PYTHON PROGRAMMING LANGUAGE

Unit-1

Introduction:

Before Learning & write computer program need to know the algorithm. It helps to write computer programs.

Algorithm: Algorithm is a step-by-step process of solving a well-defined computational problem. In practice, in order to solve any complex real life problems, first we have to define the problem and then, design algorithm to solve it. Generally, algorithms are used to simplify the program implementation. Also, defined step-by-step procedure to solve the problem is called algorithm.

Example 1:

Write an algorithm to how to Make "Maggi Noodles"?

Step 1: Start
Step 2: Take pan with water
Step 3: Put pan on the burner
Step 4: Switch on the
gas/burner Step 5: Put magi
and masala Step 6: Give two
minutes to boil Step 7: Take
off the pan
Step 8: Take out the magi with the help of fork/spoon
Step 9: Put the maggi on the plate and serve it
Step 10: Stop.

Example 2:

Write an algorithm to print "Good Morning".

Step 1: Start
Step 2: Print "Good
Morning" Step 3: Stop

Example 3:

Write an algorithm to find area of a rectangle.

```
Step 1: Start
Step 2: Take length and breadth and store them as L
and B? Step 3: Multiply by L and B and store it in area
Step 4: Print area
Step 5: Stop
```

Program: A program is a collection instructions that can do a particular task or problem. Programming is the process of taking an algorithm and writing it in a particular programming language, so that it can b executed by a computer. Programmers can use any of the programming languages to write programs.

Python Programming Introduction:

Introduction:

Python is a high-level, interpreted, interactive, scripting, object-oriented and a reliable language. Python was developed by Guido van Rossum in the late 80’s and early 90’s at the National Research Institute for Mathematics and Computer Science in the Netherlands. It has derived from many other languages such as ABC, C, C++, Small Talk, Unix Shell and other scripting languages. The first version of language was released in I991. Python has emerged as a very powerful and popular language.

Why the name Python? It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

Features of Python Programming

1) A **simple** language which is easier to learn Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#.

2. **Versatile** Python supports development of a wide range of applications ranging from simple text processing to WWW browser to games.

3. **Free and open-source** Python is an open source software. Therefore, anyone can freely distribute it, read the source code, edit it, and even use the code to write new programs.

4. **Portability** You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

5. **Extensible** and Embeddable Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

6. **A high-level**, interpreted language Unlike C/C++, you don't have to worry about difficult tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower- level operations.

7. **Large standard libraries** to solve common tasks Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself.

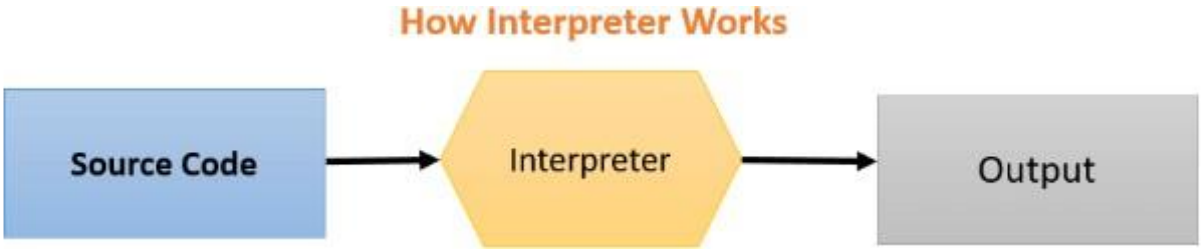
8. **Object-oriented** Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem very easily With OOP, you are able to divide these complex problems into smaller sets by creating objects.

Applications of Python

Python is a high level general purpose programming language that is used to develop a wide range of applications including image processing, text processing, web, and enterprise level applications using scientific and numeric data from network. Some of the key applications of python includes:

- In operations of Google search engine, youtube, etc.
- Bit Torrent peer to peer file sharing is written using Python
- Intel, Cisco, HP, IBM, etc use Python for hardware testing.
- Maya provides a Python scripting API
- i–Robot uses Python to develop commercial Robot.
- NASA and others use Python for their scientific programming task.
- Python is used as an Embedded scripting language.
- GUI-based Desktop applications
- Games: Python support development of games.
- Business applications
- Operating systems
- Python is used for Network programming.

Interpreter : An interpreter is a program that converts program written in high-level language(Source code) into machine code understood by the computer. Interpreter executes one statement at a time.



scripting language: A **scripting language** is a programming language designed for integrating and communicating with other programming languages. Some of the most widely used scripting languages are JavaScript, VBScript, PHP, Perl, Python, Ruby, ASP and Tcl. Since a scripting language is normally used in conjunction with another programming language, they are often found alongside HTML, Java or C++.

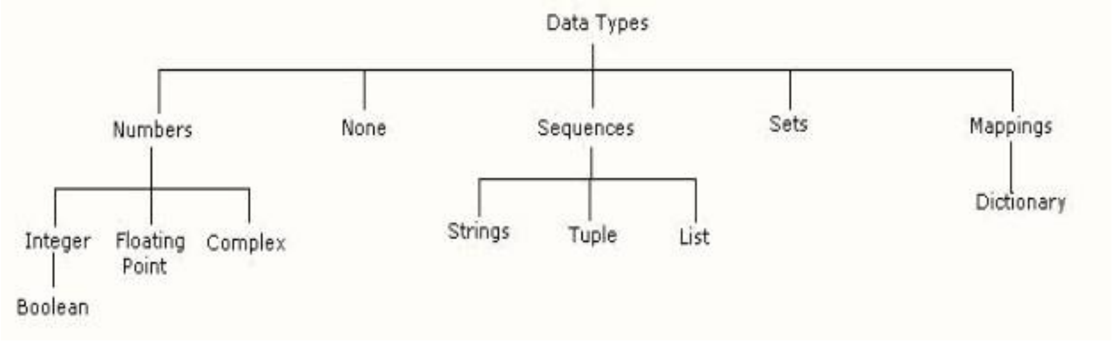
First Python Program:

Example: To print a message on the screen

```
>>> print ("Hello World!!!")
Hello World
```

Data Types:

Variables can hold values of different types called data types. The standard data types supported by Python includes -



1) Numbers
    a. Integers
    b. Floating Point Numbers
    c. Complex Numbers
2) None

3) Sequences
   a. Strings
   b. Tuple
   c. List
4) Sets
5) Mappings – Dictionary

1) **Numbers:** Number refers Numerical Values. This data type is immutable i.e. its value cannot be changed. These are of three different types:

a) Integers

b) Float/floating point

c) Complex

a) **Integers:** Numbers like 5 or other whole numbers are referred to as integers. Bigger whole numbers are called long integers. For example, 56958558585855L is a long integer. Note that a long integer must have 'l' or 'L' as the suffix.

Integers contain Boolean Type which is a unique data type, consisting of two constants, True & False. A Boolean True value is Non-Zero, Non-Null and Non-empty.

Example

```
>>> flag = True
>>> type(flag)
<type 'bool'>
```

b) **Floating point numbers:** Numbers with fractions or decimal point are called floating point numbers.

Example

```
y= 12.36
```

c) **Complex Numbers:** Numbers of **a+bj** form are complex numbers.

2) None: This is special data type with single value. It is used to signify the absence of value/false in a situation. It is represented by None.

3) Sequence:

A sequence is an ordered collection of items, indexed by positive integers. It is combination of mutable and non mutable data types. Three types of sequence data type available in Python are Strings, Lists & Tuples.

i) **Strings:** String is an ordered sequence of letters/characters. String is also defined as a group or set of characters. They are enclosed in single quotes (' ') or double (" "). Strings are immutable data types.

Example

```
>>> a = 'Ram'
```

ii) **List**: List is data type available in python. It is a sequence in which elements are written as a list of comma separated values between square brackets. List is mutable data type which means the value of its elements can be changed.

Syntax: list_variable=[var1,var2,.................]

Example

```
rgukt = *"spam", 20.5,5]
```

iii) **Tuples:** Tuples is data type available in python. Tuples are sequence of elements separated by comma enclosed in parenthesis. They are immutable data type.

Example

```
my_tuple = (4, 2, 5, 8)
```

4) **Sets:** Set is an unordered collection of values, of any type, with no duplicate entry. Sets are immutable.

Example

```
s = set ([1,2,34])
```

5) **Mapping**: This data type is unordered and mutable. A dictionary comes under Mappings. Dictionary: Dictionary is a data type available in python which store values as a pair of key and value. Each key is separated from its value by a colon (:), and consecutive items are separated by commas. The entire items in a dictionary are enclosed in curly brackets {}. Dictionary keys must be of immutable type.

Example: d = {1:'a',2:'b',3:'c'}

Mutable and Immutable Variables

Variable: Variables are reserved memory locations that stores values. Variables play a very important role in most programming languages, and Python is no exception. You can store any piece of information in a variable. Variables are nothing but just parts of your computer’s memory where information is stored. To be identified easily, each variable is given an appropriate name. Every variable is assigned a name which can be used to refer to the value in the program.

A **mutable variable** is one whose value may change , whereas in an **immutable** variable the value cannot be change.

Example: Initializing values to
variables val= “hello”

```
num=75
amt=12.65
print (val)
print (amt)
print (num)
```

Output:

```
“hello”
75
12.65
```

Multiple assignments

Python allows you to assign a single value to several variables
simultaneously. For example: a = b = c = 1

Here, an integer object is created with the value 1, and all three variables are assigned to the same memory location.

You can also assign multiple objects to multiple variables. For example – Example: a,b,c = 1,2,"john"

Here, two integer objects with values 1 and 2 are assigned to variables a and b respectively, and one string object with the value "john" is assigned to the variable

Expression and Statements

An expression is a combination of values, variable and operators that represents a value.

Example:

sum=a+b   (Expression)

a and b operands and + is operator.

Operators are the constructs that are used to manipulate the value of operands. Some basic operators include +, -, *, /.

Operands are the values.

A Python **statement** is a unit of code that the Python interpreter can execute.

Example of statement are:

```
>>> x=5
>>> area=x**2 #assignment statement
>>>print x #print statement 5
>>>print
area 25
>>> print x,
area 5 25
```

Value:

In mathematics and computer programming, a **value** is data element that may be stored in a variable. The value can be numeric (such as an integer, or a floating point number), or alphanumeric (such as a character, or a string).

Example: 1, 2.4, “Hello”

Python Identifiers:

Identifier is the name given to entities like class, functions, variables etc. For naming identifier, there are some basic rules that you must follow. These rules are

- 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 (_).
- Keywords cannot be used as identifiers.
- An identifier cannot start with a digit. 1variable is invalid, but variable1 is valid.
- Identifier names are case sensitive. For example myvar and myVar are not the same.
- Punctuation characters such @,$,£ and % are not allowed within identifiers.
- Does not include space character in the identifier name

Example:

Valid identifier names are: sum, _my_var, num1, r, var_33, first

etc Invalid identifier names are: 1num, my-var, %check, basic sal

etc

Comments

Comments are the non-executable statements in a program. They are just added to describe the code of the program. Comments make the program easily readable and understandable by the programmer as well as other users who are seeing the code. The interpreter simply ignores the comments.

In python, a has sign (#) symbol writing as a comment.

Example

```
#This is a comment
#print out Hello
print('Hello')
#program ends
here
```

Multi-line comments

If we have comments that extend multiple lines, We can use triple quotes, either ''' or """.

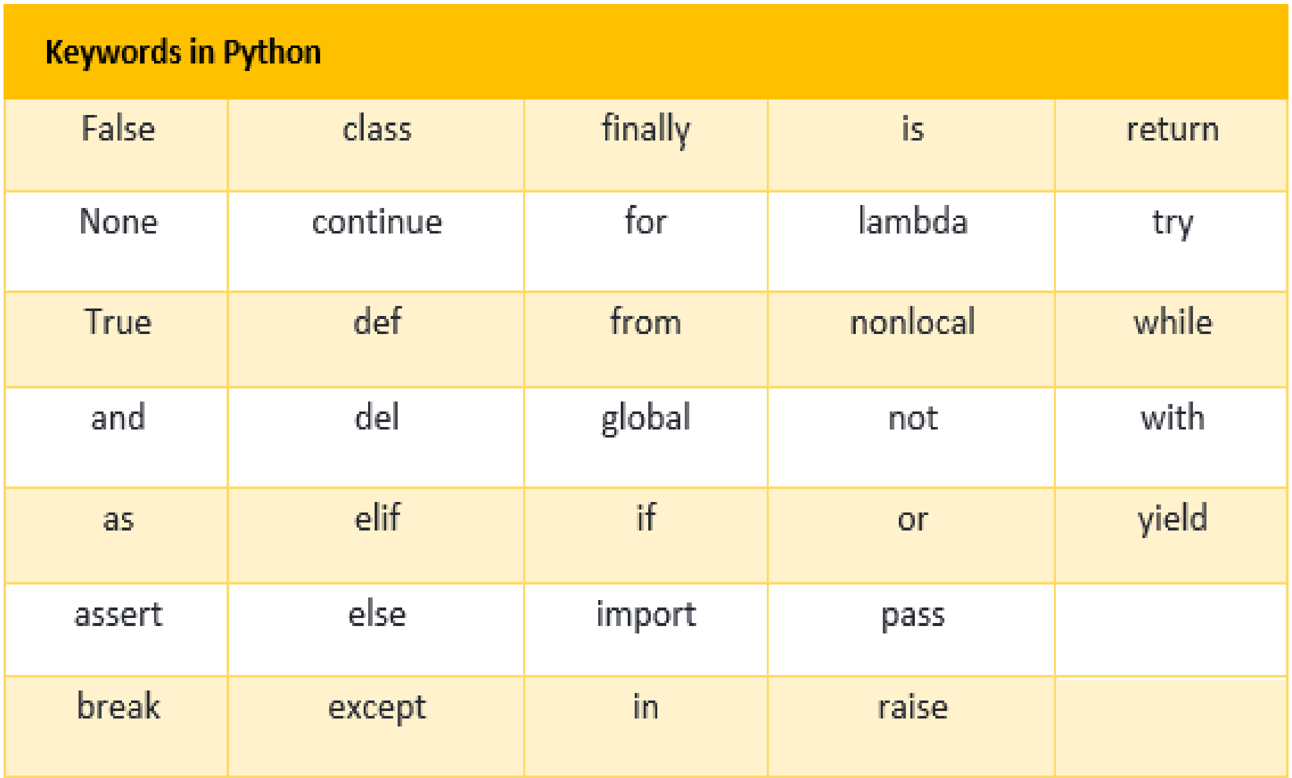
Example

```
 """This is also a
      perfect example
       of multi-line comments"""
```

Note: A program can have any number of comments.

Keywords in Python

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. In Python, 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. The list of all the keywords are given below.



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	

Indentation:

White space at the beginning of the line is called ***indentation.*** These whitespaces or the indentation are very important in python. Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation

Leading white space (spaces and taps) at the beginning of each statement, which is used to determine the group of statement, is known as “indentation”.

Example

```
If  A > B :
        print “A is Big”      # Block1
else:
        print “B is Big”     # Block2
```

In the above example, if statements are a type of code block. If the “if” expression evaluates to true, then Block1 is executed, otherwise, it executes Block2. Obviously, blocks can have multiple lines. As long as they are all indented with the same amount of spaces, they constitute one block.

Assignment-I

1. Define datatype. What are the datatypes available in Python? Explain in detail.
2. What is an Identifier? List out the rules of an identifier.
3. What is a Python? Write notes on Python history, features and Applications.
4. What is indentation? Explain with an example.
5. Define the following terms.

   i) Algorithm ii) Interpreter iii) variable iv) Expression v) Value vi) comment