

Python is a [high-level](#), [general-purpose programming language](#). Its design philosophy emphasizes [code readability](#) with the use of significant indentation via the [off-side rule](#).^[34]

Python is [dynamically typed](#) and [garbage-collected](#). It supports multiple [programming paradigms](#), including [structured](#) (particularly [procedural](#)), [object-oriented](#) and [functional programming](#). It is often described as a "batteries included" language due to its comprehensive [standard library](#).^[35]^[36]

[Guido van Rossum](#) began working on Python in the late 1980s as a successor to the [ABC programming language](#) and first released it in 1991 as Python 0.9.0.^[37] Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely [backward-compatible](#) with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.^[38]

Python consistently ranks as one of the most popular programming languages.^[39]^[40]^[41]^[42]

History

The designer of Python, [Guido van Rossum](#), at [OSCON](#) 2006

Main article: [History of Python](#)

Python was conceived in the late 1980s^[43] by [Guido van Rossum](#) at [Centrum Wiskunde & Informatica](#) (CWI) in the [Netherlands](#) as a successor to the [ABC programming language](#), which was inspired by [SETL](#),^[44] capable of [exception handling](#) and interfacing with the [Amoeba](#) operating system.^[13] Its implementation began in December 1989.^[45] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "[benevolent dictator for life](#)", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.^[46] In January 2019, active Python core developers elected a five-member Steering Council to lead the project.^[47]^[48]

Python 2.0 was released on 16 October 2000, with many major new features such as [list comprehensions](#), [cycle-detecting](#) garbage collection, [reference counting](#), and [Unicode](#) support.^[49] Python 3.0, released on 3 December 2008, with many of its major features [backported](#) to Python 2.6.x^[50] and 2.7.x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.^[51]

Python 2.7's [end-of-life](#) was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.^[52]^[53] No further security patches or other improvements will be released for it.^[54]^[55] Currently only 3.7 and later are supported. In 2021, Python 3.9.2 and 3.8.8 were expedited^[56] as all versions of Python (including 2.7^[57]) had security issues leading to possible [remote code execution](#)^[58] and [web cache poisoning](#).^[59]

In 2022, Python 3.10.4 and 3.9.12 were expedited^[60] and 3.8.13, and 3.7.13, because of many security issues.^[61] When Python 3.9.13 was released in May 2022, it was announced that the 3.9 series (joining the older series 3.8 and 3.7) would only receive security fixes in the future.^[62] On September 7, 2022, four new releases were made due to a potential [denial-of-service attack](#): 3.10.7, 3.9.14, 3.8.14, and 3.7.14.^[63]^[64]

As of November 2022, Python 3.11 is the stable release. Notable changes from 3.10 include increased program execution speed and improved error reporting.^[65]

Design philosophy and features

Python is a [multi-paradigm programming language](#). [Object-oriented programming](#) and [structured programming](#) are fully supported, and many of their features support functional programming and [aspect-oriented programming](#) (including [metaprogramming](#)^[66] and [metaobjects](#)).^[67] Many other paradigms are supported via extensions, including [design by contract](#)^[68]^[69] and [logic programming](#).^[70]

Python uses [dynamic typing](#) and a combination of [reference counting](#) and a cycle-detecting garbage collector for [memory management](#).^[71] It uses dynamic [name resolution](#) ([late binding](#)), which binds method and variable names during program execution.

Its design offers some support for functional programming in the [Lisp](#) tradition. It has `filter`, `map` and `reduce` functions; [list comprehensions](#), [dictionaries](#), sets, and [generator](#) expressions.^[72] The standard library has two modules (`itertools` and `functools`) that implement functional tools borrowed from [Haskell](#) and [Standard ML](#).^[73]

Its core philosophy is summarized in the document *The Zen of Python* (PEP 20), which includes [aphorisms](#) such as:^[74]

- ⑩ Beautiful is better than ugly.
- ⑩ Explicit is better than implicit.
- ⑩ Simple is better than complex.
- ⑩ Complex is better than complicated.
- ⑩ Readability counts.