WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF

Languages
Español
فارسی
한국어
Русский
Српски / srpski
Türkçe
Edit links

# History of Python

From Wikipedia, the free encyclopedia

*Main article: Python (programming language)*

The programming language Python was conceived in the late 1980s,[1] and its implementation was started in December 1989[2] 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.[3] 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).[4][5] (However, van Rossum stepped down as leader on July 12, 2018.[6]). Python was named after the BBC TV show *Monty Python's Flying Circus*.[7]

Old Python logo, 1990s–2006

New Python logo, 2006–present

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.[8]

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008[9] after a long period of testing. Many of its major features have also been backported to the backwards-compatible, while by now unsupported, Python 2.6 and 2.7.[10]

**Contents** [hide]



## Early history [edit]

In February 1991, Van Rossum published the code (labeled version 0.9.0) to alt.sources.[11] 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".[1] Python's exception model also resembles Modula-3's, with the addition of an `else` clause.[3] In 1994 comp.lang.python, the primary discussion forum for Python, was formed, marking a milestone in the growth of Python's userbase.[1]

## 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".[12]

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 from where 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.[13]

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.[14] 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, emailing non-programmers is not an active concern.[15]

### BeOpen [edit]

In 2000, the Python core development team moved to BeOpen.com[16] to form the BeOpen PythonLabs team, under the direction of early Google alum Domenic Merenda.[17][18] 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.[8] 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.[19]

## Version 2 [edit]

Python 2.0, released October 2000,[8] 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.[8]

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.[19] The release included a change to the language specification to support nested scopes, like other statically scoped languages.[20] (The feature was turned off by default, and not required, until Python 2.2.)

Python 2.2 was released in December 2001;[21] a major innovation 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.[22] Also added were generators which were inspired by Icon.[23]

Python 2.5 was released in September 2006 [24] 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. [25]

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.[26][10] Similarly, Python 2.7 coincided with and included features from Python 3.1,[27] 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.[28] 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.[29] Python 2.7 support ended on January 1, 2020, along with code freeze of 2.7 development branch. A final release, 2.7.18, occurred on April 20, 2020, and included fixes for critical bugs and release blockers.[30] This marks the end-of-life of Python 2.[31]

## Version 3 [edit]

Python 3.0 (also called "Python 3000" or "Py3K") was released on December 3, 2008.[9] 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 could still follow object-oriented, structured, and functional programming paradigms, among others, 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.[32] 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.[10] This is no longer recommended; as of 2012 the preferred approach was to create a single code base that can run under both Python 2 and 3 using compatibility modules.[33]

### 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[34]
- 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 code that uses `reduce` is less readable than code that uses a for loop and accumulator variable)[35][36]
- Adding support for optional function annotations that can be used for informal type declarations or other purposes[37]
- 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[38]
- 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`. (In both Python 2 (2.2 onwards) and Python 3, a separate operator exists to provide the old behavior: `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.

## Table of versions [edit]

Releases before numbered versions:

- Implementation started - December, 1989[2]
- Internal releases at Centrum Wiskunde & Informatica - 1990[2]

| Version | Latest micro version | Release date | End of full support | End of security fixes |
|---|---|---|---|---|
| 0.9 | 0.9.9[2] | 1991-02-20[2] | 1993-07-29[a][2] | |
| 1.0 | 1.0.4[2] | 1994-01-26[2] | 1994-02-15[a][2] | |

| | | | | |
|---|---|---|---|---|
| 1.1 | 1.1.1[2] | 1994-10-11[2] | 1994-11-10[a][2] | |
| 1.2 | | 1995-04-13[2] | Unsupported | |
| 1.3 | | 1995-10-13[2] | Unsupported | |
| 1.4 | | 1996-10-25[2] | Unsupported | |
| 1.5 | 1.5.2[39] | 1998-01-03[2] | 1999-04-13[a][2] | |
| 1.6 | 1.6.1[39] | 2000-09-05[40] | 2000-09[a][39] | |
| 2.0 | 2.0.1[41] | 2000-10-16[42] | 2001-06-22[a][41] | |
| 2.1 | 2.1.3[41] | 2001-04-15[43] | 2002-04-09[a][41] | |
| 2.2 | 2.2.3[41] | 2001-12-21[44] | 2003-05-30[a][41] | |
| 2.3 | 2.3.7[41] | 2003-06-29[45] | 2008-03-11[a][41] | |
| 2.4 | 2.4.6[41] | 2004-11-30[46] | 2008-12-19[a][41] | |
| 2.5 | 2.5.6[41] | 2006-09-19[47] | 2011-05-26[a][41] | |
| 2.6 | 2.6.9[26] | 2008-10-01[26] | 2010-08-24[b][26] | 2013-10-29[26] |
| 2.7 | 2.7.18[31] | 2010-07-03[31] | 2020-01-01[c][31] | |
| 3.0 | 3.0.1[41] | 2008-12-03[26] | 2009-06-27[48] | |
| 3.1 | 3.1.5[49] | 2009-06-27[49] | 2011-06-12[50] | 2012-06[49] |
| 3.2 | 3.2.6[51] | 2011-02-20[51] | 2013-05-13[b][51] | 2016-02-20[51] |
| 3.3 | 3.3.7[52] | 2012-09-29[52] | 2014-03-08[b][52] | 2017-09-29[52] |
| 3.4 | 3.4.10[53] | 2014-03-16[53] | 2017-08-09[54] | 2019-03-18[a][53] |
| 3.5 | 3.5.10[55] | 2015-09-13[55] | 2017-08-08[56] | 2020-09-30[55] |
| 3.6 | *3.6.13*[57] | 2016-12-23[57] | 2018-12-24[b][57] | 2021-12[57] |
| 3.7 | *3.7.10*[58] | 2018-06-27[58] | 2020-06-27[b][58] | 2023-06[58] |
| 3.8 | *3.8.9*[59] | 2019-10-14[59] | 2021-05-03[59] | 2024-10[59] |
| 3.9 | *3.9.4*[60] | 2020-10-05[60] | **2022-05**[61] | **2025-10**[60][61] |
| 3.10 | | 2021-10-04[62] | 2023-05[62] | 2026-10[62] |

**Legend:** Old version | Older version, still maintained | **Latest version** | Latest preview version | Future release

*Italic* is the latest micro version of currently supported versions as of 2020-10-03.

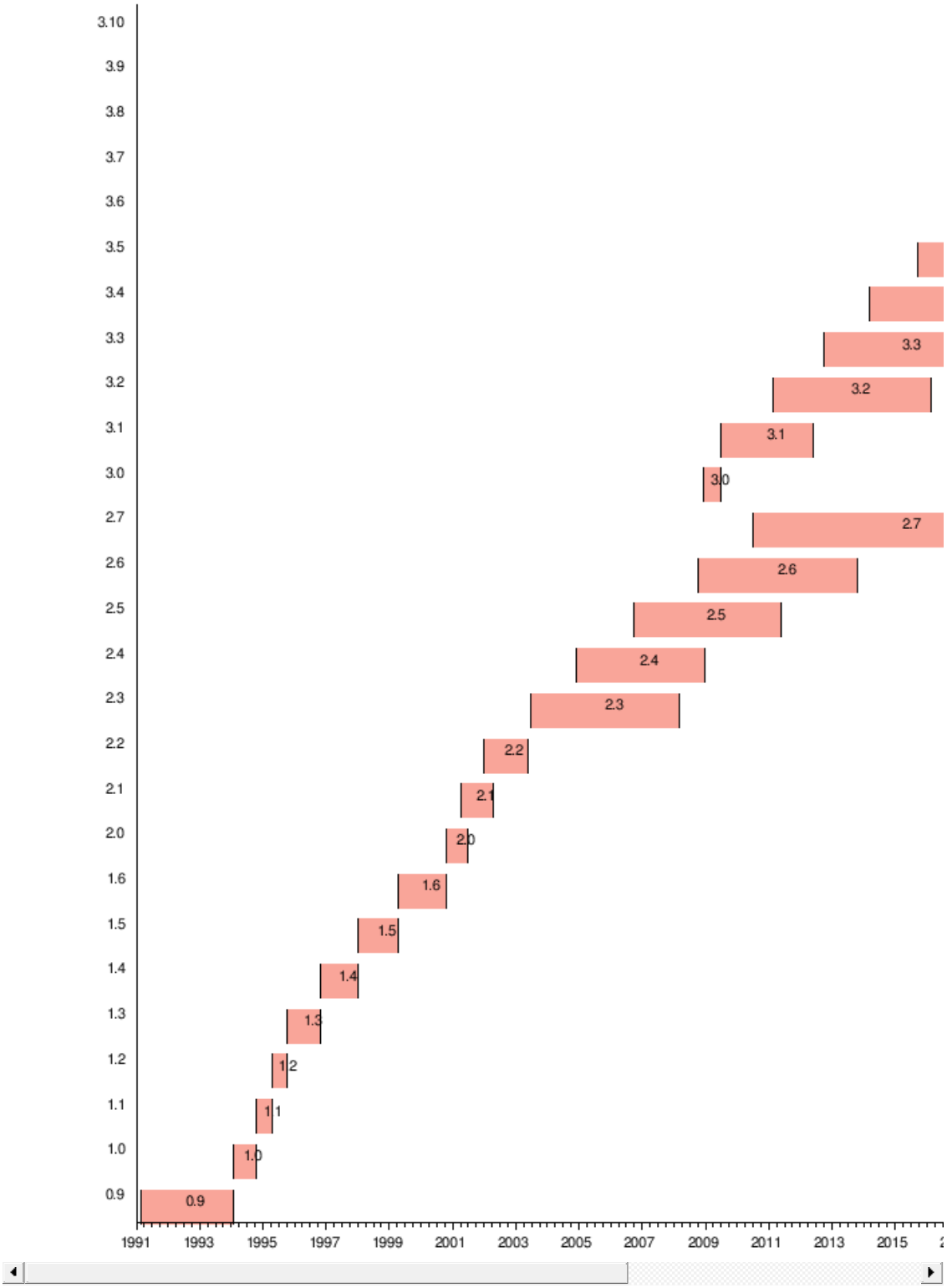
Table notes:

a. ^ *a b c d e f g h i j k l* Date of last micro release.

b. ^ *a b c d e* Date of last non security only release.

c. ^ Official support ended on 2020-01-01, but a final release of the code as it appeared on 2020-01-01 was released on 2020-04-20 as version 2.7.18.[31]

## Support [edit]




## See also [edit]

- History of software engineering

## References [edit]


1. ^ [a] [b] [c] "The Making of Python". Artima Developer. Retrieved March 22, 2007.
2. ^ [a] [b] [c] [d] [e] [f] [g] [h] [i] [j] [k] [l] [m] [n] [o] [p] [q] van Rossum, Guido (January 20, 2009). "A Brief Timeline of Python". Retrieved November 29, 2019.
3. ^ [a] [b] "Why was Python created in the first place?". Python FAQ. Retrieved March 22, 2007.
4. ^ van Rossum, Guido (July 31, 2008). "Origin of BDFL". Retrieved August 1, 2008.
5. ^ "Python Creator Scripts Inside Google". www.eweek.com. Retrieved May 13, 2008.
6. ^ Fairchild, Carlie (July 12, 2018). "Guido van Rossum Stepping Down from Role as Python's Benevolent Dictator For Life". *Linux Journal*. Retrieved July 12, 2018.
7. ^ "General Python FAQ — Python 3.8.3 documentation". *docs.python.org*.

8. ^ [a b c d] Kuchling, Andrew M.; Zadka, Moshe. "What's New in Python 2.0". Archived from the original on December 14, 2009. Retrieved March 22, 2007.
9. ^ [a b] "Welcome to Python.org". *python.org*. Retrieved December 27, 2016.
10. ^ [a b c] van Rossum, Guido (April 5, 2006). "PEP 3000 -- Python 3000". Retrieved December 27, 2016.
11. ^ "HISTORY". *Python source distribution*. Python Foundation. Retrieved November 23, 2017.
12. ^ van Rossum, Guido. "The fate of reduce() in Python 3000". Artima Developer. Retrieved March 22, 2007.
13. ^ "LJ #37: Python 1.4 Update". Archived from the original on May 1, 2007. Retrieved April 29, 2007.
14. ^ van Rossum, Guido. "Computer Programming for Everybody". Retrieved March 22, 2007.
15. ^ "Computer Programming for Everybody". Python Software Foundation. Archived from the original on March 29, 2007. Retrieved March 22, 2007.
16. ^ https://slashdot.org/story/00/05/30/1931239/python-development-team-moves-to-beopencom
17. ^ https://web.archive.org/web/20000815095541/http://www.beopen.com/company/team.html
18. ^ http://joeellsworth.com/resume/references/pybiz_beopen_partnership.pdf
19. ^ [a b] "History of the software". *Python Library Reference*. Archived from the original on March 29, 2007. Retrieved March 22, 2007.
20. ^ Hylton, Jeremy (November 1, 2000). "PEP 227 -- Statically Nested Scopes". Retrieved March 22, 2007.
21. ^ <https://www.python.org/download/releases/2.2/
22. ^ Kuchling, Andrew M. (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.
23. ^ Schemenauer, Neil; Peters, Tim; Hetland, Magnus (December 21, 2001). "PEP 255 -- Simple Generators". Retrieved September 5, 2008.
24. ^ "Python 2.5 Release". *Python.org*.
25. ^ "Highlights: Python 2.5". *Python.org*.
26. ^ [a b c d e f] Norwitz, Neal; Warsaw, Barry (June 29, 2006). "PEP 361 -- Python 2.6 and 3.0 Release Schedule". Retrieved November 29, 2019.
27. ^ Kuchling, Andrew M. (July 3, 2010). "What's New in Python 2.7". Retrieved October 7, 2012. "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."
28. ^ Warsaw, Barry (November 9, 2011). "PEP 404 -- Python 2.8 Un-release Schedule". Retrieved October 7, 2012.
29. ^ Gee, Sue (April 14, 2014). "Python 2.7 To Be Maintained Until 2020". *i-programmer.info*. Retrieved December 27, 2016.
30. ^ "Commits · python/cpython at 2.7".
31. ^ [a b c d e] Peterson, Benjamin (November 3, 2008). "PEP 373 -- Python 2.7 Release Schedule". Retrieved April 20, 2020.
32. ^ Ruby, Sam; 2to3, September 1, 2007
33. ^ Coghlan, Nick; Python 3 Q & A, June 29, 2012
34. ^ Brandl, Georg (November 19, 2007). "PEP 3105 -- Make print a function". Retrieved December 27, 2016.
35. ^ van Rossum, Guido. "Python 3000 FAQ". *artima.com*. Retrieved December 27, 2016.
36. ^ "The fate of reduce() in Python 3000". *www.artima.com*. Retrieved December 31, 2019.
37. ^ Winter, Collin; Lownds, Tony (December 2, 2006). "PEP 3107 -- Function Annotations". Retrieved December 27, 2016.
38. ^ van Rossum, Guido (September 26, 2007). "PEP 3137 -- Immutable Bytes and Mutable Buffer".
39. ^ [a b c] "Releases | Python.org". Retrieved November 29, 2019.
40. ^ Drake, Fred L., Jr. (July 25, 2000). "PEP 160 -- Python 1.6 Release Schedule". Retrieved November 29, 2019.
41. ^ [a b c d e f g h i j k l m] "Download Python | Python.org". Retrieved November 29, 2019.
42. ^ Hylton, Jeremy. "PEP 200 -- Python 2.0 Release Schedule". Retrieved November 29, 2019.
43. ^ Hylton, Jeremy (October 16, 2000). "PEP 226 -- Python 2.1 Release Schedule". Retrieved November 29, 2019.
44. ^ Warsaw, Barry; van Rossum, Guido (April 17, 2001). "PEP 251 -- Python 2.2 Release Schedule". Retrieved November 29, 2019.
45. ^ van Rossum, Guido (February 27, 2002). "PEP 283 -- Python 2.3 Release Schedule". Retrieved November 29, 2019.
46. ^ Warsaw, Barry; Hettinger, Raymond; Baxter, Anthony (July 29, 2003). "PEP 320 -- Python 2.4 Release Schedule". Retrieved November 29, 2019.
47. ^ Norwitz, Neal; van Rossum, Guido; Baxter, Anthony (February 7, 2006). "PEP 356 -- Python 2.5 Release Schedule". Retrieved November 29, 2019.
48. ^ "17. Development Cycle — Python Developer's Guide". Retrieved November 29, 2019.
49. ^ [a b c] Peterson, Benjamin (February 8, 2009). "PEP 375 -- Python 3.1 Release Schedule". Retrieved November 29, 2019.
50. ^ Peterson, Benjamin (June 12, 2011). "[RELEASED] Python 3.1.4". *python-announce* (Mailing list). Retrieved November 29, 2019.
51. ^ [a b c d] Brandl, Georg (December 30, 2009). "PEP 392 -- Python 3.2 Release Schedule". Retrieved November 29, 2019.

52. ^ a b c d Brandl, Georg (March 23, 2011). "PEP 398 -- Python 3.3 Release Schedule". Retrieved November 29, 2019.
53. ^ a b c Hastings, Larry (October 17, 2012). "PEP 429 -- Python 3.4 Release Schedule". Retrieved November 29, 2019.
54. ^ Hastings, Larry (August 9, 2017). "[RELEASED] Python 3.4.7 is now available". *python-announce* (Mailing list). Retrieved November 29, 2019.
55. ^ a b c Hastings, Larry (September 22, 2014). "PEP 478 -- Python 3.5 Release Schedule". Retrieved March 17, 2020.
56. ^ Hastings, Larry (August 8, 2017). "[RELEASED] Python 3.5.4 is now available". *python-announce* (Mailing list). Retrieved November 29, 2019.
57. ^ a b c d Deily, Ned (May 30, 2015). "PEP 494 -- Python 3.6 Release Schedule". Retrieved April 20, 2020.
58. ^ a b c d Deily, Ned (December 23, 2016). "PEP 537 -- Python 3.7 Release Schedule". Retrieved April 20, 2020.
59. ^ a b c d Langa, Łukasz (January 27, 2018). "PEP 569 -- Python 3.8 Release Schedule". Retrieved April 20, 2020.
60. ^ a b c Langa, Łukasz (October 13, 2020). "PEP 596 -- Python 3.9 Release Schedule". Retrieved October 13, 2020.
61. ^ a b Langa, Łukasz (June 4, 2019). "PEP 602 -- Annual Release Cycle for Python". Retrieved November 29, 2019.
62. ^ a b c Pablo, Pablo (May 25, 2020). "PEP 619 -- Python 3.10 Release Schedule". Retrieved May 25, 2020.


## External links [edit]

- Guido Van Rossum blog on Python's History

Categories: History of software | Python (programming language) | Software version histories