Real-world code using the most natural idioms in Python 2.x usually does various things that are not quite compatible with Python 3.x, and the code that is version-neutral must be written in a careful way to achieve that.

The bottom line is that Python 3.x is a better language than Python 2.x

it takes time to move users to the latest version.

Programs written on Python 2.1—or even on Python 1.5.2—where simply keeping the platform consistent was easier than testing a transition. “If it ain’t broke, don’t fix it!”

free to download from python.org and other sites (as are many other programming lan‐ guages and developer tools). Apple OS X ships with Python 2.x pre‐ installed.Linux® distributions have traditionally shipped with Python 2.x installed, but increasingly the latest versions of these “distros” use Python 3.x. Microsoft does do this.

Unix: follows (PEP 394)

“python” Command on Unix-Like Systems, specifies the recommended configuration of Python on Unix-like systems.(FreeBSD, OpenBSD, NetBSD)(Berkeley Software Distribution or Berkeley Standard Distribution)

Code for Unix (redhat and fedora):

#!/path/to/executable (or) #!/usr/bin/env executable

How to distribute ?

* DNF (default package manager) instead of Yum, works for Python 2.
* Python 3 in the root
* Anaconda for python 3
* Cloud init python 3

Ubuntu :(Ubuntu and Debian)

Python 3 the default, preferred Python version in the distros.

Python distributed by ISO standards are **internationally recognized specifications for products, services and systems**.

‘to remove Python 2 from the scene.’

Usage statistics :

---------------------------------------------------------------------

how many local applications? How many servers? Serving how many clients? How much CPU time used in the process? How important are the various applications? How many lines of code in the version?

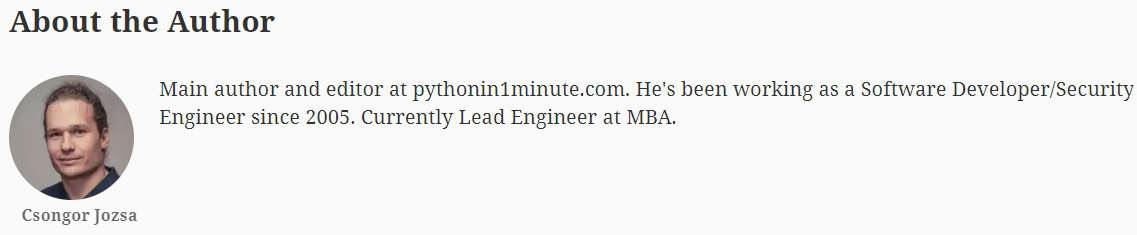
---------------------------------------------------------------------

CSVs

Fate of 2 series:

My expectation would be to see Python 2 fade away quite soon. I certainly don’t expect it to stay relevant for more than a few years. It’s true that it’s been around for a while and yes, there are ancient languages like Fortran or COBOL that just seem to never die, but I don’t think Python 2 is anything like that. Unlike these systems, Python 2 has a successor and migration to Python 3 is rather simple.

Python 2 has no place anymore in the startup world, no new projects should choose Python 2 to build on. More mature companies with an existing, bigger codebase will also be forced to move to Python 3. With the end of official support for Python 2, maintainability, the lack of security fixes and compliance issues will be a huge driving force for migration.



--------------------------------------------------------------------------------------------------------------------------------

There are 25 search engines that are used to calculate the TIOBE index. The selected search engines are the 25 highest ranked websites of [Alexa](http://www.alexa.com/" \t "https://www.tiobe.com/tiobe-index//programming-languages-definition/_blank) that meet the following conditions:

* The entry page of the site contains a search facility
* The result of querying the site contains an indication of the number of page hits
* The results should be available in HTML with clear tags
* Search engines in languages with special characters should be encoded properly
* The search engine should at least return 1 hit for 1 query
* The results of querying the site shouldn't contain too many outliers
* Porn sites are excluded

### End of Service Notice

We retired Alexa.com on May 1, 2022, after more than two decades of helping you find, reach, and convert your digital audience. Thank you for making us your go-to resource for content research, competitive analysis, keyword research, and so much more.

Where python is used in the scientific communities :

**Astropy** is a collection of astronomy-related software tools. The main Astropy package has features geared toward professional astronomers and astrophysicists, although it may be beneficial to anyone designing astronomy applications.

**Biopython** is an open-source Python library for computational biology and bioinformatics. It has classes for representing biological sequences and annotations. A number of file formats can be read and written by the library.

**Bokeh** is a Python interactive visualization package that displays content in modern web browsers. It may assist anyone who wants to create interactive charts, dashboards, and data applications fast and easily. Bokeh's goal is to give elegant, concise development of innovative graphics in the style of D3.js while also providing high-performance interactivity over very big or streaming datasets.

**Cubes** is a lightweight Python framework and set of tools for developing reporting and analytical applications, OLAP, multidimensional analysis, and data browsing.

**Dask** is a parallel computing package for analytic computing that is made up of two parts: Big Data collections such as parallel arrays, dataframes, and lists that expand standard interfaces such as NumPy, Pandas, or Python iterators to larger-than-memory or distributed settings; dynamic task scheduling optimized for computation and interactive computational workloads.

**DEAP** is a system for rapid prototyping and testing of ideas based on evolutionary computation. It includes the data structures and methods needed to construct the most common evolutionary computation approaches, including genetic algorithms, genetic programming, evolution strategies, particle swarm optimization, differential evolution, and distribution algorithm estimates.

**DataMelt**, sometimes known as DMelt, is a program for quantitative computing, statistics, Big Data analysis, and scientific visualization. Python/Jython, BeanShell, Groovy, Ruby, and Java are among the scripting languages supported. Natural sciences, engineering, modeling, and financial market analysis are just some of the applications of the library.Graph-tool is a module that allows you to manipulate and analyze graphs statistically.

**Matplotlib** is a Python 2D charting package that generates high-quality figures in a range of hard-copy and interactive cross-platform formats. Plots, histograms, power spectra, bar charts, error charts, scatter plots, and more may all be generated using it.

**Mlpy** is a machine learning library based on the GNU Scientific Libraries NumPy/SciPy. It offers a wide range of supervised and unsupervised machine learning methods, with the goal of striking a balance between modularity, maintainability, reproducibility, usability, and efficiency.

**NetworkX** is a graph-analysis library that allows you to design, alter, and investigate complex networks' structure, dynamics, and functions.

**Nilearn** is a Python tool for learning statistics from neuroimaging data quickly and easily. This library simplifies the application of advanced machine learning, pattern recognition, and multivariate statistical techniques to neuroimaging data for applications like MVPA (Multi-Voxel Pattern Analysis), decoding, predictive modeling, functional connectivity, brain parcellations, and connectomes.

**NumPy** is a Python module that adds support for massive, multidimensional arrays and matrices, as well as a vast library of high-level mathematical functions to operate on them.

**Pandas** is a data manipulation and analysis package that includes data structures and methods for working with numerical tables and time series.

**Pipenv** is a Python utility that aims to combine the best of all packaging worlds. It builds and manages a virtualenv for your projects automatically, as well as adding and removing packages from your Pipfile as you install and uninstall packages. Pipenv is primarily designed to make setting up a working environment for consumers and developers of apps simple.

**PsychoPy** is a Python tool for creating neuroscience and experimental psychology investigations. It is intended to be used in a variety of neuroscience, psychology, and psychophysical research to show stimuli and gather data.

**PySpark** is Apache Spark's Python API. Spark is a massive data processing distributed computing framework. It's a unified analytics engine that prioritizes speed, simplicity, and generality. Spark has streaming, machine learning, and graph processing modules. It's also entirely free and open-source.

**Weka** is a Java-based machine learning software suite developed by New Zealand's University of Waikato. It includes a set of data analysis and predictive modeling visualization tools and algorithms, as well as graphical user interfaces for quick access to these operations. The python-weka-wrapper package makes it simple to use Python to run Weka algorithms and filters.

**PyTorch** is a deep learning framework that allows you to experiment quickly and easily.

Tensor computing with powerful GPU acceleration and deep neural networks built on a tape-based autodiff system are two high-level features provided by this package. It can be used as a deep learning research platform that gives maximum flexibility and performance, or as a replacement for numpy to take advantage of GPU capability.

**SQLAlchemy** is an open-source SQL toolkit and Object-Relational Mapper that provides complete SQL capability and flexibility to application developers. It offers a complete set of well-known enterprise-level persistence patterns, all of which have been translated into a simple and Pythonic domain language for efficient and high-performance database access. The library's major purpose is to transform the way we think about databases and SQL.

**SageMath** is a mathematical software system that includes capabilities for algebra, combinatorics, numerical mathematics, number theory, and calculus, among other topics. It supports procedural, functional, and object-oriented constructs with Python.

**ScientificPython** is a collection of scientific computing modules written in Python. Geometry, mathematical functions, statistics, physical units, IO, visualization, and parallelization are all supported.

**Scikit-image** is a library for image processing. Segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature recognition, and other algorithms are included.

**Scikit-learn** is a library for machine learning. Support vector machines, random forests, gradient boosting, k-means, and DBSCAN are among the classification, regression, and clustering algorithms included. The library is designed to work with the NumPy and SciPy Python numerical and scientific libraries.

**SciPy** is a scientific and technical computing library used by scientists, analysts, and engineers. It includes modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, and other scientific and engineering applications.

**SCOOP** is a Python package for distributing concurrent parallel workloads over a variety of platforms, ranging from heterogeneous workstation grids to supercomputers.

**SunPy** is a Python-based data analysis environment that focuses on delivering the software needed to evaluate solar and heliospheric data.

**SymPy** is a symbolic computation package that includes functions for anything from simple symbolic arithmetic through calculus, algebra, discrete mathematics, and quantum physics. It offers computer algebra capabilities as a standalone application, a library for other applications, or as a live online application.

**TensorFlow** is an open-source software library for machine learning that was created by Google to suit their demands for systems that could design and train neural networks to discover and decode patterns and correlations, similar to how humans learn and reason. It is being utilized by Google for both research and production, frequently replacing its closed-source predecessor, DistBelief.

**Theano** is a Python toolkit for numerical computing that allows you to quickly write, optimize, and evaluate mathematical equations involving multidimensional arrays.

**TomoPy** is an open-source Python toolbox for processing tomographic data and reconstructing images. provides a collaborative platform for analyzing synchrotron tomographic data with the purpose of unifying comparable jobs.

**Veusz i**s a scientific plotting and graphing tool that creates publication-quality charts in vector formats such as PDF, PostScript, and SVG.

**Beautiful Soup** is a useful tool that can help you save time and effort. Scraping information from web pages is simple using the library. It extracts data from HTML and XML files and integrates with your preferred parser to offer fluent navigation, search, and modification of the parse tree.

**Scrapy** :It can be used for everything from data mining to monitoring and testing. Built-in support for choosing and extracting data from HTML/XML sources using CSS selectors and XPath expressions, as well as an interactive shell console for trying out expressions to scrape data, are two of the most powerful capabilities.

**Plotly** is an open-source framework for creating dynamic, web-based visualizations that may be presented in Jupyter notebooks, saved as independent HTML files, or included in Dash-based Python web apps. It has over four different chart kinds that can be used to illustrate data in a variety of fields, such as statistics, finance, geography, and science. It's frequently referred to as "plotly.py" to distinguish it from the JavaScript library.

**Seaborn** is a popular Python data visualization toolkit for creating statistical visuals. It's built on matplotlib and can be used in any of matplotlib's supported contexts. It provides a high-level interface, unlike matplotlib.

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. For example, a Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program. Because of the run-time typing, Python's run time must work harder than Java's. For example, when evaluating the expression a+b, it must first inspect the objects a and b to find out their type, which is not known at compile time. It then invokes the appropriate addition operation, which may be an overloaded user-defined method. Java, on the other hand, can perform an efficient integer or floating point addition, but requires variable declarations for a and b, and does not allow overloading of the + operator for instances of user-defined classes.

For these reasons, Python is much better suited as a "glue" language, while Java is better characterized as a low-level implementation language. In fact, the two together make an excellent combination. Components can be developed in Java and combined to form applications in Python; Python can also be used to prototype components until their design can be "hardened" in a Java implementation. To support this type of development, a Python implementation written in Java is under development, which allows calling Python code from Java and vice versa. In this implementation, Python source code is translated to Java bytecode (with help from a run-time library to support Python's dynamic semantics).

|  |  |
| --- | --- |
| **Python** | **C++** |
| Supports Garbage Collection | Does not support Garbage Collection |
| Python programs are easier to write | Not easy in contrast to Python because of its complex syntax. |
| Run through interpreter | C++ is pre-compiled |
| Rapid Prototyping is possible because of the small size of the code | Rapid Prototyping not possible because of larger code size |
| Python is difficult to be installed on a windows box | Not have an issue while installing in the windows system. |
| Python is nearer to plain English language. Therefore, it is easy to learn language. | C++ has a stiff learning curve as it has lots of predefined syntaxes and structure |
| Python is slower. | C++ is faster than Python |
| Python has more English like syntax, so readability is very high. | C++ code readability is weak when compared with Python code. |
| In Python, variables are accessible outside the loop. | The scope of the C++ variables is limited within the loops. |
| Famous companies using Python are Google, Lyft, Twitch, Telegram. | Famous companies using C++ are Uber technologies, Netflix, Spotify, Instagram. |
| TIOBE rating is 3 | TIOBE rating is 4 |
| The average salary for a Python Developer is $120,359 per year in the United States of America. | The average salary for a C++ Developer is $108,809 per year in the United States. |

## **Disadvantage of C++**

Here, are cons/drawbacks of using C++ language

* It offers no security for your code
* Complex language to use in a very large high-level program.
* It is used for platform-specific applications commonly.
* When C++ used for web applications it is complex and difficult to debug.
* C++ can’t support garbage collection.
* No built-in support for threads
* C++ is not as portable as other high-level programming languages. So, when you want to compile the C++ code, you need to run it on another machine.
* If the same operation has to be executed more than one time, the same sequence has to copy at some places, which increases code redundancy.