

## History

Python was developed in the late 1980's by Guido van Rossum. The first release was version 0.90 in February, 1991, at the National Research Institute for Mathematics and Computer Science in the Netherlands. The Institute is more popularly known by the Dutch name and initials Centrum Wiskunde and Informatica–CWI.<sup>1</sup>

It is derived from many languages like ABC, C, Unix shell and other scripting languages. But its primary influence in the early stages was ABC and the desire to design an easy to learn, first language for learning programming.

Since 2001, the Python Software Foundation (PSF) a non-profit organization, created specifically to own Python-related Intellectual Property, directs the development, evangelising and directions of the language.

Guido van Rossum continues to be the prime mover and is fondly referred to as BDFL–Benevolent Dictator For Life.

## What is Python

Python is a high-level, interpreted, interactive and object-oriented language. Its hallmark is an elegant syntax that enables writing very easy to read programs.

## Notable features

- Uses an elegant syntax, making the programs you write easier to read.
- Makes it easy to get your program working – ideal for prototype development and other ad-hoc programming tasks, without compromising maintainability.
- Comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, reading and modifying files. Often referred to as *batteries included* feature of Python.
- Runs on many different computers and operating systems: Windows, MacOS, many brands of Unix, OS/2, ...
- Has excellent Unicode support.

---

<sup>1</sup>CWI is famous for its association with Adriaan van Wijngaarden and Edsger Dijkstra.

# Introduction to Python

---

## Language characteristics

- All basic data types are available: numbers (floating point, complex, and unlimited-length integers), strings.
- Higher level containers such as lists, dictionaries and sets are part of the core language.
- Python supports object-oriented programming with classes and multiple inheritance.
- Modules and packages are the mechanisms to design, build, and distribute applications and libraries.
- Exception handling is available.
- Data types are strongly and dynamically typed. Mixing incompatible types (e.g. attempting to add a string and a number) causes an exception to be raised, so errors are caught sooner.
- Provides advanced programming features such as generators and list comprehensions.
- Automatic memory management frees you from having to manually allocate and free memory in your code.

## Application Areas

Python has now become a widely used professional language; it is used by organizations such as Google, NASA, Industrial Light and Magic, Cerenova, ABN Amro Bank ...

The following is a partial list of tools, frameworks and applications developed in python.

**Web Development** Python offers many choices for web development:

- Full-stack frameworks such as Django, Pyramid, and Zope.
- Micro-frameworks such as Flask and Bottle.
- Advanced content management systems such as Plone.
- Python's standard library supports many Internet protocols:
  - HTML, XML, JSON.
  - E-mail processing.
  - FTP, IMAP, sockets ...

## Introduction to Python

---

- Requests, a powerful HTTP client library.
- BeautifulSoup, a 'fault-tolerant' HTML parser.
- Feedparser for parsing RSS/Atom feeds.
- Paramiko implements the SSH2 protocol.
- Twisted Python, for asynchronous network programming.

**Scientific Computation** Python is widely used in scientific and engineering computing. In fact it has become the de-facto standard toolkit for such work.

- SciPy is **the** collection packages for mathematics, science, and engineering.
- Pandas is a data analysis and modeling library.
- IPython is currently leading the effort at providing an interactive computation and exploration platform.

**Education** Python is ideal for teaching programming, both at the introductory level and in more advanced courses. The most famous of the MIT's courses, 6.00x, switched to using Python a few years back.

**GUI Toolkits** Python has wide variety of graphical interface libraries.

- The Tk GUI library is included with Python.
- wxWidgets is a powerful big multiplatform gui toolkit
- Kivy, for writing multitouch applications, on say Android
- Qt can be used via pyqt or pyside libraries

## Versions and Implementations

The canonical reference implementation of the python language is what is mostly used. This is written in C and is the one originally developed by Guido van Rossum. When required to distinguish this from other implementations the term *CPython* is used to refer to it.

One of the earliest implementations is called Jython—as you can guess it is implemented in Java and is quite popular among companies that are predominantly Java based, but like to use Python in appropriate areas.

On the .NET platform there is IronPython.

# Introduction to Python

---

Pypy is an interesting attempt to implement Python in Python! It provides an optimising JIT compiler.

There are two major versions of Python currently in use. 2.7 and 3.3. Python 3 is backward incompatible effort to clean up some of the errors and inconsistencies in the original implementation.

We will use Python 3.3 in our course.

## Tool Set

There are many ways to create Python programs:

### 1. Interactive Interpreter

You can start the interpreter just by giving the name `python3`<sup>2</sup>

Once Python has started, you will see the interpreter startup message indicating version and platform and be given the interpreter prompt "`>>>`" to enter commands.

```
$ python3
Python 3.3.4 (default, Feb  9 2014, 12:04:33)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Figure 1: Interpreter - Linux

### 2. Writing Script on file

- Follow these steps to write and execute a script.
- At the (\$) prompt create a file using vim (or any text editor),  
\$ vim filename.py.
  - Write code in the file save it.
  - Execute - come back to terminal and type \$ python filename.py it generates the output in the console

As you can gather python programs have the extension .py

---

<sup>2</sup>In most Linux distributions, Python 2.x is by default installed with the executable named `python`. So python 3.x executable is installed as `python3`. This expected to change over time. In some distributions, such as ArchLinux, `python3` is the default and you have to run `python2` by typing `python2`.

## Introduction to Python

---

**3. Integrated Development Environment** We can also use graphical Interactive Development Environment - IDE to enter, and run python programs. The default ide in Linux is called *IDLE*

IDLE is the basic editor and interpreter environment with standard distribution of python. It looks like,

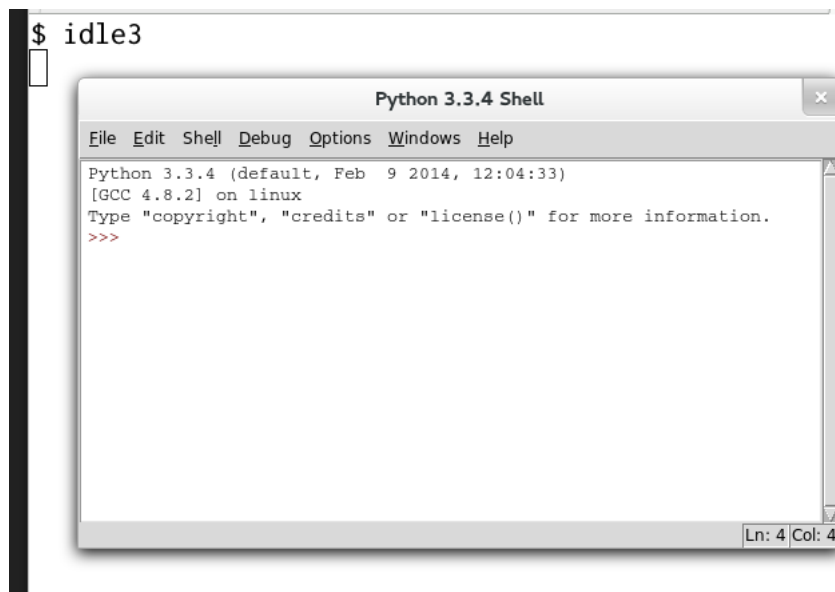


Figure 2: IDLE

**4. Using iPython** Our preferred interactive environment is called iPython. It is invoked by typing `ipython3` at the \$ prompt.

The reason we prefer iPython is that it provides much more help interactively.

### Sample Program

Now let us write our first Python program. The program will display a message:  
*Welcome to the World of Python*

Let us open the terminal and run the python interpreter: that is type `python3` at the \$prompt. At the `>>>` prompt type the line (and hit ENTER).

```
>>> print("Welcome to the World Of Python")
```

It produces the following result:

## Introduction to Python

```
$ ipython3
Python 3.3.4 (default, Feb  9 2014, 12:04:33)
Type "copyright", "credits" or "license" for more information.

IPython 1.2.0-rc1 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: █
```

Figure 3: iPython

Welcome to the World of Python

Now let us write the same program in a script.

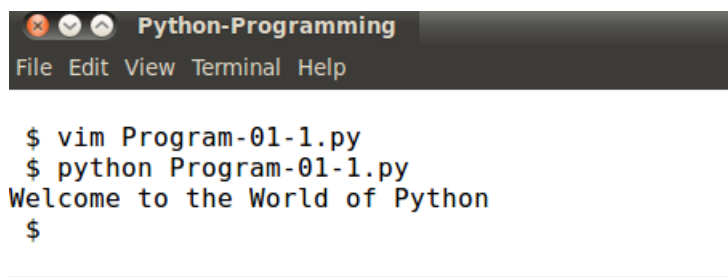
1. Open the file using `$ vim Program-01-1.py` and type the following line and save the file.

```
print("Welcome to the World Of Python")
```

2. Now type the below command in terminal and press Enter Key to Execute.

```
$ python Program-01-1.py
```

### Output



```
Python-Programming
File Edit View Terminal Help

$ vim Program-01-1.py
$ python Program-01-1.py
Welcome to the World of Python
$
```

**Note**

Recall that python is an interpreted language. So the execution takes place without any compiling, linking or producing an executable.

## Building blocks of Python Program

In order to understand the functioning of a program, we need to understand the role of the following elements. It should be noted that this perspective is in no way intended as describing the structure of a program.

**Functions** Functions are the main building blocks of any programming language. Unlike C, python does not have a `main()` function to begin. We will discuss functions in detail later.

**Variables** Variables are very flexible. You do not have to declare them first, like in other languages like C. You can assign any value to them, even if they already have a value of a different type.

**Expressions** An expression is a combination of variables, operators and values which represents a single result value.

**Statements** Statements can be expressions, assignments, function calls, or control flow statements which make up python program.

**Block** A block is a lexical grouping of statements and can be thought of as *one* logical statement. A block in python is delimited by indentation.

**Comments** Comments start with the hash character '#', and extend to end of the line. A comment may appear at the start of the line or following white space or code, but not within a string literal. A hash character within a string literal is just a hash character. Since comments are to clarify code and are not interpreted by Python, they can be omitted.

## Naming Rules in Python

There are a few set of rules for choosing variable names:

## Introduction to Python

- Must begin with a alphabet(a - z, A - Z) or underscore(\_).
- Other characters can be alphabets, digits or underscore(\_).
- Python is case sensitive; uppercase and lower case alphabets are treated as distinct.

You should ensure that you use meaningful names for your identifiers. Please note meaningful *does not* necessarily mean long. The goal is to make the program easier to read and be self-documenting.

## Keywords

Keywords are reserved identifiers that have strict meaning which cannot be used as identifiers in the program. Note that some of the keywords are capitalized.

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

Table 1: List of Python keywords

## Additional notes



### Origin of the name

Though the logo is a stylized representation of the reptile, the language is *NOT* named after a snake, but a famous British comedian Monty Python – Guido is a great fan!