string_operations

May 20, 2024

[1]:
```
iti = 'information technology institute'
print(iti,type(iti))
```

```
information technology institute <class 'str'>
```

1- get len of string

[2]:
```
print(len(iti))
```

```
32
```

count char occurence in the string

[3]:
```
print(iti.count('i'))
```

```
4
```

string is treated like an array » each char is assigned to index –> start from 0 get index of char

[5]:
```
print(iti.index('t')) # return with index of the first occurrence (first hit)
print(iti.count('t'))
```

```
7
5
```

slicing string

[6]:
```
iti= 'information technology institute'
print(iti[1:6])
```

```
nform
```

[7]:
```
print(iti[3:])
```

```
ormation technology institute
```

[8]:
```
print(iti[::])
```

```
information technology institute
```

[9]:
```
print(iti[::-1])
```

```
etutitsni ygolonhcet noitamrofni
```

[10]:
```
print(iti[2::2])
```

```
fraintcnlg nttt
```

string concat

[12]:
```
fname= 'noha'
midname= 'abdelhady'
lname='shehab'
fullname = fname + ' ' + midname + ' ' + midname +" "+lname
print(fullname)
```

```
noha abdelhady abdelhady shehab
```

string interpolation

[13]:
```
fullname = fname + ' ' + (midname + ' ')*2 +lname
```

[14]:
```
print(fullname)
```

```
noha abdelhady abdelhady shehab
```

[15]:
```
print('iti_'*10)
```

```
iti_iti_iti_iti_iti_iti_iti_iti_iti_iti_
```

Format string

[16]:
```
no_of_students = 25
course_name= 'python'
msg = 'we have {0} students studies {1} course'
print(msg)
```

```
we have {0} students studies {1} course
```

format string

[17]:
```
newmsg = msg.format(no_of_students,course_name)
print(newmsg)
```

```
we have 25 students studies python course
```

[19]:
```
newmsg2= msg.format(course_name, no_of_students)
print(newmsg2)
```

```
we have python students studies 25 course
```

define for template using keywords

[20]:
```
msg = 'we have {anynum} students studies {anycourse} course'
print(msg)
```

```
we have {anynum} students studies {anycourse} course
```

[21]:
```
newmsg3= msg.format(anynum=no_of_students, anycourse=course_name)
print(newmsg3)
```

```
we have 25 students studies python course
```

format string –> f-format

[22]:
```
newmsg4= f'we have {no_of_students}, studies {course_name} course'
print(newmsg4)
```

```
we have 25, studies python course
```

format string functions always returns with string

[23]:
```
m1= f'My name is {username}'  # f-format depends on existing variables in memory
print(m1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[23], line 1
----> 1 m1= f'My name is {username}'
      2 print(m1)

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

[24]:
```
m2='My name is {username}'
print(m2)
```

```
My name is {username}
```

[25]:
```
age =31
name='test'
bio = name +  age
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 3
      1 age =31
      2 name='test'
----> 3 bio = name +  age

TypeError: can only concatenate str (not "int") to str
```

[26]:
```
bio = name ,  age
```

[27]:
```
bio
print(bio)
```

```
('test', 31)
```

String is immutable datatype ? change string format

[31]:
```
print(msg)
msg = msg.format(anynum='rrr', anycourse='33')
print(msg)
```

```
we have {anynum} students studies {anycourse} course
we have rrr students studies 33 course
```

changes on string

[32]:
```
fullname = 'noha abdelhady shehab'
print(fullname.upper())
```

```
NOHA ABDELHADY SHEHAB
```

[33]:
```
print(fullname.lower())
```

```
noha abdelhady shehab
```

[34]:
```
print(fullname.title())
```

```
Noha Abdelhady Shehab
```

[35]:
```
print(fullname.capitalize())
```

```
Noha abdelhady shehab
```

strip string

[36]:
```
message  =' we love python                              '
print(len(message))
```

```
40
```

[37]:
```
res = message.strip()
print(res, len(res))
```

```
we love python 14
```

[38]:
```
res2 = message.rstrip() # strip white spaces from right
print(res2, len(res2))
```

```
 we love python 15
```

strip set of chars ???

[42]:
```
message = '@   we love python   @@'
print(message.strip('@ nw')) # strip these chars from the beginning and the end␣
 ↪of the string
```

```
e love pytho
```

replace part of the string

[45]:
```
content = 'we love python o o o o'  # o---> @
print(content.replace('o', '@',2)) # this returns with new string
```

```
we l@ve pyth@n o o o o
```

examine string content

[46]:
```
num = "10"
num = int(num)  # it works because content of the string is numeric
print(num, type(num))
```

```
10 <class 'int'>
```

[47]:
```
name = 'noha'
name = int(name)  # raise runtime error
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[47], line 2
      1 name = 'noha'
----> 2 name = int(name)  # raise runtime error

ValueError: invalid literal for int() with base 10: 'noha'
```

[ ]:
```
# check content of the string befor opertaion
```

make sure the string content consists of numbers only

[51]:
```
num = '89274'
print(num.isdigit()) # return True if string consists of only numbers␣
 ↪(integers)
```

```
True
```

[52]:
```
if num.isdigit():
    num = int(num)
    print("-- converted ")
else:
    print("--cannot convert it ")
```

```
-- converted
```

ask user to enter value

[53]:
```
anyval = input('Enter any number/ value: ')    # always return with string
print(anyval, type(anyval))
```

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

check if string consists only from alphas [a—z]

[54]:
```
name = 'ahmed10'
print(name.isalpha())
```

```
False
```

[55]:
```
name='noha shehab' # this string contains space
print(name.isalpha())
```

```
False
```

NEVER TRUST USER INPUT

[59]:
```
anyval = int(input('Enter any number/ value: ')) # not accurate
print(anyval, type(anyval))# always return with string
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[59], line 1
----> 1 anyval = int(input('Enter any number/ value: '))
      2 print(anyval, type(anyval))# always return with string

ValueError: invalid literal for int() with base 10: 'eee'
```

[ ]:
```
num = input('Enter any number/ value: ')
if num.isdigit():
    num = int(num)
    print("-- converted ")
```

check state of string

[60]:
```
message ='iti'
print(message.isupper())
print(message.islower())
print(message.isspace())
print(message.istitle())
```

```
False
True
False
False
```

check if char exists in string –> in operator

[61]:
```
name = 'noha'
print('n' in name)
```

```
True
```

loop over the string

[63]:
```
for char in 'noha':
    print(char)
```

```
n
o
h
a
```

”noha ??? 2”

’noha’ ??? ==?nh

iti —> 0 , 2