String

string is a collection or group of characters

In [1]:

```
name = "python"
```

In [2]:

```
sentence = """This is my Line1
this is my line2"""
```

In [3]:

```
print(sentence)
```

```
This is my Line1
this is my line2
```

In [4]:

```
sentence = '''Line 1
Line 2
Line3
'''
```

In [5]:

```
print(sentence)
```

```
Line 1
Line 2
Line3
```

```
python programming
```

forward index 0 len(string)-1
reverse index -len(string) -1

In [6]:

```
name = "python programming"
```

```
str_variable[index]
```

In [7]:

```
print(name[0])
```

```
p
```

In [8]:

```
print(name[1])
```

```
y
```

In [9]:

```
print(name[-1])
```

```
g
```

How many characters in my string

In [10]:

```
len(name)
```

Out[10]:

```
18
```

In [11]:

```
print(name[17])
```

```
g
```

In [12]:

```
print(name[25])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call las
t)
<ipython-input-12-43d13d678c67> in <module>
----> 1 print(name[25])

IndexError: string index out of range
```

In [13]:

```
print(name[-18])
```

```
p
```

Strings are immutable

mutable means we can modify after creation

Immutable means we can not modify

In [14]:

```
name
```

Out[14]:

```
'python programming'
```

In [15]:

```
name[0] = 'j'
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-15-e3626fd8db2b> in <module>
----> 1 name[0] = 'j'

TypeError: 'str' object does not support item assignment
```

for loop for strings also

In [ ]:

```
for each in name:
    print(each)
```

string slicing

getting sub string from the original string

syntax

```
str_variable[startindex:endindex_excluded:step]
```

In [ ]:

```
name
```

start=0 ending = 6-1=5 step =2

0 2 4

p t o

In [ ]:

```
name[0:6:1]
```

In [ ]:

```
name[0:6:2]
```

In [ ]:

```
starting = len(name)-1
ending = -len(name)-1
step = -1
```

In [ ]:

```
name[starting:ending:-1]
```

In [ ]:

```
name[::-1]
```

In [ ]:

```
name[::-1]
```

here i am not specified staring index it can assume 0

based step ending index 5 -1 =4

In [ ]:

```
name[:5]
```

In [ ]:

```
name[0:5]
```

In [ ]:

```

```

In [ ]:

```

```

for 0 1 2 3 4 5 6 7 8 9 10
p y t h o n " p r o g rev -10 -9 -8 -7 -6 -5 -4 -3 -2 -1

if i will decrement value from -1 by giving -1 am i get 5 value

In [ ]:

```
name = 'python prog'
```

In [ ]:

```
name[-1:6:-1]
```

Working with string methods

I want to check my name staring with substring or not?

In [ ]:

```
name = "python programming" # is this staring with 'python'?
```

actual_str.startswith(sub_str)

    if True else False

In [ ]:

```
name.startswith('python')
```

actual_str.endswith(sub_str)

In [ ]:

```
name.endswith('ing')
```

In [ ]:

```
name.endswith('inG')
```

str.isdigit()

    if all characters are digits then return True otherwise False

In [ ]:

```
'1234'.isdigit()
```

In [ ]:

```
'123a123'.isdigit()
```

str.lower()

In [ ]:

```
name = 'AyyaPPA'

name.lower()
```

In [ ]:

```
print(name)
```

None

    string methods can not modify origanl string

replace o with uppercase O

In [ ]:

```
name = 'python'
```

In [ ]:

```
res = ''
for each in name:
    if each == 'o':
        value = 'O'
    else:
        value = each
    res += value
```

In [ ]:

```
print(res)
```

upper() it converts all the characters into uppercase

In [ ]:

```
name = 'ayyappa'
name.upper()
```

How to verify all the characters are lower case or not?

In [ ]:

```
name = 'python'
```

islower()

```
    if all characters are lower return True
    otherwise False
```

In [ ]:

```
name.islower()
```

In [ ]:

```
'Ayyappa'.islower()
```

isupper()

```
        if all the characters are upper case return True
        else return False
```

In [ ]:

```
'AYYAPPA'.isupper()
```

replace(old_character,new)

In [ ]:

```
name = 'python programming'

name = name.replace('o','O')
print(name)
```

In [ ]:

```

```

In [ ]:

```
name = '       python        '
```

I want remove spacing characters from left

str.lstrip()

    it remove string spacing characters

In [ ]:

```
print(name)
print(name.lstrip())
```

str.rstrip()

remove ending spaces

In [ ]:

```
print(name)
print(len(name))
```

In [ ]:

```
res = name.rstrip()

print(res)
print(len(res))
```

strip()

    remove both staring and ending spaces

In [16]:

```
print(name)
```

```
python programming
```

In [17]:

```
print(name.strip())
```

```
python programming
```

In [18]:

```
name = 'python'
print(name)
print(name)
```

```
python
python
```

How to convert sentence into words

In [19]:

```
sentence = "every one has story"
```

str.split()

    it can cut the sentence at the spacing character

In [20]:

```
res = sentence.split()
print(res)
```

```
['every', 'one', 'has', 'story']
```

In [21]:

```
print(type(res))
```

```
<class 'list'>
```

How to cut sentence at required character

In [22]:

```
sentence = "python program in apssdc"
```

In [23]:

```
sentence.split('a')
```

Out[23]:

```
['python progr', 'm in ', 'pssdc']
```

find the sum of digits in given string

In [24]:

```
string = '1234'

# implement logic
# result 1+2+3+4 = 10
```

In [25]:

```
res = 0
for each in string:
    if each.isdigit():
        res += int(each)
print(res)
```

```
10
```

How to conert words into sentence

In [26]:

```
words = [ 'every' ,'one','has','story'  ]
```

str_seperator.join(collection)

In [27]:

```
''.join(words)
```

Out[27]:

```
'everyonehasstory'
```

i want to find the index of character

str.find(sub_str)

```
    return index of charcter if available else return -1
```

In [28]:

```
name = 'python'
name.find('t')
```

Out[28]:

```
2
```

In [29]:

```
name.find('j')
```

Out[29]:

```
-1
```

i want to find how many times character available

str.count(sub_str)

In [30]:

```
name = 'python program'
name.count('p')
```

Out[30]:

```
2
```

In [31]:

```
name.count('j')
```

Out[31]:

```
0
```

find(sub_str,starIndex,endIndex)

```
    here searching character starts from starting index
```

In [32]:

```
name = 'python programming'
```

In [33]:

```
name.find('p',4) # search character from index 4
```

Out[33]:

```
7
```

what is the index of 'p' second occurs

In [34]:

```
firstIndex = name.find('p')
print(firstIndex)
```

```
0
```

In [35]:

```
name.find('p',firstIndex+1)
```

Out[35]:

```
7
```

In [36]:

```
name.count('p')
```

Out[36]:

```
2
```

In [37]:

```
name[7]
```

Out[37]:

```
'p'
```

In [38]:

```
name = 'program program'
```

In [39]:

```
name.count('program')
```

Out[39]:

```
2
```

center to make your string center

In [40]:

```
name = 'python'
name.center(15)
```

Out[40]:

```
'     python    '
```

Casefold

In [41]:

```
name = 'AYYAPPA'
name.casefold()
```

Out[41]:

```
'ayyappa'
```

swapcase

convert lower case into upper and upper into lower

In [42]:

```
name = 'AyYappa'
```

In [43]:

```
name.swapcase()
```

Out[43]:

```
'aYyAPPA'
```

isalpha() to check characters are alphabets or not

In [44]:

```
name = 'python'
name.isalpha()
```

Out[44]:

```
True
```

In [45]:

```
'password123'.isalpha()
```

Out[45]:

```
False
```

isalnum()

to verify all the characters are alphabes and numbers

In [46]:

```
name ='password'
name.isalnum()
```

Out[46]:

```
True
```

In [47]:

```
name='password123'
name.isalnum()
```

Out[47]:

```
True
```

str.title()

In [48]:

```
name = 'every one has story'
name.title()
```

Out[48]:

```
'Every One Has Story'
```

ljust(width,replace_value)

In [49]:

```
name = 'python'
name.ljust(15)
```

Out[49]:

```
'python         '
```

In [50]:

```
name = 'python'
name.ljust(15,'@')
```

Out[50]:

```
'python@@@@@@@@@'
```

rjust(width,replace_value)

In [51]:

```
name = "python"
name.rjust(15,'@')
```

Out[51]:

```
'@@@@@@@@@python'
```

List

```
List is a collection items
```

to create a list 2 ways

1. symbols [] square braces
2. list() function

In [52]:

```
students_name = []
```

How many items in List

In [53]:

```
print(len(students_name))
```

```
0
```

In [54]:

```
print(type(students_name))
```

```
<class 'list'>
```

In [55]:

```
student_names = list()
```

In [56]:

```
print(type(student_names))
```

```
<class 'list'>
```

In [57]:

```
student_names = ['sairam','lakshimi','ram']
```

In [58]:

```
len(student_names)
```

Out[58]:

```
3
```

we can apply indexing for list also

In [59]:

```
student_names[0]
```

Out[59]:

```
'sairam'
```

In [60]:

```
student_names[-1]
```

Out[60]:

```
'ram'
```

list slicing

In [61]:

```
student_names[0:3]
```

Out[61]:

```
['sairam', 'lakshimi', 'ram']
```

In [62]:

```
student_names[::-1]
```

Out[62]:

```
['ram', 'lakshimi', 'sairam']
```

In [63]:

```
print(student_names)
```

```
['sairam', 'lakshimi', 'ram']
```

List is mutable

In [64]:

```
student_names
```

Out[64]:

```
['sairam', 'lakshimi', 'ram']
```

replace item in the list

```
    list_variable[index] = newValue
```

In [65]:

```
student_names
```

Out[65]:

```
['sairam', 'lakshimi', 'ram']
```

In [66]:

```
student_names[1] = 'python'
```

In [67]:

```
student_names
```

Out[67]:

```
['sairam', 'python', 'ram']
```

In [ ]:

```

```

In [ ]:

```
```

List methods

list.append(item)

    to add the item into end of list

In [68]:

```
students = ['sai','ram','ramu']
```

In [69]:

```
students.append('lakshmi')
```

In [70]:

```
students
```

Out[70]:

```
['sai', 'ram', 'ramu', 'lakshmi']
```

In [ ]:

```
```

In [71]:

```
var = [10,20]
```

In [72]:

```
students.append(var)
```

In [73]:

```
students
```

Out[73]:

```
['sai', 'ram', 'ramu', 'lakshmi', [10, 20]]
```

In [74]:

```
print(len(students))
```

```
5
```

extend(collection)

In [75]:

```
students
```

Out[75]:

```
['sai', 'ram', 'ramu', 'lakshmi', [10, 20]]
```

In [76]:

```
students.extend(var)
```

In [77]:

```
students
```

Out[77]:

```
['sai', 'ram', 'ramu', 'lakshmi', [10, 20], 10, 20]
```

In [ ]:

```

```

copy()

In [78]:

```
students
```

Out[78]:

```
['sai', 'ram', 'ramu', 'lakshmi', [10, 20], 10, 20]
```

In [79]:

```
backup = students.copy()
```

In [80]:

```
backup
```

Out[80]:

```
['sai', 'ram', 'ramu', 'lakshmi', [10, 20], 10, 20]
```

count(item)

In [81]:

```
numbers = [10,20,30,40,20]
numbers.count(20)
```

Out[81]:

```
2
```

index(item)

to find position/index of the given item

In [82]:

```
numbers
```

Out[82]:

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

In [83]:

```
numbers.index(20)
```

Out[83]:

```
1
```

insert(index,item)

insert the given item at given index

In [84]:

```
numbers
```

Out[84]:

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

In [85]:

```
numbers.insert(3,100)
```

In [86]:

```
print(numbers)
```

```
[10, 20, 30, 100, 40, 20]
```

what happens if given index not available

In [87]:

```
numbers
```

Out[87]:

```
[10, 20, 30, 100, 40, 20]
```

In [88]:

```
numbers.insert(100,1000)
```

In [89]:

```
numbers
```

Out[89]:

```
[10, 20, 30, 100, 40, 20, 1000]
```

clear()

    to remove all the items in the list

In [90]:

```
numbers
```

Out[90]:

```
[10, 20, 30, 100, 40, 20, 1000]
```

In [91]:

```
numbers.clear()
```

In [92]:

```
numbers
```

Out[92]:

```
[]
```

pop(index)

    remove the item at given index and return removed value

In [93]:

```
numbers = [10,20,30]
print(numbers.pop(1))
print(numbers)
```

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

In [94]:

```
numbers.pop(10)
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
<ipython-input-94-dbe076c6be3d> in <module>
----> 1 numbers.pop(10)

IndexError: pop index out of range
```

In [95]:

```
numbers
```

Out[95]:

```
[10, 30]
```

In [96]:

```
numbers.pop()
```

Out[96]:

```
30
```

In [97]:

```
numbers
```

Out[97]:

```
[10]
```

remove(item)

    remove given item from list

In [98]:

```
numbers = [10,20,30,40]
numbers.remove(20)
```

In [99]:

```
numbers
```

Out[99]:

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

In [100]:

```
numbers.remove(20)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-100-2eabd9e39779> in <module>
----> 1 numbers.remove(20)

ValueError: list.remove(x): x not in list
```

In [101]:

```
numbers = [10,20,30,20,20]
```

In [102]:

```
numbers.remove(20)
```

In [103]:

```
numbers
```

Out[103]:

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

In [104]:

```
numbers = [10,20,50,40,70]
```

sort()

In [105]:

```
numbers.sort()
```

In [106]:

```
numbers
```

Out[106]:

```
[10, 20, 40, 50, 70]
```

In [107]:

```
numbers.sort(reverse=True)
```

In [108]:

```
numbers
```

Out[108]:

```
[70, 50, 40, 20, 10]
```

reverse()

In [109]:

```
numbers = [10,20,50,40]
numbers.reverse()
```

In [110]:

```
numbers
```

Out[110]:

```
[40, 50, 20, 10]
```

In [111]:

```
numbers = [10,'sai',True]
```

In [112]:

```
numbers
```

Out[112]:

```
[10, 'sai', True]
```