Introduction to Python

Guido van Rossum

From Wikipedia, the free encyclopedia

Guido van Rossum ([pronunciation?] (born 31 January[5] 1956) is a Dutch computer programmer who is best known as the author of the Python programming language. In the Python community, Van Rossum is known as a "Benevolent Dictator For Life" (BDFL), meaning that he continues to oversee the Python development process, making decisions where necessary.[6] He was employed by Google from 2005 until 7 December 2012, where he spent half his time developing the Python language. In January 2013, Van Rossum started working for Dropbox.[3]

Contents [hide]

1 Biography
2 Personal life
3 Work
   3.1 Python
   3.2 Computer Programming for Everybody
   3.3 Mondrian
   3.4 Dropbox
   3.5 Recognition
4 References
5 External links

https://en.wikipedia.org/wiki/Guido_van_Rossum
Python Timeline

1980
  Conceived
  v0.9.0
1989
  v1.0
1991
  Implementation Started
1994
  v1.5
1997
  v1.6
2000
  v2.0
2001
  v2.2
2003
  v2.3
2004
  v2.4
2006
  v2.5
2008
  v2.6
2009
  v3.0
2010
  v3.1
2011
  v3.2
2012
  v3.3
2014
  v3.4
2015
  v3.5

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

https://en.wikipedia.org/wiki/Python_%28programming_language%29
Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

https://en.wikipedia.org/wiki/Python_%28programming_language%29
Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

https://en.wikipedia.org/wiki/Python_%28programming_language%29
Python combines remarkable expressive power with very clean, simple, and compact syntax. Python is easy to learn and very well suited for an introduction to computer programming. Python is also quite similar to Matlab and a good language for doing mathematical computing. It is easy to combine Python with compiled languages, like Fortran, C, and C++, which are widely used languages for scientific computations. A seamless integration of Python with Java is offered by a special version of Python called Jython.

https://en.wikipedia.org/wiki/Python_%28programming_language%29
What is Python?

- **Website:** [www.python.org](http://www.python.org)
- **Filename extensions:** .py, .pyc, .pyd, .pyo, pyw, .pyz
- **OS:** Cross-platform
- **Major implementations:** CPython, IronPython, Jython, PyPy
- **Influenced by:** ABC, ALGOL 68, C, C++, Dylan, Haskell, Icon, Java, Lisp, Modula-3, Perl
- **Supported Paradigms:** object-oriented, imperative, functional, procedural, reflective

[https://en.wikipedia.org/wiki/Python_%28programming_language%29](https://en.wikipedia.org/wiki/Python_%28programming_language%29)
There are many different programming languages in the world, and most were designed to perform tasks in a certain way or even make it easier for a particular profession’s work to be done with greater ease. Choosing the correct tool makes your life easier!

Python vs R vs SAS vs SQL
Python (general purpose): Many data scientists prefer to use Python because it provides a wealth of libraries, such as NumPy, SciPy, MatPlotLib, pandas, and Scikit-learn, to make data science tasks significantly easier. Python is also a precise language that makes it easy to use multi-processing on large datasets, reducing the time required to analyze them. The data science community has also stepped up with specialized IDEs, such as Anaconda, that implement the IPython Notebook concept, which makes working with data science calculations significantly easier.
R (special purpose statistical): In many respects, Python and R share the same sorts of functionality but implement it in different ways. Depending on which source you view, Python and R have about the same number of proponents, and some people use Python and R interchangeably (or sometimes in tandem). Unlike Python, R provides its own environment, so you don’t need a third-party product such as Anaconda. However, R doesn’t appear to mix with other languages with the ease that Python provides.
**R (special purpose statistical):** In many respects, Python and R share the same sorts of functionality but implement it in different ways. Depending on which source you view, Python and R have about the same number of proponents, and some people use Python and R interchangeably (or sometimes in tandem). Unlike Python, R provides its own environment, so you don’t need a third-party product such as Anaconda. However, R doesn’t appear to mix with other languages with the ease that Python provides.
**SAS (business statistical analysis):** The Statistical Analysis System (SAS) language is popular because it makes data analysis, business intelligence, data management, and predictive analytics easy. The SAS Institute originally created SAS as a means to perform statistical analysis. In other words, this is a business-specific language — one used to make decisions rather than to perform handwriting analysis or to detect specific natural patterns.
**SAS (business statistical analysis):** The Statistical Analysis System (SAS) language is popular because it makes data analysis, business intelligence, data management, and predictive analytics easy. The SAS Institute originally created SAS as a means to perform statistical analysis. In other words, this is a business-specific language — one used to make decisions rather than to perform handwriting analysis or to detect specific natural patterns.
SQL (database management): The most important thing to remember about Structured Query Language (SQL) is that it focuses on data rather than tasks. Businesses can’t operate without good data management — the data is the business. Large organizations use some sort of relational database, which is normally accessible with SQL, to store their data. Most Database Management System (DBMS) products rely on SQL as their main language, and DBMS usually has a large number of data analysis and other data science features built in. Because you’re accessing the data natively, there is often a significant speed gain in performing data science tasks this way. Database Administrators (DBAs) generally use SQL to manage or manipulate the data rather than necessarily perform detailed analysis of it.
A common problem among Python programmers is to choose between v2 or v3. The general recommendation is to go for version 3, but programs are then not compatible with v2 and vice versa. There is still a problem that much useful mathematical software in Python has not yet been ported to v3! Therefore, scientific computing with Python still goes mostly with v2.

A widely used strategy for software developers who want to write Python code that works with both versions, is to develop for v2.7, which is very close to what is accepted in v3, and then use the translation tool ‘2to3’ to automatically translate the code to v3.
Version 2 or 3?

When using v2.7, one should employ the newest syntax and modules that make the differences between version 2 and 3 very small! The two most significant differences for most users between v2 and v3 are:

• `a/b` implies float division in v3 if `a` and `b` are integers

• `print 'Hello'` in v2 must be turned into the function call `print('Hello')` in v3.

You can read more at:

• https://wiki.python.org/moin/Python2orPython3
Where to learn more about the differences between one or the other:

• https://wiki.python.org/moin/Python2orPython3
Version 2 or 3?

http://astrofrog.github.io/blog/2015/05/09/2015-survey-results/
Version 2 or 3?

http://astrofrog.github.io/blog/2015/05/09/2015-survey-results/
Introduction to Python

Version 2 or 3?

http://astrofrog.github.io/blog/2015/05/09/2015-survey-results/
The following identifiers are used as reserved words, or keywords of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

<table>
<thead>
<tr>
<th>v2.7</th>
<th>and</th>
<th>del</th>
<th>from</th>
<th>not</th>
<th>while</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>as</td>
<td>elif</td>
<td>global</td>
<td>or</td>
<td>with</td>
</tr>
<tr>
<td>assert</td>
<td>break</td>
<td>else</td>
<td>if</td>
<td>pass</td>
<td>yield</td>
</tr>
<tr>
<td>class</td>
<td>continue</td>
<td>except</td>
<td>import</td>
<td>print</td>
<td></td>
</tr>
<tr>
<td></td>
<td>class</td>
<td>exec</td>
<td>in</td>
<td>raise</td>
<td></td>
</tr>
<tr>
<td></td>
<td>continue</td>
<td>finally</td>
<td>is</td>
<td>return</td>
<td></td>
</tr>
<tr>
<td></td>
<td>def</td>
<td>for</td>
<td>lambda</td>
<td>try</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>v3.5.1</th>
<th>False</th>
<th>class</th>
<th>finally</th>
<th>is</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td>continue</td>
<td>for</td>
<td></td>
<td>try</td>
</tr>
<tr>
<td>True</td>
<td>and</td>
<td>def</td>
<td>from</td>
<td>lambda</td>
<td></td>
</tr>
<tr>
<td></td>
<td>as</td>
<td>del</td>
<td>global</td>
<td>nonlocal</td>
<td></td>
</tr>
<tr>
<td>assert</td>
<td>assert</td>
<td>elif</td>
<td>if</td>
<td>not</td>
<td></td>
</tr>
<tr>
<td>break</td>
<td></td>
<td>else</td>
<td>import</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>except</td>
<td>in</td>
<td>pass</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>raise</td>
<td></td>
</tr>
</tbody>
</table>
How to get Python?

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

It’s entirely possible to obtain a generic copy of Python and add all of the required data science libraries to it. The process can be difficult because you need to ensure that you have all the required libraries in the correct versions to ensure success. In addition, you need to perform the configuration required to ensure that the libraries are accessible when you need them. Fortunately, going through the required work is not necessary because a number of Python data science products are available for you to use. These products provide everything needed to get started with science projects.
First example

For example, if you want to display something on-screen, you simply tell Python to print it, like this:

```
v2.7   print "Hello There!"

v3.5.1  print("Hello There!")
```
Scientific Distributions

It’s entirely possible to obtain a generic copy of Python and add all of the required data science libraries to it. The process can be difficult because you need to ensure that you have all the required libraries in the correct versions to ensure success. In addition, you need to perform the configuration required to ensure that the libraries are accessible when you need them. Fortunately, going through the required work is not necessary because a number of Python data science products are available for you to use. These products provide everything needed to get started with science projects.
Continuum Analytics Anaconda

https://www.continuum.io/downloads
Enthought Canopy Express is a free product for producing both technical and scientific applications using Python. Only Canopy Express is free, the full Canopy product comes at a cost.

One of the advantages of Canopy Express is that Enthought is heavily involved in providing support for both students and teachers. People also can take classes, including online classes, that teach the use of Canopy Express in various ways (see https://training.enthought.com/courses). Also offered is live classroom training specifically designed for the data scientist; read about this training at https://www.enthought.com/services/training/data-science.
The python(x,y) Integrated Development Environment (IDE) is a community project hosted on Google at https://code.google.com/p/pythonxy/. It’s a Windows-only product, so you can’t easily use it for cross-platform needs. (In fact, it supports only Windows Vista, Windows 7, and Windows 8.)

Because pythonxy uses the GNU General Public License (GPL) v3 (see http://www.gnu.org/licenses/gpl.html), you have no add-ons, training, or other paid features to worry about. No one will come calling at your door hoping to sell you something. In addition, you have access to all the source code for pythonxy, so you can make modifications if desired.
The name tells you that WinPython is a Windows-only product that you can find at http://winpython.sourceforge.net/. This product is actually a takeoff of pythonxy and isn’t meant to replace it. Quite the contrary: WinPython is simply a more flexible way to work with pythonxy. You can read about the motivation for creating WinPython at http://sourceforge.net/p/winpython/wiki/Roadmap/.

The bottom line for this product is that you gain flexibility at the cost of friendliness and a little platform integration. However, for developers who need to maintain multiple versions of an IDE, WinPython may make a significant difference.
Type the following piece of code and run it. NOTE: This program will need a parameter from the command line!

```python
#!/usr/bin/env python
import sys, math
r = float(sys.argv[1])
s = math.sin(r)
print "Hello, World! sin(" + str(r) + ")=" + str(s)
```
References

Python Crash Course: A Hands-On, Project-Based Introduction to Programming
Eric Matthes

A Primer on Scientific Programming with Python
Hans Petter Langtangen

Python for Data Science For Dummies
John Paul Mueller, Luca Massaron
Python relies on indentation to create various language features, such as conditional statements. One of the most common errors that developers encounter is not providing the proper indentation for code.

```python
if 1 < 2:
    print("1 is less than 2")
```
Operators

The following tokens are operators

v2.7

+  -  *  **  /  //  %
<<  >>  &  |  ^  ~
<  >  <=  >=  ==  !=  <>

v3.5.1

+  -  *  **  /  //  %  @
<<  >>  &  |  ^  ~
<  >  <=  >=  ==  !=