Python Programming

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

PYTHON LANGUAGE FUNDAMENTALS

Guido Van Rossum

Learning Mantra

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

Agenda:

1. Data types: str data type - Slice operator
2. Data types: + and * operators for str data type

Data types: str data type - Slice operator

Now, we are going to learn one very important Python specific special operator known as **Slice operator.**

What is Slice?

- Suppose, if you have an apple and if you cut it into multiple pieces. Each piece is called as a slice.
- Similarly, string slice means a part of the string. You will get the part of the string by using slice operator.

Data types: str data type - Slice operator

s = 'abcdefghijklmnopqrstuvwxyz‘

- Now, If we want to get which character is locating at specific index position, simply writing **s[index]** will automatically get that character.
- If we want to get slice or piece of the string, for example we want the piece of the string from index position 3 to index position 7 (i.e., total 5 characters). You can get this piece of the string by using slice operator.

Syntax of slice operator:

stringName [beginIndex:endIndex]

This operator returns the substring (slice) from **beginIndex** to **endIndex - 1**.

Data types: str data type - Slice operator

```
s = 'abcdefghijklmnopqrstuvwxyz'

slice = s[3:8]          # returns characters from 3 to 7 index

print(slice)            # defgh
```

Suppose, If you are not specifying the begin index, then the default value of the begin index is starting index of the string [i.e,. 0].

```
s = 'abcdefghijklmnopqrstuvwxyz'

slice = s[:8]           # returns characters from 0 to 7 index

print(slice)            # abcdefgh
```

Data types: str data type - Slice operator

- Suppose, If you are not specifying the end index, then the default value of the end index is ending index of the string [i.e,. -1].

```
s = 'abcdefghijklmnopqrstuvwxyz'

slice = s[3:]          # returns characters from 3 to last index

print(slice)           # defghijklmnopqrstuvwxyz
```

- Suppose, If we are not specifying begin index and end index. What happens?

```
s = 'abcdefghijklmnopqrstuvwxyz'

slice = s[ : ]          # returns all characters

print(slice)            # abcdefghijklmnopqrstuvwxyz
```

Data types: str data type - Slice operator

Another Example:

```
s = 'abcdefghijklmnopqrstuvwxyz'

slice = s[3:1000]          # returns characters from 3 to last index

print(slice)               # defghijklmnopqrstuvwxyz
```

- slice operator never goes to raise index error

```
s = 'abcdefghijklmnopqrstuvwxyz'

slice = s[ 5:1 ] #It starts from 5 and goes in forward direction and never gets 1

print(slice)     #Empty string will be displayed
```

Data types: str data type - Slice operator Applications

1. Convert the first letter of the string into uppercase letter.

```
s = 'karthi'
output = s[0].upper() + s[1:]
print(output)
```
➔Karthi

2. Convert the last letter of the string into uppercase letter.

```
s = 'karthi'
output = s[0:len(s)-1] + s[-1].upper()
print(output)
```
➔karthI

Data types: str data type - Slice operator Applications

3. Convert the first and last letter of the string into uppercase letter.

```
s = 'karthisahasra'

output = s[0].upper() + s[1:len(s)-1] + s[-1].upper()

print(output)          ➔ KarthisahasrA
```

Data types: + and * operators for str data type

- Related to strings there are two important points we want to discuss with respect to mathematical operations.

1. '+' operator for the string:

```
s = 'karthi' + 'sahasra'          # concatenation
print(s)                          ➔karthisahasra
```

s = 'karthi' + 10 **#** in Java, output is **karthi10,** but in python it gives **error**

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

Note:

- In python, if you are performing concatenation operation (i.e., '+' operation on strings), then both operands must be string type.

Data types: + and * operators for str data type

2. '*' operator [String repetition operator] for the string:

This speciality is not there in other programming languages.

Eg:

```
s = 'karthi' * 3    #string repetition operator
print(s)            # karthikarthikarthi
```

In python, if you are performing string repetition operation (i.e., '*' operation on strings), one operand should be an integer type and another one is string type.

Eg:

```
print('#' * 10)      ➔##########
print("karthi")      ➔ karthi
print('#' * 10)      ➔##########
```

Important Conclusions:

1. So far, we covered the following data types of Python:
    1. int
    2. float
    3. complex
    4. bool
    5. str

- These 5 data types are called as **fundamental data types of Python**.

2. long data type is available in Python-2, but not in Python-3. long values also you can represent by using int type in Python-3.

3. There is no char data type in Python, char values also you can represent by using str type.

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