

# Introduction to Python for Education

Smita Shashikant Patil (B.sc.Comp & MCA)

Asst. Teacher, Computer Science, Deccan Education society's NES & Jr.College Ramanbaug , Pune, India

**Abstract** – Python is a general purpose and high level programming language. It language use for developing desktop.GUI application websites and web applications. It also use to focus on core functionality of the application by taking care of common programming tasks. It is readable and maintainable. It compatible with major platforms and systems. It have robust standard library and have it open source frameworks and tools.

**Keywords** – Python is the winner in easy learning, cross platforms development the convenience of open supply libraries python is more dynamic language.

## I. INTRODUCTION

In this paper we are going to introduce the characteristics of python. It's structure of program, Data type, Control statement, Python IDE and code editors. Python interpreters are available for many operating system and its features.

### Guidelines

**Best python IDE's and code editors:-**IDE is development environment which provides many features like coding, compiling , debugging ,executing, auto complete, libraries .in one place for developers thus making tasks simpler where as code editors is a platforms. IDE:-Pycharm, Spyder, PyDev, Idle, Wing....

### Features of python programming

Python is a very high level, dynamic, object oriented, general purpose programming language that uses interpreter and can be used in a vast domain of applications. Python is very flexible, because of its ability to use modules that were designed in other programming languages. Features of Python are:

#### 1. Python is fast and powerful:-

It includes an internal standard library that provides all facilities that are needed for programming from the basic operations to the advanced functions.

#### 2. Python is simple and lovely:-

Readability was the main thought out point when designing the syntax of Python language. The language is very simple for a beginner to understand as compared with other languages C, C++.

In C and C++ the learners have to be very careful of the syntax. In Python the learner need to take care of only the indentation. The readability achieved as the result of indentation has number of benefits. The capacity of Python to balance high level programming with low level optimization is the strongest point of Python. One more likable point of Python from coder's point of view is the amount of code you write to solve a particular program.

Let us for example write a code which reads 2 files and writes those files one after the other to the 3rd file.

#### 3. Python is portable:-

python scripts can be used on different operating systems such as windows,Linux,UNIX,Amigo,mac OS etc.

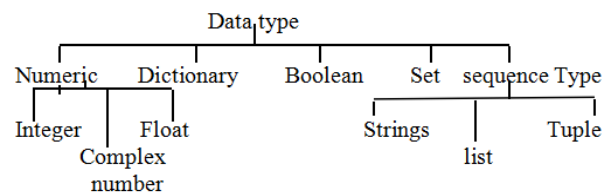
#### 4. Python is open source:-

Even though all rights of this program are reserved for the python institute,but it is open source and there is no limitation in using changing and distributing.

#### 5. Python supports other technologies:-

It can supports COM,Net etc.

**Data type of python:-** Text type:-str, Numeric type:-int, float, complex, Sequence type:-list ,tuple, range(strings),Mapping type:-dict,Set type:-set ,frozenset,**Boolean**:-bool,**BinaryTypes**:- bytes,bytearray,memory view.



Structure of Python:- Program of given number is prime or not

#taking input from user

```
Number=int(input("Enter any no:"))
```

```
If number>1:
```

```
    For i in range (2,number):
```

```
        If (number % i)==0:
```

```
            Print(number," is not a prime number")
```

```
        break
```

```
    else:
```

```
        print(number, "is a prime number")
```

```
else:
```

```
    print(number, "is not a prime no")
```

### Python Control Statements :

Python provides us with 3 types of Control Statements:

Continue  
Break  
Pass

### 1. Continue Statement:

When the program encounters continue statement, it will skip the statements which are present after the continue statement inside the loop and proceed with the next iterations.

Syntax: continue

Example:

```
for char in 'Python':  
    if (char == 'y'):  
        continue  
    print("Current character: ", char)
```

Output:-Current character: P

Current character: t

Current character: h

Current character: o

Current character: n

In the above example during the second iteration if the condition evaluates to true, then it will execute the continue statement. So whatever statements are present below, for loop will be skipped, hence letter 'y' is not printed.

### 2. Break Statement:

The break statement is used to terminate the loop containing it, the control of the program will come out of that loop.

Syntax: break

Example:

```
for char in 'Python':  
    if (char == 'h'):  
        break  
    print("Current character: ", char)
```

Output:

Current character: P

Current character: y

Current character: t

### 3. Pass Statement:

Pass statement in python is a null operation, which is used when the statement is required syntactically.

Syntax: pass

Example:

```
for char in 'Python':  
    if (char == 'h'):  
        pass  
    print("Current character: ", char)
```

Output:

Current character: P

Current character: y

Current character: t

Current character: h

Current character: o

Current character: n

Application of Python Programming:-Python is a best choice for real world programming as it is a very well

designed language. The most common program types that can be written by Python are as below:

### 1. System Programming:

Python provides internal interfaces for working with services of operating system due to which it is suitable for system programming. These interfaces provide some functions such as: files and directories operations, parallel processing etc. It contains tools for working with system resources such as: environmental variables, files, sockets, pipe, processes, multiple threads, command line, standard stream interfaces, shell programming etc

### 2. Graphical User Interface:

Tkinter & GxPython are standard object oriented interfaces that provides essential tools for designing GUI. Generated GUIs give a similar look on different platforms.

### 3. Network and Internet Programming:

Many tools are provided for network programmers namely client server connection, socket programming, FTP, Telnet, email functions, RPC, SOAP etc. Also some third party tools like mod-Python allow servers like apache to run python scripts.

### 4. Component integrity:

Tools like Swing and SIP can import the compiled codes of other languages for using in Python.

### 5. Database Programming: Python and Database:-

Python supports various databases like MySQL, oracle, Sybase, postgresQL. For database programming the python DB API is a widely used module that provides. A database application programming interface. mongoDB it is a NoSQL database management system.

How to link database:-

- Import  
import MySQLdb

Establish a connection with the database

Command:->

```
Conn=MySQLdb.connect(host='localhost',user='root',password='')
```

### 6. Numerical programming:

Python with NumPy module provides powerful tools for working with mathematical libraries, by using simple Python codes.

### 7. Other Programming applications: PyGame is a tool for Game Programming

- PIL is used for image processing

- PyRo for Robotic Programming
- NLTK is a package for artificial intelligence, network simulation and shell programming.
- Python is a free open source program and consequently it causes more popularity among users.

C	C++	Python
C was developed by Dennis Ritchie between the year 1969 and 1973 at AT&T Bell Labs.	C++ was developed by Bjarne Stroustrup in 1979.	Python was created by Guido van Rossum, and released in 1991.
More difficult to write code in contrast to both Python and C++ due to complex syntax.	C++ code is less complex than C but more complex in contrast to python.	Easier to write code.
Longer lines of code as compared to python.	Longer lines of code as compared to python.	3-5 times shorter than equivalent C/C++ programs.
Variables are declared in C	Variables are declared in C++	Python has no declaration.
C is a compiled language.	C++ is a compiled language.	Python is an interpreted language.
C contains 32 keywords.	C++ contains 52 keywords.	Python contains 33 keywords.
For the development of code, C supports procedural programming.	C++ is known as hybrid language because C++ supports both procedural and object oriented programming paradigms.	Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
C does not support inheritance.	C++ support both single and multiple inheritance	Python supports all 5 types of inheritance i.e. single inheritance, multiple inheritance, multilevel inheritance, hierarchical inheritance, and hybrid inheritance
C provides malloc() and calloc() functions for dynamic memory allocation, and free() for memory de-allocation.	C++ provides new operator for memory allocation and delete operator for memory de-allocation.	Python's memory allocation and deallocation method is automatic.
Direct support for exception handling is not supported by C.	Exception handling is supported by C++.	Exception handling is supported by Python.
<b>program in c to find leap year:-</b> #include <stdio.h> int main() { int year; printf("Enter a year: "); scanf("%d", &year);  // leap year if perfectly visible by 400 if (year % 400 ==	<b>program in c++ to find leap year:-</b> #include <iostream> using namespace std;  int main() { int year;  cout << "Enter a year:	<b># Python program to check if year is a leap year or not</b> year = 2000 # To get year (integer input) from the user # year = int(input("Enter a year: "))  if (year % 4) == 0:

0) { printf("%d is a leap year.", year); } // not a leap year if visible by 100 // but not divisible by 400 else if (year % 100 == 0) { printf("%d is not a leap year.", year); } // leap year if not divisible by 100 // but divisible by 4 else if (year % 4 == 0) { printf("%d is a leap year.", year); } // all other years are not leap year else { printf("%d is not a leap year.", year); }  return 0; }	}; cin >> year;  if (year % 4 == 0) { if (year % 100 == 0) { if (year % 400 == 0) cout << year << " is a leap year."; else cout << year << " is not a leap year."; } else cout << year << " is not a leap year."; }  return 0; }	if (year % 100) == 0: if (year % 400) == 0: print("{0} is a leap year".format(year)) else: print("{0} is not a leap year".format(year)) else: print("{0} is a leap year".format(year)) else: print("{0} is not a leap year".format(year))
--	---	---

**Used of Python:-** Python is a general purpose and high level programming language. You can use Python for developing desktop GUI applications websites and web applications.

## II. CONCLUSION

We have seen various features of Python Programming. We have seen its applications and a list of corporations using Python tools. It is a very vast language and has got various facets which can be researched upon. Few of the topics in which Python's application can be researched are Information Security, Artificial Intelligence, and Big Data Analytics etc.

## REFERENCES

- [1]. <http://www.pythontutor.com/>, <http://docs.python.org>
- [2]. <https://www.softwaretestinghelp.com/python>
- [3]. Sanders ID and Langford Students of Python, Fanhoer H A comparison of C

- [4]. MATLAB and python as teaching languages in engineering.
- [5]. Python 2.7 Release Schedule" . python.org . Retrieved 9 January 2017 "PEP 466,Network Security Enhancements for Python 2.7.x". python.org. Retrieved 9 January 2017.
- [6]. G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz
- [7]. Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529 551, April 1955. (references) J. Clerk Maxwell,
- [8]. A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.6873.
- [9]. Python for Education: Permutations Andrzej Kapanowski Marian Smoluchowski Institute of Physics Jagiellonian University, Cracow, Poland
- [10]. "History and License" Retrieved 5 December 2016 "All Python releases are Open Source"Venners, Bill (13 January 2003). "The Making of Python". Artima Developer. Artima. Retrieved 22 March 2007
- [11]. van Rossum, Guido (20 January 2009). "A Brief Timeline of Python"
- [12]. The History of Python. Google. Retrieved 20 January 2009."PEP 373