)
```

Out[44]: 'HELLO ECE'

In [53]:
```
s2='problem solving and programming in python'
s2.count('g')
```

Out[53]: 3

In [63]:
```
s2.find('py')
```

Out[63]: 35

In [64]:
```
s2.find('ph')
```

Out[64]: -1

In [65]:
```
s1='gowri'
s2='bindu'
s1.join(s2)
```

Out[65]: 'bgowriigowringowridgowriu'

split method

- a split method is used to split a string into two method

In [70]:
```
s2.split('u')
```

Out[70]: ['bind', '']

In [71]:
```
s2='hello world'
s2[0]
```

Out[71]: 'h'

In [73]:
```
s2=s2.split('')
s2
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-73-d2379f0f7d38> in <module>
----> 1 s2=s2.split('')
      2 s2

ValueError: empty separator
```

In [8]:
```
# in='python work shop'
#o/p=w. python
st=input('enter the value')
#print(len(st))
st=st.split()
#print(st)
print(st[1][0]+'.',st[0])
```

```
enter the valuepython workshop
w. python
```

In [78]:
```
len(st)
```

Out[78]: 2

In [15]:
```
#string reversal
s='hello bindu'
s[::-1]
```

Out[15]: 'udnib olleh'

In [20]:
```
#find the end value
s='jessy star'
s.endswith('r')
```

Out[20]: True

In [25]:
```
# strip  - to remove unwanted space
s1='hello bindu'
s1.strip()
```

Out[25]: 'hello bindu'

In [30]:
```
#upper and lower case of the string
#s1.upper()
s1.lower()
```

Out[30]: 'hello bindu'

In [31]:
```
s1.title()
```

Out[31]: 'Hello Bindu'

In [32]:
```
s1.swapcase()
```

Out[32]: 'HELLO BINDU'

In [35]:
```
s1='Hello JESSY'
s1.swapcase()
```

Out[35]: 'hELLO jessy'

data structures in python

- lists
- tuples
- dictionaries
- sets

lists

- list is a collection of data of different data types
- lists are mutable
- represented in [] , comma seperated values

In [40]:
```
li=[]
type(li)
```

Out[40]:
```
list
```

In [44]:
```
li=[1,2,3,4,5,6,7,8,9,'a','abc']
li[0]
```

Out[44]:
```
1
```

In [45]:
```
li[-1]
```

Out[45]:
```
'abc'
```

In [46]:
```
li[::-1]
```

Out[46]:
```
['abc', 'a', 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

In [51]:
```
li1=[1,2,3,4,5,6,7]
print(max(li1))
print(min(li1))
print(sum(li1))
```

```
7
1
28
```

In [52]:
```
del(li1)
```

In [53]:
```
li1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-53-ac6881d35ffe> in <module>
----> 1 li1

NameError: name 'li1' is not defined
```

In [57]:
```
print(dir(list),end='')
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir
__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__geti
tem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_sub
class__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__
new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul_
_', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook_
_', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']
```

In [61]:
```
l1=[1,2,3,4,'a','b','c']
l1.append('jessy')
print(l1)
```

```
[1, 2, 3, 4, 'a', 'b', 'c', 'jessy']
```

In [68]:
```
# add the two lists
l2=['abc',1,4]
l2.append(l1)
l2
```

Out[68]:
```
['abc',
 1,
 4,
 [1,
  2,
  3,
  4,
  'a',
  'b',
  'c',
  'jessy',
  ['abc', 1, 4],
  ['abc', 1, 4],
  'abc',
  1,
  4,
  [...]]]
```

In [70]:
```
#add the elements in the list one by one
l1.extend(l2)
l1
```

Out[70]:
```
[1,
 2,
 3,
 4,
 'a',
 'b',
 'c',
 'jessy',
 ['abc', 1, 4],
 ['abc', 1, 4],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...]]
```

In [71]:
```
# copy the elements from one list into another list
l2=l1.copy()
```

In [73]:
```
print(l2)
```

```
[1, 2, 3, 4, 'a', 'b', 'c', 'jessy', ['abc', 1, 4], ['abc', 1, 4], 'abc', 1,
4, [1, 2, 3, 4, 'a', 'b', 'c', 'jessy', ['abc', 1, 4], ['abc', 1, 4], 'abc',
1, 4, [...], 'abc', 1, 4, [...], 'abc', 1, 4, [...]], 'abc', 1, 4, [1, 2, 3,
4, 'a', 'b', 'c', 'jessy', ['abc', 1, 4], ['abc', 1, 4], 'abc', 1, 4, [...],
'abc', 1, 4, [...], 'abc', 1, 4, [...]], 'abc', 1, 4, [1, 2, 3, 4, 'a', 'b',
'c', 'jessy', ['abc', 1, 4], ['abc', 1, 4], 'abc', 1, 4, [...], 'abc', 1, 4,
[...], 'abc', 1, 4, [...]]]
```

In [76]:
```
l1.count(1)
```

Out[76]: 4

In [77]:
```
l1.count(10)
```

Out[77]: 0

In [83]:
```
# index - is used to return the position of the value
l1.index('jessy')
```

Out[83]: 7

In [90]:
```
#splitting of strings
s='himagowri'
s[0]
#s[0]='h'
s=s.split('i')
print(s)
print(s[0])
s[1]='hello'
print(s)
```

```
['h', 'magowr', '']
h
['h', 'hello', '']
```

In [91]:
```
l1[1]='apssdc'
```

In [92]:
```
l1
```

Out[92]:
```
[1,
 'apssdc',
 3,
 4,
 'a',
 'b',
 'c',
 'jessy',
 ['abc', 1, 4],
 ['abc', 1, 4],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...]]
```

In [94]:
```
l1.insert(7,5)
l1
```

Out[94]:
```
[1,
 'apssdc',
 3,
 4,
 'a',
 'b',
 'c',
 5,
 5,
 'jessy',
 ['abc', 1, 4],
 ['abc', 1, 4],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...]]
```

In [96]:
```
#pop is used to remove the last element
l1.pop()
```

Out[96]:
```
[1,
 'apssdc',
 3,
 4,
 'a',
 'b',
 'c',
 5,
 5,
 'jessy',
 ['abc', 1, 4],
 ['abc', 1, 4],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4]
```

In [97]:
```
l1
```

Out[97]:
```
[1,
 'apssdc',
 3,
 4,
 'a',
 'b',
 'c',
 5,
 5,
 'jessy',
 ['abc', 1, 4],
 ['abc', 1, 4],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4,
 [...],
 'abc',
 1,
 4]
```

In [103]:
```
l1=[1,2,3,'a','b','c']
```

In [107]:
```
l1
```

Out[107]:
```
[2, 3, 'a', 'b', 'c']
```

In [108]:
```
l1.remove(2)
l1
```

Out[108]:
```
[3, 'a', 'b', 'c']
```

In [111]:
```
# reverse--used to reverse the elements in the list
l=[1,6,3,2,9]
l.reverse()
l
```

Out[111]:
```
[9, 2, 3, 6, 1]
```

In [112]:
```
l.sort(reverse=True)
l
```

Out[112]:
```
[9, 6, 3, 2, 1]
```

In [114]:
```
l.clear()
l
```

Out[114]:
```
[]
```

In [ ]: