Unit 1 : Introduction to Python

By : Bhavika Vaghela

Asst. Prof. Bhavika Vaghela

PICA - PU

What is Python?

- **Python** is a general purpose, dynamic, high-level, and interpreted programming language.
- It supports Object Oriented programming approach to develop applications.
- Python is a cross-platform programming language, which means that it can run on multiple platforms like Windows, macOS, Linux, and has even been ported to the Java and .NET virtual machines.
- It is free and open-source.
- Python is not intended to work in a particular area, such as web programming.
- That is why it is known as **multipurpose** programming language because it can be used with web, enterprise, 3D CAD, etc.

- We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.
- Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.
- Python is also incredibly useful at integration of tasks. Websites like YouTube, Quora, Flipkart, Slack, Uber, Cloudera, Instagram, Zenefits and Spotify are created using Python.

Python 2 VS Python 3

- Python 2 uses **print** as a statement and used as print "something" to print some string on the console. On the other hand, Python 3 uses **print** as a function.
- Python 2 uses the function raw_input() to accept the user's input. Python 3 uses input() function which automatically interpreted the type of input entered by the user. However, we can cast this value to any type by using primitive functions (int(), str(), etc.).
- In Python 2, the implicit string type is ASCII, whereas, in Python 3, the implicit string type is Unicode.
- Python 3 doesn't contain the xrange() function of Python 2. The xrange() is the variant of range() function which returns a xrange object that works similar to Java iterator. The range() returns a list for example the function range(0,3) contains 0, 1, 2.

- There is also a small change made in Exception handling in Python 3. It defines a keyword **as** which is necessary to be used.

VERSIONS RELEASED and History of Python

- Python2.1 was released on April 17, 2001.
- Python2.2 was released on December 21, 2001.
- Python2.3 was released on July 29, 2003.
- Python2.4 was released on November 30, 2004.
- Python2.5 was released on September 19, 2006.
- Python2.6 was released on October 1, 2008.
- Python 2.7 was released on July 3, 2010.
- Python 3.9.2 is the latest version released on February 19, 2021

https://www.python.org/downloads/

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.
- In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
- In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
- Python 2.0 added new features like: list comprehensions, garbage collection system.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language.
- *ABC programming language* is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
- Python is influenced by following programming languages:
  - ABC language.
  - Modula-3

Features or advantages of python

- Python is easy to learn and use. It is developer-friendly and high level programming language.
- Python language is more expressive means that it is more understandable and readable.
- is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.
- Cross-platform Language, Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.
- Free and Open Source Python language is freely available at offical web address. The source-code is also available.
- Object-Oriented Language, Python supports object oriented language and concepts of classes and objects come into existence.

- Large Standard Library Python has a large and broad library and prvides rich set of module and functions for rapid application development.
- GUI Programming Support Graphical user interfaces can be developed using Python.
- Integrated It can be easily integrated with languages like C, C++, JAVA etc.

What is Program?

- A program is a sequence of instructions that specifies how to perform a computation.
- The computation might be something mathematical, such as solving a system of equations or finding the roots of a polynomial etc.

What is debugging?

- Programming errors are called **bugs** and the process of tracking them down is called **debugging**.

Three types of Error

- **Syntax Error** : it occur when there is problem in programming syntax.
- Run Time Error : A **runtime error** is a program **error** that occurs while the program is running.
- Semantic Error : Writing invalid program logic that produces incorrect results when the instructions are executed. The syntax of the source code may be valid, but the algorithm being employed is not.

** error is also known as exception

Value and type

- A **value** is one of the basic things a program works with, like a letter or a number.
- The values we have seen so far are 1, 2, and 'Hello, World!'.
- These values belong to different **types**: 2 is an integer, and 'Hello, World!' is a **string**, 0.4 is a float etc.
- If you are not sure what type a value has, the interpreter can tell you the type of value using **type()** method.
- When you type a large integer, you might be tempted to use commas between groups of three digits, as in 1,000,000. This is not a legal integer in Python

```
a=1,000,000
>>> print(a)
(1, 0, 0)
** very first example of semantic error
```

Example to check datatype and long value with python variable

```
In [3]: value = 100
        type(value)
Out[3]: int

In [4]: mes = "hello"
        type(mes)
Out[4]: str

In [5]: value = 10.567
        type(value)
Out[5]: float

In [6]: mes = 'a'
        type(mes)
Out[6]: str

In [7]: value = '10.30'
        type(value)
Out[7]: str
```

```
>>> print(num)
(1, 0, 0)
>>> num = 12,567,234
>>> print(num)
(12, 567, 234)
>>> num = 1,234,567
>>> num
(1, 234, 567)
>>>
```

Variable

- Variable is a name which is used to refer memory location. Variable also known as identifier and used to hold value.
- A variable is a name that refers to a value.
- An **assignment statement** creates new variables and gives them values.
- If you type an integer with a leading zero, you might get a confusing error:

```
>>> zipcode = 02492
```

SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers

```
In [8]: num = 092156
        print(num)

          File "<ipython-input-8-9db9b265ae22>", line 1
            num = 092156
                  ^
        SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers
```

Variable names and keywords

- Variable names can be a group of both letters and digits, but they have to begin with a letter or an underscore.
- The underscore character (_) can appear in a name. It is often used in names with multiple words, such as my_name or airspeed_of_unladen_swallow.
- Identifier name must not contain any white-space, or special character (!, @, #, %, ^, &, *).
- Identifier name must not be similar to any keyword defined in the language.
- Identifier names are case sensitive for example my name, and MyName is not the same.
- Declare variable name using reserved keyword is not allow.

e.g _name_, _myname123, my_name, hello123, _123name,

Reserved Keyword in Python

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

Statements

- A statement is a unit of code that the Python interpreter can execute. We have seen two kinds of statements: print and assignment.
- When you type a statement in interactive mode, the interpreter executes it and displays the result, if there is one.
- A script usually contains a sequence of statements. If there is more than one statement, the results appear one at a time as the statements execute.

Expressions

- An **expression** is a combination of values, variables, and operators.
- **E.g. x + 17, (1+1)**(5-2), (minute * 100) / 60**

Difference between Statement and Expression

Expression	Statement
Which evaluates the value or code and gives output	Is complete line of code which perform something
Combine horizontally using operator and operands	Combine vertically (any length)
Expression can be use as statement	Statement can not be use as expression
Output of expression always comes in single value	Its not true for statement
Example of expression : **2+2, max(10,67), min(10,4),** **"wel"+"come" etc..**	**Example of Statement :** **Number1=10, number1>=7,** **for I in range(10), break, etc..**

Order of operations

- When more than one operator appears in an expression, the order of evaluation depends on the **rules of precedence**
- **PEMDAS** is a useful way to remember the rules.
- **P**arentheses have the highest precedence.
- **E**xponentiation has the next highest precedence, so 2**1+1 is 3, not 4.
- **M**ultiplication and **D**ivision have the same precedence, which is higher than **A**ddition and **S**ubtraction, which also have the same precedence. So 2*3-1 is 5, not 4, and 6+4/2 is 8, not 5.
- Operators with the same precedence are evaluated from left to right. So in the expression degrees / 2 * pi, the division happens first and the result is multiplied by pi.
- **Only assignment operator execute right to left.**

```
>>> print(num1)
18187
>>> Elif=100
>>> elif
SyntaxError: invalid syntax
>>> __myname__="bhavika"
>>> print("hi")
hi
>>> print(5-3+7)
9
>>> print(5%3**2)
5
>>> print(5%3)
2
>>> print(5%9)
5
>>> print(5%2-10)
-9
>>> print(5%2/3)
0.3333333333333333
>>> print(1/3)
0.3333333333333333
>>> print(3*2**3)
24
>>> print(10/5*3)
6.0
>>> print(3*5/10)
1.5
>>>
```

Comment

- Information in a program that is meant for other programmers (or anyone reading the source code) and has no effect on the execution of the program.
- # is use for single line comment and ‘’’ ‘’’ or “”” “”” is use for multiline comment.

Same value to multiple variable.

num1=num2= num3 = 12 //12 assign to all three variable

Different value to multiple variable.

Num1, num2, num3 = 10,20,30

**Please example on next slide

```
In [1]: #assign same value to multiple variable
        n1=n2=v1 = 671
        print("value of v1 : ",v1)
        print("value of n1 : ",n1)
        print("value of n2 : ",n2)

        value of v1 :  671
        value of n1 :  671
        value of n2 :  671

In [2]: #assign different value to multiple variable
        n1,n2,v1 = 10,23.78,'hello'
        print("value of v1 : ",v1)
        print("value of n1 : ",n1)
        print("value of n2 : ",n2)

        value of v1 :  hello
        value of n1 :  10
        value of n2 :  23.78
```