Python Programming

EDUCATION FOR PEACE

(ESTD-1995)

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

STRINGS IN PYTHON - II

Guido Van Rossum

Learning Mantra

If you really strong in the basics, then remaining things will become so easy.

Agenda:

1. **Replacing a string with another string.**
2. **Splitting of Strings.**

Replacing a string with another string:

- We can replace a string with another string in python using a library function **replace().**

Syntax:

```
s.replace(oldstring,newstring)
```

Here, inside '**s**', every occurrence of oldstring will be replaced with newstring.

Eg:

```
s="Learning Python is very difficult"
s1=s.replace("difficult","easy")
print(s1)          ➔Learning Python is very easy
```

Eg : All occurrences will be replaced

```
s="ababababababab"
print(id(s))                    ➔2168146994672
s1=s.replace("a","b")
print(id(s))                    ➔2168146994672
print(s1)                       ➔bbbbbbbbbbbbbb
```

Eg:

```
s="ababababababab"
print(id(s))                                        ➔2168149958000
s=s.replace("a","b") # two objects are created
print(id(s))                                        ➔2168149958128
print(s)                                            ➔bbbbbbbbbbbbbb
```

Q. String objects are immutable then how we can change the content by using replace() method?

Ans:

- Once we creates string object, we cannot change the content. This non changeable behavior is nothing but immutability. If we are trying to change the content by using any method, then with those changes a new object will be created and changes won't be happened in existing object.
- Hence with replace() method also a new object got created but existing object won't be changed.

Eg:

```
s="abab"
s1=s.replace("a","b")
print(s,"is available at :",id(s))
print(s1,"is available at :",id(s1))
```

Output:

```
abab is available at : 2168149950512
bbbb is available at : 2168149953312
```

- In the above example, original object is available and we can see new object which was created because of replace() method.

Eg : Consider the string : Python is easy but Java is difficult.

How can you replace the string 'difficult' with 'easy' and 'easy' with 'difficult‘?

```
s = 'Python is easy but Java is difficult'
s = s.replace('difficult','easy')
s = s.replace('easy','difficult')
print(s)                        # it is not giving correct output
```

Output:

```
Python is difficult but Java is difficult
```

```
s = 'Python is easy but Java is difficult'
s = s.replace('difficult','d1')
s = s.replace('easy','e1')
print(s)            ➔ Python is e1 but Java is d1
```

```
s = 'Python is easy but Java is difficult'
s = s.replace('difficult','d1')
s = s.replace('easy','e1')
s = s.replace('d1','easy')
s = s.replace('e1','difficult')
print(s)            ➔Python is difficult but Java is easy
```

Splitting of Strings:

- We can split the given string according to specified separator by using **split()** method.
- We can split the given string according to specified separator in reverse direction by using **rsplit()** method.

Syntax :

```
l=s.split(seperator, Maximum splits)
```

Here,

- Both parameters are optional
- The default separator is space.
- Maximum split defines maximum number of splits
- The return type of split() method is List.

Note:

- rsplit() breaks the string at the separator staring from the right and returns a list of strings.

Eg:

```
s="karthi sahasra sri"
l=s.split()                # Default separator (i.e., space) is taken
for x in l:
    print(x)
```

Output:

```
karthi
sahasra
sri
```

Eg:

```
s="22-04-2020"
l=s.split('-')
for x in l:
    print(x)
```

Output:

```
22
04
2020
```

Eg:

```
s="22-04-2020"
l=s.split( )
for x in l:
      print(x)
```

Output:

```
22-04-2020
```

Eg: Demonstration of rsplit() method

```
s = 'karthi sahasra sri nandyal india'
l=s.rsplit(' ',3)            # rsplit(): from reverse direction, it considers the given separator
for x in l:
    print(x)
```

Output:

```
karthi sahasra
sri
nandyal
india
```

Eg:

```
s = 'karthi sahasra sri nandyal ap india'
l=s.rsplit(' ',3)
for x in l:
    print(x)
```

Output:

```
karthi sahasra sri
nandyal
ap
india
```

Eg:

```
s = 'karthi sahasra sri nandyal india'
l=s.lsplit(' ',3)                    # There is no lsplit() method in python
for x in l:
    print(x)
```

AttributeError: 'str' object has no attribute **'lsplit'**

Eg:

```
s = '10,20,30,40,50,60,70,80'
l = s.split(',',3)
for x in l:
    print(x)
```

Output:

```
10
20
30
40,50,60,70,80
```

Eg:

```
s = '10,20,30,40,50,60,70,80'
l = s.rsplit(',',3)
for x in l:
print(x)
```

Output:

```
10,20,30,40,50
60
70
80
```

Eg:

```
s = '10,20,30,40,50,60,70,80'
l = s.split(',',-1)
for x in l:
    print(x)
```

Output:

```
10
20
30
40
50
60
70
80
```

Any question?

If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You