



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)  
[Donate to Wikipedia](#)  
[Wikipedia store](#)

Interaction

[Help](#)  
[About Wikipedia](#)  
[Community portal](#)  
[Recent changes](#)  
[Contact page](#)

Tools

[What links here](#)  
[Related changes](#)  
[Upload file](#)  
[Special pages](#)  
[Permanent link](#)  
[Page information](#)  
[Wikidata item](#)  
[Cite this page](#)

Print/export

[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)

Languages

[Español](#)  
[Русский](#)  
[Српски / srpski](#)  
[Türkçe](#)

[Edit links](#)

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

# History of Python

From Wikipedia, the free encyclopedia

*Main article: [Python \(programming language\)](#)*

The [programming language Python](#) was conceived in the late 1980s,<sup>[1]</sup> and its implementation was started in December 1989<sup>[2]</sup> by [Guido van Rossum](#) at [CWI](#) in [the Netherlands](#) as a successor to [ABC](#) capable of [exception handling](#) and interfacing with the [Amoeba operating system](#).<sup>[3]</sup> Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, *[Benevolent Dictator for Life](#)* ([BDFL](#)).<sup>[4]</sup><sup>[5]</sup> Python was named for the BBC TV show *[Monty Python's Flying Circus](#)*.<sup>[6]</sup>




Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting [garbage collector](#) (in addition to [reference counting](#)) for [memory management](#) and support for [Unicode](#). However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.<sup>[7]</sup>

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008<sup>[8]</sup> after a long period of testing. Many of its major features have also been [backported](#) to the backwards-compatible Python 2.6 and 2.7.<sup>[9]</sup>

## Contents

- [Early history](#)
- [Version 1](#)
  - [BeOpen](#)
- [Version 2](#)
- [Version 3](#)
  - [Compatibility](#)
  - [Features](#)
- [Version release dates](#)
- [See also](#)
- [References](#)
- [External links](#)

## Early history [\[edit\]](#)

In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.<sup>[10]</sup> Already present at this stage in development were [classes with inheritance](#), exception handling, functions, and the core datatypes of `list`, `dict`, `str` and so on. Also in this initial release was a [module system](#) borrowed from [Modula-3](#); Van Rossum describes the module as "one of Python's major programming units".<sup>[1]</sup> Python's exception model also resembles Modula-3's, with the addition of an `else` clause.<sup>[3]</sup> In 1994 [comp.lang.python](#), the primary discussion forum for Python, was formed, marking a milestone in the growth of Python's userbase.<sup>[1]</sup>

## Version 1 [\[edit\]](#)

Python reached version 1.0 in January 1994. The major new features included in this release were the functional programming tools `lambda`, `map`, `filter` and `reduce`. Van Rossum stated that "Python acquired lambda, reduce(), filter() and map(), courtesy of a [Lisp](#) hacker who missed them and submitted working patches".<sup>[11]</sup>

The last version released while Van Rossum was at CWI was Python 1.2. In 1995, Van Rossum continued his work on Python at the [Corporation for National Research Initiatives](#) (CNRI) in [Reston, Virginia](#) whence he released several versions.

By version 1.4, Python had acquired several new features. Notable among these are the Modula-3 inspired [keyword arguments](#) (which are also similar to [Common Lisp](#)'s keyword arguments) and built-in

support for [complex numbers](#). Also included is a basic form of [data hiding](#) by [name mangling](#), though this is easily bypassed.<sup>[12]</sup>

During Van Rossum's stay at CNRI, he launched the [Computer Programming for Everybody](#) (CP4E) initiative, intending to make programming more accessible to more people, with a basic "literacy" in programming languages, similar to the basic English literacy and mathematics skills required by most employers. Python served a central role in this: because of its focus on clean syntax, it was already suitable, and CP4E's goals bore similarities to its predecessor, ABC. The project was funded by [DARPA](#).<sup>[13]</sup> As of 2007, the CP4E project is inactive, and while Python attempts to be easily learnable and not too arcane in its syntax and semantics, reaching out to non-programmers is not an active concern.<sup>[14]</sup>

## BeOpen [\[edit\]](#)

In 2000, the Python core development team moved to [BeOpen.com](#) to form the BeOpen [PythonLabs](#) team. CNRI requested that a version 1.6 be released, summarizing Python's development up to the point at which the development team left CNRI. Consequently, the release schedules for 1.6 and 2.0 had a significant amount of overlap.<sup>[7]</sup> Python 2.0 was the only release from BeOpen.com. After Python 2.0 was released by BeOpen.com, Guido van Rossum and the other PythonLabs developers joined [Digital Creations](#).

The Python 1.6 release included a new CNRI license that was substantially longer than the CWI license that had been used for earlier releases. The new license included a clause stating that the license was governed by the laws of the [State of Virginia](#). The [Free Software Foundation](#) argued that the choice-of-law clause was incompatible with the [GNU General Public License](#). BeOpen, CNRI and the FSF negotiated a change to Python's [free software license](#) that would make it GPL-compatible. Python 1.6.1 is essentially the same as Python 1.6, with a few minor bug fixes, and with the new GPL-compatible license.<sup>[15]</sup>

## Version 2 [\[edit\]](#)

Python 2.0 introduced [list comprehensions](#), a feature borrowed from the [functional programming](#) languages [SETL](#) and [Haskell](#). Python's syntax for this construct is very similar to Haskell's, apart from Haskell's preference for punctuation characters and Python's preference for alphabetic keywords. Python 2.0 also introduced a [garbage collection](#) system capable of collecting reference cycles.<sup>[7]</sup>

Python 2.1 was close to Python 1.6.1, as well as Python 2.0. Its license was renamed [Python Software Foundation License](#). All code, documentation and specifications added, from the time of Python 2.1's alpha release on, is owned by the [Python Software Foundation](#) (PSF), a non-profit organization formed in 2001, modeled after the [Apache Software Foundation](#).<sup>[15]</sup> The release included a change to the language specification to support nested scopes, like other [statically scoped](#) languages.<sup>[16]</sup> (The feature was turned off by default, and not required, until Python 2.2.)

A major innovation in Python 2.2 was the unification of Python's types (types written in C) and classes (types written in Python) into one hierarchy. This single unification made Python's object model purely and consistently object oriented.<sup>[17]</sup> Also added were [generators](#) which were inspired by [Icon](#).<sup>[18]</sup>

Python 2.5 was released on September 2006 <sup>[19]</sup> and introduced the `with` statement, which encloses a code block within a context manager (for example, acquiring a [lock](#) before the block of code is run and releasing the lock afterwards, or opening a [file](#) and then closing it), allowing [Resource Acquisition Is Initialization](#) (RAII)-like behavior and replacing a common try/finally idiom. <sup>[20]</sup>

Python 2.6 was released to coincide with Python 3.0, and included some features from that release, as well as a "warnings" mode that highlighted the use of features that were removed in Python 3.0.<sup>[21][9]</sup> Similarly, Python 2.7 coincided with and included features from Python 3.1,<sup>[22]</sup> which was released on June 26, 2009. Parallel 2.x and 3.x releases then ceased, and Python 2.7 was the last release in the 2.x series.<sup>[23]</sup> In November 2014, it was announced that Python 2.7 would be supported until 2020, but users were encouraged to move to Python 3 as soon as possible.<sup>[24]</sup>

## Version 3 [\[edit\]](#)

Python 3.0 (also called "Python 3000" or "Py3K") was released on December 3, 2008.<sup>[8]</sup> It was designed to rectify fundamental design flaws in the language—the changes required could not be implemented while retaining full backwards compatibility with the 2.x series, which necessitated a new major version number. The guiding principle of Python 3 was: "reduce feature duplication by removing old ways of doing things".

Python 3.0 was developed with the same philosophy as in prior versions. However, as Python had accumulated new and redundant ways to program the same task, Python 3.0 had an emphasis on

removing duplicative constructs and modules, in keeping with "There should be one— and preferably only one —obvious way to do it".

Nonetheless, Python 3.0 remained a [multi-paradigm language](#). Coders still had options among [object-orientation](#), [structured programming](#), [functional programming](#) and other paradigms, but within such broad choices, the details were intended to be more obvious in Python 3.0 than they were in Python 2.x.

## Compatibility [\[edit\]](#)

Python 3.0 broke [backward compatibility](#), and much Python 2 code does not run unmodified on Python 3. Python's [dynamic typing](#) combined with the plans to change the semantics of certain methods of dictionaries, for example, made perfect [mechanical translation](#) from Python 2.x to Python 3.0 very difficult. A tool called "2to3" does the parts of translation that can be done automatically. At this, 2to3 appeared to be fairly successful, though an early review noted that there were aspects of translation that such a tool would never be able to handle.<sup>[25]</sup> Prior to the roll-out of Python 3, projects requiring compatibility with both the 2.x and 3.x series were recommended to have one source (for the 2.x series), and produce releases for the Python 3.x platform using 2to3. Edits to the Python 3.x code were discouraged for so long as the code needed to run on Python 2.x.<sup>[9]</sup> This is no longer recommended; as of 2012 the preferred approach is to create a single code base that can run under both Python 2 and 3 using compatibility modules.<sup>[26]</sup>

## Features [\[edit\]](#)

Some of the major changes included for Python 3.0 were:

- Changing `print` so that it is a built-in function, not a statement. This made it easier to change a module to use a different print function, as well as making the syntax more regular. In Python 2.6 and 2.7 `print()` is available as a builtin but is masked by the print statement syntax, which can be disabled by entering `from __future__ import print_function` at the top of the file.<sup>[27]</sup>
- Removal of the Python 2 `input` function, and the renaming of the `raw_input` function to `input`. Python 3's `input` function behaves like Python 2's `raw_input` function, in that the input is always returned as a string rather than being evaluated as an expression.
- Moving `reduce` (but not `map` or `filter`) out of the built-in namespace and into `functools` (the rationale being that operations using *reduce* are expressed more clearly using an accumulation loop);<sup>[28]</sup>
- Adding support for optional function annotations that can be used for informal type declarations or other purposes;<sup>[29]</sup>
- Unifying the `str` / `unicode` types, representing text, and introducing a separate immutable `bytes` type; and a mostly corresponding mutable `bytearray` type, both of which represent arrays of `bytes`;<sup>[30]</sup>
- Removing backward-compatibility features, including old-style classes, string exceptions, and implicit relative imports.
- A change in [integer division](#) functionality. (In Python 2, `5 / 2` is `2`. In Python 3, `5 / 2` is `2.5`, and `5 // 2` is `2`).

Subsequent releases in the Python 3.x series have included additional, substantial new features; all ongoing development of the language is done in the 3.x series.

## Version release dates [\[edit\]](#)

Release dates for the major and minor versions:<sup>[31][32]</sup>

- Implementation started - December, 1989
- Internal releases at [Centrum Wiskunde & Informatica](#) - 1990
- Python 0.9.0 - February 20, 1991
  - Python 0.9.1 - February, 1991
  - Python 0.9.2 - Autumn, 1991
  - Python 0.9.4 - December 24, 1991
  - Python 0.9.5 - January 2, 1992
  - Python 0.9.6 - April 6, 1992
  - Python 0.9.8 - January 9, 1993
  - Python 0.9.9 - July 29, 1993
- Python 1.0 - January 1994

- Python 1.2 - April 10, 1995
- Python 1.3 - October 12, 1995
- Python 1.4 - October 25, 1996
- Python 1.5 - December 31, 1997
- Python 1.6 - September 5, 2000
- Python 2.0 - October 16, 2000
  - Python 2.1 - April 17, 2001
  - Python 2.2 - December 21, 2001
  - Python 2.3 - July 29, 2003
  - Python 2.4 - November 30, 2004
  - Python 2.5 - September 19, 2006
  - Python 2.6 - October 1, 2008
  - Python 2.7 - July 3, 2010
- Python 3.0 - December 3, 2008
  - Python 3.1 - June 27, 2009
  - Python 3.2 - February 20, 2011
  - Python 3.3 - September 29, 2012
  - Python 3.4 - March 16, 2014
  - Python 3.5 - September 13, 2015
  - Python 3.6 - December 23, 2016

## See also [edit]

- [History of software engineering](#)

## References [edit]

- ↑ **<sup>a</sup> <sup>b</sup> <sup>c</sup>** "The Making of Python" . Artima Developer. Retrieved March 22, 2007.
- ↑ "A Brief Timeline of Python" . Guido van Rossum. Retrieved 2009-01-20.
- ↑ **<sup>a</sup> <sup>b</sup>** "Why was Python created in the first place?" . Python FAQ. Retrieved March 22, 2007.
- ↑ Guido van Rossum (July 31, 2008). "Origin of BDFL" . Retrieved August 1, 2008.
- ↑ "Python Creator Scripts Inside Google" . www.eweek.com. Retrieved May 13, 2008.
- ↑ "General Python FAQ - Why is it called Python?" .
- ↑ **<sup>a</sup> <sup>b</sup> <sup>c</sup>** A.M. Kuchling and Moshe Zadka. "What's New in Python 2.0" . Archived from the original  on December 14, 2009. Retrieved March 22, 2007.
- ↑ **<sup>a</sup> <sup>b</sup>** "Welcome to Python.org" . python.org. Retrieved December 27, 2016.
- ↑ **<sup>a</sup> <sup>b</sup> <sup>c</sup>** "PEP 3000 – Python 3000" . python.org. Retrieved December 27, 2016.
- ↑ "HISTORY" . *Python source distribution*. Python Foundation. Retrieved 2017-11-23.
- ↑ Guido van Rossum. "The fate of reduce() in Python 3000" . Artima Developer. Retrieved 2007-03-22.
- ↑ "LJ #37: Python 1.4 Update" . Archived from the original  on May 1, 2007. Retrieved April 29, 2007.
- ↑ Guido van Rossum. "Computer Programming for Everybody" . Retrieved 2007-03-22.
- ↑ "Computer Programming for Everybody" . Python Software Foundation. Archived from the original  on March 29, 2007. Retrieved March 22, 2007.
- ↑ **<sup>a</sup> <sup>b</sup>** "History of the software" . *Python Library Reference*. Archived from the original  on March 29, 2007. Retrieved March 22, 2007.
- ↑ Jeremy Hylton. "Statically Nested Scopes" . Retrieved 2007-03-22.
- ↑ A.M. Kuchling (December 21, 2001). "PEPs 252 and 253: Type and Class Changes" . *What's New in Python 2.2*. Python Foundation. Archived from the original  on September 17, 2008. Retrieved September 5, 2008.
- ↑ A.M. Kuchling (2001-12-21). "PEP 255: Simple Generators" . *What's New in Python 2.2*. Python Foundation. Retrieved 2008-09-05.
- ↑ https://www.python.org/download/releases/2.5/
- ↑ https://www.python.org/download/releases/2.5/highlights/
- ↑ Neal Norwitz; Barry Warsaw (2006-06-29). "PEP 361 – Python 2.6 and 3.0 Release Schedule" . Retrieved 2012-10-07.
- ↑ A.M. Kuchling (2010-07-03). "What's New in Python 2.7" . Retrieved 2012-10-07. "Much as Python 2.6 incorporated features from Python 3.0, version 2.7 incorporates some of the new features in Python 3.1. The 2.x series continues to provide tools for migrating to the 3.x series."
- ↑ Barry Warsaw (2011-11-09). "PEP 404 – Python 2.8 Un-release Schedule" . Retrieved 2012-10-07.
- ↑ Editor. "Python 2.7 To Be Maintained Until 2020" . *i-programmer.info*. Retrieved December 27, 2016.
- ↑ Sam Ruby, 2to3 , September 1, 2007
- ↑ Nick Coghlan, Python 3.0 & 3.1 , June 29, 2012

26. -- Nick Coghlan, [Python 3 Q & A](#), June 29, 2012
27. ^ "PEP 3105 -- Make print a function" [python.org](#). Retrieved December 27, 2016.
28. ^ Rossum, Guido van van. "Python 3000 FAQ" [artima.com](#). Retrieved December 27, 2016.
29. ^ "PEP 3107 -- Function Annotations" [python.org](#). Retrieved December 27, 2016.
30. ^ PEP 3137 [python.org](#): Immutable Bytes and Mutable Buffer
31. ^ "Welcome to Python.org" [python.org](#). Retrieved December 27, 2016.
32. ^ Guido van Rossum (January 20, 2009). "The History of Python" [python.org](#). Retrieved March 3, 2018.



## External links [\[edit\]](#)

---

- [Guido Van Rossum blog on Python's History](#)[🔗](#)
-

This page was last edited on 31 March 2018, at 11:16.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#) [Mobile view](#)

