Python Programming

EDUCATION FOR PEACE

(ESTD-1995)

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

PYTHON LANGUAGE FUNDAMENTALS-14

Guido Van Rossum

Learning Mantra

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

Agenda:

1. Python Data Types : range

2. How to create range object?

3. Python Data Types : bytes and byte array

Python Data Types : range

- **range** Data Type represents a sequence of numbers.
- **range()** is the in-built function of Python.
- The elements present in range Data type are not modifiable. i.e., range Data type is immutable.

Eg:

```
r = range(10)        # it represents the sequence of values from 0 to 9
print(type(r))       ➔<class 'range'>
print(r)             ➔range(0, 10)
```

How you can print the values present in the given range?

- We have to make use of loops, such as for, while etc., to display the elements in the given range.

Eg:

```
r = range(10)   # it represents the sequence of values from 0 to 9

print(type(r))   ➔<class 'range'>

print(r)         ➔range(0, 10)

for x in r:

    print(x,end=‘ ‘)      ➔0  1  2  3  4  5  6  7  8  9
```

How to create range object? what are the various options are available?

Option 1:

range(n) => It represents the sequence of numbers from 0 to n-1.

```
range(10)          # it represents the sequence of numbers from 0 to 9
range(100)         # it represents the sequence of numbers from 0 to 99
```

Option 2:

Sometimes, our requirement is, We don't want from 0,we want to print the numbers from any specific number.

```
r = range(1,11)
for x in r:
    print(x,end = ' ')    ➔1 2 3 4 5 6 7 8 9 10
```

How to create range object? what are the various options are available?

Option 3:

range(begin, end, increment/decrement value) ➔ It represents the sequence of numbers from begin to end by increment/decrement value.

Eg:

```
r = range(1,21,1)
for x in r:
    print(x, end = ' ')   ➔1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

We can access elements present in the range Data Type by using index.

Eg:

```
r = range(10,21)
print(r[0])        ➔10
print(r[-1])       ➔20
r1 = r[1:5]
print(r1)          ➔range(11, 15)
for x in r1:
    print(x,end=‘ ‘)      ➔11 12 13 14
```

- Once order is important, obviously indexing, slicing concepts are also applicable.
- range object is immutable.

Eg:

```
r = range(10,21)
print(r[0])          →10
print(r[-1])         →20
r1 = r[1:5]
print(r1)            →range(11, 15)
for x in r1:
    print(x,end=‘ ‘)      →11 12 13 14
r[1] = 100                →TypeError: 'range' object does not support item assignment
```

We can create a list of values with range data type.

Eg:

```
l = list(range(10))
print(l)
```

Output:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Important conclusions observed with respect to range data type

1. range Data type represents a sequence of numbers.

2. Different forms of range data type are as follows:

    - range with one argument - range(10)
    - range with Two arguments - range(10,21)
    - range with three arguments - range(10,21,2)

3. Once order is important, obviously indexing, slicing concepts are also applicable.

4. range object is immutable.

Python Data Types : bytes and byte array

1. bytes data type:

- ❑ It's not that much frequently used data type in Python.
- ❑ bytes data type represents a group of byte numbers just like an array.

Eg:

```
l = [10,20,30,40]
b = bytes(l)      # If you want to create bytes object, you have to use in-built function
                    'bytes'
print(type(b))        ➔<class 'bytes'>
```

Python Data Types : bytes and byte array

Now, we want to print all values present inside 'b'.

Eg:

```
l = [10,20,30,40]
b = bytes(l)
for x in b:
    print(x)
```

Output:

```
10
20
30
40
```

Python Data Types : bytes and byte array

Where byte data type is helpful?

- If you want to handle binary data, like images, video files and audio files, we need to make use of byte and bytearray data types.

Two important conclusions about bytes data type:

Conclusion 1:

- The only allowed values for byte data type are 0 to 255. By mistake if we are trying to provide any other values then we will get value error.

Eg:

l = [10,20,30,40,256]

b = bytes(l)

ValueError: bytes must be in range(0, 256)

Python Data Types : bytes and byte array

Conclusion 2:

- ❑ Once we creates bytes data type value, we cannot change its values, otherwise we will get TypeError. i.e.,Immutable.

Eg:

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

b = bytes(l)

print(b[0])        ➔10

b[0] = 100         ➔TypeError: 'bytes' object does not support item assignment
```

Python Data Types : bytes and byte array

2. bytearray data type:

- bytearray is exactly same as bytes data type except that its elements can be modified. i.e., **Mutable.**

Eg:

```
l = [10,20,30,40]
b = bytearray(l)
for i in b:
    print(i,end=‘ ‘)          ➔10  20  30  40
print(b[0])                   ➔10
b[0] = 188
print(b[0])                   ➔188
```

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