Data Science & AI
&
NIC - Param

Python-For Data Science

String

Lecture No.- 01

By- Pankaj Sharma Sir

Recap of Previous Lecture

Topic Control Flow Statement - 04

Slide 2

Topics to be Covered

Topic Strings Part-01

Slide 3

Topic : Strings

String

Python

'Pankaj'

"Neeraj"

''' Pankaj sir is worst faculty''

""" Pankaj sir is rawon type faculty"""

6 → 0 - - 5

```
for i in range(len(s)):
    print(i, s[i])
```

	0	1	2	3	4	5
s =	P	a	n	k	a	j
	-6	-5	-4	-3	-2	-1

0	p
1	a
2	n
3	k
4	a
5	j

	+ve	-ve
p	0	-6
a	1	-5
n	2	-4
k	3	-3
a	4	-2
j	5	-1

len(s) = 6

6 → 0 - - - 5

for i in range(len(s)):

~~print(i, s[i])~~

print(i - len(s), s[i - len(s)])

0 1 2 3 4 5

s = Pankaj

-6 -5 -4 -3 -2 -1

len(s) = 6

i	i - len(s)	s[i - len(s)]
0	-6	P
1	-5	a
2	-4	n
3	-3	k
4	-2	a
5	-1	j

Slicing

s = "Pankaj"
0 1 2 3 4 5

→ forward (left to right)

s[startindex : end : step]

end-1

s[1 : 4] ⇒ index
1, 2, 3
ank

+ve index

① s[2 : 6] ⇒ 2 to 6-1
2 to 5
2, 3, 4, 5 (index)
nkaj

The default value of step is 1

s[1 : 4 : 1]

s[2 : 6 : 1]

① if we don't specify start ⇒ by default from begin

② if we don't specify end ⇒ by default till end of string

s = "Pankaj"
0 1 2 3 4 5

s[: 4] ⇒ Pank

↓

till (4-1) index 3

from begin ⟹ 3

s[1 :] → till end of string (len(s))

1, . . . 5

ankaj

s[:] → entire string

same as s

s[3 : 3] Empty

index 3 ≮ index 2 (forward)

←

0 1 2 3 4 5

s = "Pankaj"

⟶ index increase

index dec. ⟵

-ve value ⇒ backward direction

end

$s[4:2:-1]$

from index 4 to 2 (decreasing) ✓

4 to (2+1)

$\boxed{4,3}$

0 1 2 3 4 5

S = "Pankaj"

-6 -5 -4 -3 -2 -1

→ index increase

index dec. ←

S[1:4:-1] → -ve value ⇒ backward direction

⇓

but

no such string ""

Confusion

convert to
-ve index ⇒ +ve

Pankaj
0 1 2 3 4 5

$s[\underset{+6}{-2} : \underset{+6}{-5} : -1]$ backward

$s[4 : 1 : -1]$

$\boxed{4, 3, 2}$

s[: : -1]

Reverse

Comparision

"Neeraj" > "Pankaj"

N > P False

Character by character

"Pankaj" < "Sharma"

P < S ✓

True

"Ramesh" > "Ramu"

✓✓✓ ✓✓✓

e > u

False

" jahah" ⟹ to remove spaces from begin/end

space

rstrip ⟹ remove spaces from right of string
side

lstrip ⟹ remove spaces from left of string.
side

Problem solving

string

1. slicing
2. len( )
3. in ⇒ u can find a substring in given string
4. To remove spaces
   - (a) rstrip
   - (b) lstrip
   - (c) strip

t.me/PWpankajsirP

s len X

9 PM

Day 10 : strins -1

In [1]:

```
s='pankaj'
t="neeraj"
a='''pankaj sir is worst faculty'''
```

In [2]:

```
s
```

Out[2]:

```
'pankaj'
```

In [3]:

```
t
```

Out[3]:

```
'neeraj'
```

In [4]:

```
a
```

Out[4]:

```
'pankaj sir is worst faculty'
```

In [11]:

```
a='''Twinkle twinkle little star
 how i wonder what u are '''
```

In [12]:

```
a
```

Out[12]:

```
'Twinkle twinkle little star\n how i wonder what u are '
```

In [14]:

```
a="pankaj sharma's world of programming"
print(a)
```

```
pankaj sharma's world of programming
```

In [15]:

```
a='This is \'single quote and we are printing it'
```

In [16]:

```
print(a)
```

```
This is 'single quote and we are printing it
```

In [17]:

```
s="pankaj"
#  012345    index
#s[0]==>p
```

In [18]:

```
print(s[0])
```

```
p
```

In [19]:

```
print(s[5])
```

```
j
```

In [20]:

```
print(s[6])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[20], line 1
----> 1 print(s[6])

IndexError: string index out of range
```

In [21]:
```
s="pankaj"
#       -6-5-4-3-2-1
#        p a n k a j
```

In [22]:
```
print(s[-1]) #negative index
print(s[5]) #positive index
```

```
j
j
```

In [23]:
```
print(s[-2])
print(s[4])
```

```
a
a
```

In [24]:
```
s="pankaj"
#len ===> length of string dega
```

In [25]:
```
print(len(s))
```

```
6
```

In [26]:
```
for i in range(len(s)):
    print(i-len(s),s[i-len(s)])
```

```
-6 p
-5 a
-4 n
-3 k
-2 a
-1 j
```

In [27]:
```
s="pankaj"
```

In [28]:
```
print(s[1:])#from index 1 till end of string is printed(sliced)
```

```
ankaj
```

In [29]:
```
print(s[ :4])#from beginning of string till index number 3
```

```
pank
```

In [30]:
```
print(s[:])#entire string
```

```
pankaj
```

In [31]:
```
print(s[3:1])#by default forward ==>increasing index 3,2,1 is not inc
```

In [32]:
```
print(s[1:100])
```

```
ankaj
```

In [33]:
```
#slice operator
'''step can be positive or negative
   1)
   if it is positive ===>it should be in forward direction
   OR left to right within string
   and s[begin:end] will slice from begin till end-1 index.
   2)if it is negative  ==>reverse direction
   OR right to left in string
   and
   s[begin,end,]==>will slice from begin till end+1 with specified steps

'''
```

Out[33]:
```
'step can be positive or negative\n   1) \n   if it is positive ===>it should be in f
orward direction \n   OR left to right within string\n   and s[begin:end] will slice
from begin till end-1 index.\n   2)if it is negative  ==>reverse direction\n   OR rig
ht to left in string\n   and \n   s[begin,end,]==>will slice from begin till end+1 wi
th specified steps\n  \n'
```

In [34]:
```
#in forward direction
#default begin  0
#default end  len(string)
#default step : 1

#in backward direction
#default begin : -1
#default end : -(len(string)+1)

#index in any way can be negative or positive
```

In [35]:
```
s="pankaj"
s[4:2:-1] # -1 ===>direction is bacward or rigth to left
          # moving rTol ===>index decrease
          #4 ==>2 decreasing
          #4,3
```

Out[35]:
```
'ak'
```

In [36]:
```
s[1:4:-1]
```

Out[36]:
```
''
```

In [37]:
```
s[-2:-5:-1]#-2,-3,-4 ====>decrease
           #-1 backward direction
#from index -2 till index -5 +1  ==>-2,-3,-4
```

Out[37]:
```
'akn'
```

In [39]:
```
#multiply operator ==>repeatition
'pankaj'*3
```

Out[39]:
```
'pankajpankajpankaj'
```

In [40]:
```
#addition ==> concatenation
"pankaj" + "sharma"
```

Out[40]:

```
'pankajsharma'
```

In [41]:

```python
for i in s[::-1] :
    print(i)
```

```
j
a
k
n
a
p
```

In [42]:

```python
#membership operator ==> in
s="pankaj sharma is the worst faculty"
print("pankaj" in s)
```

```
True
```

In [43]:

```python
print("faculty in s")
```

```
faculty in s
```

In [44]:

```python
print("faculty" in s)
```

```
True
```

In [45]:

```python
#to find whether a substring is present in a string or not
#we have this built in function in python
s=input("enter original string")
t=input("enter the substring u want to find ")
if t in s:
    print("YES")
else:
    print("NO")
```

```
enter original stringpankaj sharma is rawan type faculty
enter the substring u want to find ram
NO
```

In [46]:

```python
s=input("enter original string")
t=input("enter the substring u want to find ")
if t in s:
    print("YES")
else:
    print("NO")
```

```
enter original string Pankaj sharma is the worst faculty
enter the substring u want to find Pankaj
YES
```

In [47]:

```python
#palindrome string only one word
s=input("enter one word string")
if s[::-1]==s :
    print("YES")
else:
    print("NO")
```

```
enter one word stringmalayalam
YES
```

In [48]:

```
#palindrome string only one word
s=input("enter one word string")
if s[::-1]==s :
    print("YES")
else:
    print("NO")
```

```
enter one word stringjahaj
YES
```

In [49]:

```
#palindrome string only one word
s=input("enter one word string")
if s[::-1]==s :
    print("YES")
else:
    print("NO")
```

```
enter one word string jahaj
NO
```

In [50]:

```
#in above code i provided string as spacejahaj
```

In [51]:

```
#string comparision is performed char by char
"pankaj">"sharma"
```

Out[51]:

```
False
```

In [52]:

```
"Ramesh">"Ramu" #==>e>u
```

Out[52]:

```
False
```

In [53]:

```
"Pankaj"=="pankaj"#capital P is different from small p
```

Out[53]:

```
False
```

In [54]:

```
"pankaj"<"Pankaj"#to proove that capiltal letters comes before small letters
```

Out[54]:

```
False
```

In [55]:

```
#removing spaces in string
#rstrip
s="pankaj   "#length ==>9
updated=s.rstrip()#rstrip is not a generalized method==>only for string
print(updated)
print(len(updated))
```

```
pankaj
6
```

In [56]:

```
#removing spaces in string
#lstrip
s="   pankaj"#length ==>9
updated=s.lstrip()#lstrip is not a generalized method==>only for string
print(updated)
print(len(updated))
```

```
pankaj
6
```

In [58]:

```
#removing spaces in string
#strip ==>rmove spaces from begining as well as from end of the string
s="   pankaj   "#length ==>12
updated=s.strip()#strip is not a generalized method==>only for string
print(len(s))
print(updated)
print(len(updated))
```

```
12
pankaj
6
```

In [ ]:

THANK - YOU