

What is Python?

- Python is an open source, object-oriented, high-level powerful programming language.
- Developed by Guido van Rossum in the early 1990s. Named after Monty Python
- Python runs on many Unix variants, on the Mac, and on Windows 2000 and later.
- Available for download from <http://www.python.org>.

Python Program

- Python programs are composed of modules
- Modules contain statements
- Statements contain expressions
- Expressions create and process objects

Features of Python

Open source: Python is publicly available open source software, any one can use source code that doesn't cost anything.

Easy-to-learn: Popular (scripting/extension) language, clear and easy syntax, no type declarations, automatic memory management, high-level data types and operations, design to read (more English like syntax) and write (shorter code compared to C, C++, and Java) fast.

High-level

Language:

High-level language (closer to human) refers to the higher level of concept from machine language (for example assembly languages). Python is an example of a high-level language like C, C++, Perl, and Java with low-level optimization.

Portable:

High level languages are portable, which means they are able to run across all major hardware and software platforms with few or no change in source code. Python is portable

and can be used on Linux, Windows, Macintosh, Solaris, FreeBSD, OS/2, Amiga, AROS, AS/400 and many more.

Object-Oriented: Python is a full-featured object-oriented programming language, with features such as classes, inheritance, objects, and overloading.

Python is Interactive :

Python has an interactive console where you get a Python prompt (command line) and interact with the interpreter directly to write and test your programs. This is useful for mathematical programming.

Interpreted : Python programs are interpreted, takes source code as input, and then compiles (to portable byte-code) each statement and executes it immediately. No need to compiling or linking

Extendable : Python is often referred to as a "glue" language, meaning that it is capable to work in mixed-language environment. The Python interpreter is easily extended and can add a new built-in function or modules written in C/C++/Java code.

Libraries : Databases, web services, networking, numerical packages, graphical user interfaces, 3D graphics, others.

Supports :Support from online Python community

Python Interpreter

- In interactive mode, type Python programs and the interpreter displays the result:
- Type python into your terminal's command line
- After a short message, the >>> symbol will appear
- The above symbol signals the start of a Python interpreter's command line.
- Python interpreter evaluates inputs (For example >>> 4*(6-2) return 16)

How stable is Python?

Very stable. New, stable releases have been coming out roughly every 6 to 18 months since 1991, and this seems likely to continue. Currently there are usually around 18 months between major releases.

The latest stable releases can always be found on the [Python download page](#). There are two recommended production-ready versions at this point in time, because at the moment there are two branches of stable releases: 2.x and 3.x.

History

The name Python was selected from "Monty Python's Flying Circus" which was a British sketch comedy series created by the comedy group Monty Python and broadcast by the BBC from 1969 to 1974.

Python was created in the early 1990s by [Guido van Rossum](#) at the National Research Institute for Mathematics and Computer Science in Netherlands.

Python was created as a successor of a language called ABC (All Basic Code) and released publicly in 1991. Guido remains Python's principal author, although it includes many contributions from active user community.

Between 1991 and 2001 there are several versions released, current stable release is 3.2. In 2001 the Python Software Foundation (PSF) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

All most all Python releases are Open Source. To see the details of release versions and licence agreement of Python [check here](#).

Python Language Overview:

Python Readability - Variables:

Python:

```
is_valid = True
```

PHP:

```
$isValid = true;
```

JavaScript:

```
let isValid = true;
```

Java / C# / C++ (Strongly Typed Languages):

```
Boolean isValid = true
```

Python Readability - Control Structures:

Python:

```
price = 200

if price == 15:

    """ Do something"""

elif price == 30:

    """Do something else"""

else:

    """ Do something else"""
```

PHP:

```
$price = 200;

if ($price == 15){

    //Here

} elseif ($price == 30){

    //Here

} else {

    // Here

}
```

Python Readability - Ternary:

Python:

```
price = 200 if True else 0  
  
print (price)  
  
# 200
```

PHP:

```
<?php  
  
$price = (true) ? 200 : 0;  
  
print $price;  
  
# 200
```

Ruby:

```
price = if true then 200 else 0 end  
  
# 200
```

JavaScript:

```
let price = true ? 200 : 0;  
  
console.log(price);  
  
# 200
```

A simple program written in C++, C, Java and Python. All program prints "Hello world".

Python Program:

```
print ( "Hello World")
```

Java Program:

```
public class Hello  
{  
  
    public static void main(String argv[])  
    {
```

```
        System.out.println("Hello, World!");
    }
}
```

C++ Program:

```
#include <iostream>

int main()
{
    std::cout << "Hello World" << std::endl;

    return 0;
}
```

C Program:

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```

Python Readability - Classes:

Python:

```
class MotorCar(object):

    def __init__(self):

        """Base class constructor"""


class Car(MotorCar):

    def __init__(self):

        """Class constructor"""

        super(Car, self).__init__()
```

```
def drive(self):  
    """Drive the car"""  
  
red_car = Car()  
red_car.drive()
```

PHP:

```
// /src/Transport/MotorCar.php
```

```
namespace Transport;
```

```
class MotorCar
```

```
{  
    /**  
     * Base class constructor  
     */  
    public function __construct()  
    {  
        # ...  
    }  
}
```

```
// /src/Transport/Car.php
```

```
namespace Transport;
```

```
class Car extends MotorCar
```

```
{  
    public function __construct()
```

```

{
    //parent::__construct()
}

/**
 * Drive the car
 */
public function drive()
{
    # ...
}
}

$redCar = new Car();
$redCar->drive();

```

Python Environment:

AIX	AROS	AS/400 (OS/400)	BeOS
MorphOS	MS-DOS	OS/2	OS/390 and z/OS
Palm OS	PlayStation and PSP	Psion	QNX
RISC OS	Series 60	Solaris	VMS
Windows CE or Pocket PC	HP-UX	Linux	

Major uses of Python:

- System utilities (system admin tools, command line programs).
- Web Development.
- Graphical User Interfaces (Tkinter, gtk, Qt).
- Internet scripting.
- Embedded scripting.
- Database access and programming.
- Game programming.
- Rapid prototyping and development.
- Distributed programming

Organizations Using Python (sector wise)

Web Development : Yahoo Maps, Yahoo Groups, Google, Zope Corporation, Ultraseek, Linux Weekly News, ElasticHosts Cloud Servers, Mojam.com, hunch, Shopzilla, Movieplayer.it, Multiplayer.it.

Games: Battlefield 2, Crystal Space, Star Trek Bridge Commander, The Temple of Elemental Evil, Vampire: The Masquerade: Bloodlines, Civilization 4, QuArK (Quake Army Knife)

Graphics : Industrial Light & Magic, Walt Disney Feature Animation, HKS, Inc. (ABAQUS/CAE), RoboFog, Caligari Corporation, Blender 3D, Jasc Software, Paint Shop Pro.

Financial : Altis Investment Management, ABN AMRO Bank, Treasury Systems, Bellco Credit Union, Journyx Timesheet and Resource Management Software.

Science : National Weather Service, Radar Remote Sensing Group, Applied Maths, Biosoft, The National Research Council of Canada, Los Alamos National Laboratory (LANL) Theoretical Physics Division, AlphaGene, Inc., LLNL, NASA, Swedish Meteorological and Hydrological Institute (SMHI), Environmental Systems Research Institute (ESRI), Objexx Engineering, Nmag Computational Micromagnetics

Electronic Design Automation: Ciranova, Productivity Design Tools, Object Domain, Pardus, Red Hat, SGI, Inc., MCI Worldcom, Nokia,

Education : University of California, Irvine, Smeal College of Business, The Pennsylvania State University, New Zealand Digital Library, IT Certification Exam preparation, SchoolTool,

Business Software : Raven Bear Systems Corporation, Thawte Consulting, Advanced Management Solutions Inc., IBM, Arakn<E9>, RealNetworks, dSPACE, Escom, The Tiny Company, Nexedi, Piensa Technologies - Bufete Consultor de Mexico, Nektra, WuBook.

To see the details of the above organizations [check here](#).

Is Python a good language for beginning programmers?

Yes. It is still common to start students with a procedural and statically typed language such as Pascal, C, or a subset of C++ or Java. Students may be better served by learning Python as their first language. Python has a very simple and consistent syntax and a large standard library and, most importantly, using Python in a beginning programming course lets students concentrate on important programming skills such as problem decomposition and data type design. With Python, students can be quickly introduced to basic concepts such as loops and procedures. They can probably even work with user-defined objects in their very first course.

For a student who has never programmed before, using a statically typed language seems unnatural. It presents additional complexity that the student must master and slows the pace of the course. The students are trying to learn to think like a computer, decompose problems, design consistent interfaces, and encapsulate data. While learning to use a statically typed language is important in the long term, it is not necessarily the best topic to address in the students' first programming course.

Many other aspects of Python make it a good first language. Like Java, Python has a large standard library so that students can be assigned programming projects very early in the course that do something. Assignments aren't restricted to the standard four-function calculator and check balancing programs. By using the standard library, students can gain the satisfaction of working on realistic applications as they learn the fundamentals of programming. Using the standard library also teaches students about code reuse. Third-party modules such as PyGame are also helpful in extending the students' reach.