Review of Python

Lecture - 2

By : Sandip kumar

Any doubts?

Binary Number System!

How is data stored?

I. Whole Numbers / Positive Integers – Binary Equivalent. E.g: 23 10111

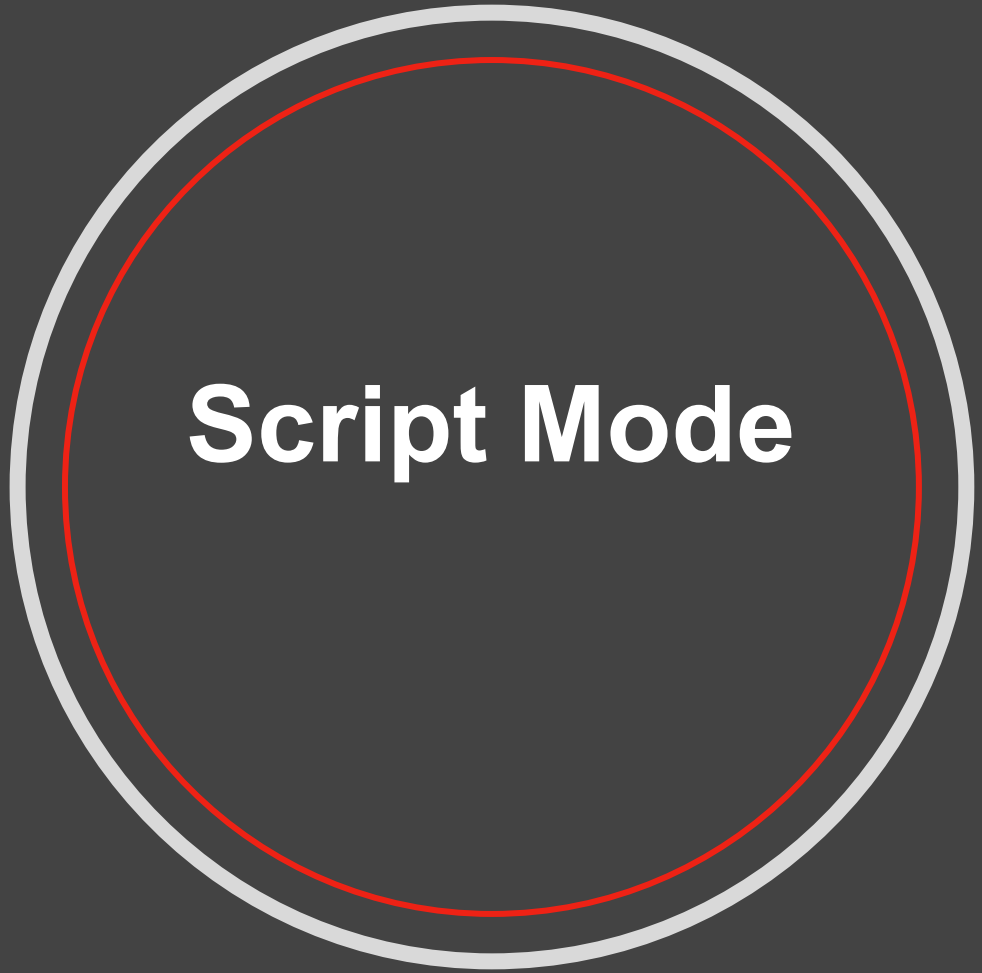
II. Characters – Binary Equivalent of their ASCII Values. E.g: ‘A’ 65 1000001

III. Negative Numbers – 2’s complement of their counter part

IV. Floating Values – Binary Equivalent of 2 integers [Significant and Exponent]. E.g: 1.23 can be represented as 123 * 10^-2. So we now have two integers 123 and -2.

Interactive Mode

Allows us to interact with OS. Hear, when we type Python statement, interpreter displays the result(s) immediately. That means, when we type Python expression / statement / command after the prompt (>>>), the Python immediately responses with the output of it.


Script Mode

In script mode, we type Python program in a file and then use interpreter to execute the content of the file. Working in interactive mode is convenient for beginners and for testing small pieces of code, as one can test them immediately. But for coding of more than few lines, we should always save our code so that it can be modified and reused.

Python Identifiers

Identifier is the name given to entities like class, functions, variables etc. in Python. It helps differentiating one entity from another.

Rules for writing identifiers

- Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore (_).
- An identifier cannot start with a digit.
- Keywords cannot be used as identifiers.
- We cannot use special symbols like !, @, #, $, % etc. in our identifier.
- Identifier can be of any length.

Python Keywords

- Keywords are the reserved words in Python.
- We cannot use a keyword as variable name , function name or any other identifier. They are used to define the syntax and structure of the Python language.
- keywords are case sensitive.
- There are 33 keywords in Python.
- All the keywords except True, False and None are in lowercase and they must be written as it is.

Keywords in Python

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Python Variables

- A variable is a location in memory used to store some data (value).
- They are given unique names to differentiate between different memory locations. The rules for writing a variable name is same as the rules for writing identifiers in python.
- We use objects (variables) to capture data, which then can be manipulated by computer to provide information. By now, we know that object/variable is a name which refers to a value. Every object has: Identity, Type, and Value.

Data types in Python

Every value in Python has a datatype. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

Standard Data Types

- Numbers
- String
- List
- Tuple
- Dictionary

Numbers

- Number data type stores Numerical Values.
- Range of an integer in Python can be from -2147483648 to 2147483647.
- Numbers are of three different types:
  Integer & Long (to store whole numbers i.e. decimal digits without fraction part)
  Float/floating point (to store numbers with fraction part)
  Complex (to store real and imaginary part)

```
>>> c= 4298114
>>> type(c)              # type ( ) is used to check data type of value
>>> c = c * 5669
>>> type(c)



>>> flag = True
>>> type(flag)



>>> x = 1+0j
>>> print x.real,x.imag
```

String

- String is an ordered sequence of letters/characters.
- They are enclosed in single quotes (‘ ’) or double (“ ”).

```
>>> a = 'Ram'
>>> type (“Good Morning“)
>>> type (“3.2“)
```

Lists

- List is also a sequence of values of any type.
- Values in the list are called elements / items.
- These are mutable and indexed/ordered.
- List is enclosed in square brackets.

w = [“spam‟ , 20.5 , 5]

Tuples

- Tuples are a sequence of values of any type, and are indexed by integers.
- They are immutable.
- Tuples are enclosed in ().

```
tuple = ( 'abcd', 786 , 2.23, 'john', 70.2  )
tinytuple = (123, 'john')
print tuple               # Prints complete list
print tuple[0]            # Prints first element of the list
print tuple[1:3]          # Prints elements starting from 2nd till 3rd
print tuple[2:]           # Prints elements starting from 3rd element
print tinytuple * 2       # Prints list two times
print tuple + tinytuple # Prints concatenated lists
```

What's the difference between lists and tuples?

- **List** is mutable.
- Lists are enclosed in brackets ([ ])
- their elements and size can be changed

- **Tuples** is immutable.
- Tuples are enclosed in parentheses (( ))
- cannot be updated.

Dictionaries

- Dictionaries is mutable.
- Dictionaries Can store any number of python objects.
- They store is a key – value pairs, which are accessed using key.
- Dictionary is enclosed in curly brackets.

```
tinydict = {'name': 'john','code':6734, 'dept': 'sales'}

print tinydict          # Prints complete dictionary
print tinydict.keys()   # Prints all the keys
print tinydict.values() # Prints all the values
```

Mutable and Immutable Variables

A mutable variable is one whose value may change in place, whereas in an immutable variable change of value will not happen in place.

Type conversion / type casting

- It is possible to change one type of value/ variable to another type. It is known as type conversion or type casting.
- The conversion can be done explicitly (programmer specifies the conversions) or implicitly (Interpreter automatically converts the data type).
- For explicit type casting, we use functions (constructors):
  int ( ) float () str () bool ()

```
>>> a= 12.34
>>> b= int(a)
>>> print b
```

Operators and Operands

Operators are special symbols that represent computation like addition and multiplication. The values that the operator is applied to are called operands.

Mathematical/Arithmetic operators: +, -, *, /, %, ** and //
Relational operators: <, <=, >, >=, != or <> and ==.
Logical operators: or, and, and not
Assignment Operator: =, +=, -=, *=, /=, %=, **= and //=

Thank You!

Sandip Kumar
sk1957339@gmail.com
+91 9873138842