

1.1. Introduction

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, ALGOL-68, SmallTalk, and Unix shell and other scripting languages.

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python features

Python's feature highlights include:

- **Easy-to-learn:** Python has relatively few keywords, simple structure and a clearly defined syntax. This allows the student to pick up the language in a relatively short period of time.
- **Easy-to-maintain:** Python's success is that its source code is fairly easy-to-maintain.
- **A broad standard library:** One of Python's greatest strengths is the bulk of the library is very portable and cross-platform compatible on UNIX, Windows and Macintosh.
- **Interactive Mode:** Support for an interactive mode in which you can enter results from a terminal right to the language, allowing interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Expandable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh and the X Window system of Unix.
- **Scalable:**
 - Python provides a better structure and support for large programs than shell scripting.
 - Apart from the above-mentioned features, Python has a big list of good features, few are listed below:
 - Support for functional and structured programming methods as well as OOP.
 - It can be used as a scripting language or can be compiled to byte-code for building large applications.
 - Very high-level dynamic data types and supports dynamic type checking.
 - Supports automatic garbage collection.
 - It can be easily integrated with C, C++, ActiveX, CORBA and Java.

Available online interpreters

Refer these links for online interpreters to execute python code,

Link 1: <http://codepad.org/> (<http://codepad.org/>)

Link 2: <http://www.pythontutor.com/visualize.html#mode=edit>
(<http://www.pythontutor.com/visualize.html#mode=edit>)

Link 3: <http://www.codeskulptor.org/> (<http://www.codeskulptor.org/>)

A sample screen shot from pythontutor.com explaining execution of the program.

Command line Interpreter

Python has command line interpreter that helps in executing python commands by directly entering into python without writing a script.

Example:

If we type 1+2 the output would be 3.

The command line starts with >>> symbol as shown in examples,

```
>>> 2 + 2
```

```
4
```

```
>>> 50 - 5*6
```

```
20
```

```
>>> (50 - 5*6) / 4.0
```

```
5.0
```

```
>>> 8 / 5 # division always returns a floating point number
```

```
1.6
```

```
>>> width = 20
```

```
>>> height = 5 * 9
```

```
>>> width * height
```

```
900
```

Python on desktop

Refer to this link for downloading Python interpreter on your desktop.

<https://store.enthought.com/downloads/> (<https://store.enthought.com/downloads/>)