Python (programming language)

From Wikipedia, the free encyclopedia

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.[32]

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.[33][

History

Main article: History of Python

Python was conceived in the late 1980s[41] 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,[42] capable of exception handling (from the start plus new capabilities in Python 3.11) and interfacing with the Amoeba operating system.[12] Its implementation began in December 1989.[43] 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.[44] In January 2019, active Python core developers elected a five-member Steering Council to lead the project.[45][46]

Python 2.0 was released on 16 October 2000, with many major new features.[47] Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x[48] and 2.7.x. Releases of Python 3 include the `2to3` utility, which automates the translation of Python 2 code to Python 3.[49]

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.[50][51] No further security patches or other improvements will be released for it.[52][53] Currently only 3.7 and later are supported. In 2021, Python 3.9.2 and 3.8.8 were expedited[54] as all versions of Python (including 2.7[55]) had security issues leading to possible remote code execution[56] and web cache poisoning.[57]

In 2022, Python 3.10.4 and 3.9.12 were expedited[58] and 3.8.13, and 3.7.13, because of many security issues.[59] 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) will only receive security fixes going forward.[60] 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.[61][62]

As of November 2022, Python 3.11.0 is the current stable release and among the notable changes from 3.10 are that it is 10–60% faster and significantly improved error reporting.[63]

Python 3.12 (alpha 2) has improved error messages.

Removals from Python

The deprecated `smtpd` module has been removed from Python 3.12 (alpha). And a number of other old, broken and deprecated functions (e.g. from `unittest` module), classes and methods have been removed. The deprecated `wstr` and `wstr_` length members of the C implementation of Unicode objects were removed,[64] to make UTF-8 the default in later Python versions.

Historically, Python 3 also made changes from Python 2, e.g. changed the division operator.

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[65] and metaobjects).[66] Many other paradigms are supported via extensions, including design by contract[67][68] and logic programming.[69]

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management.[70] 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.[71] The standard library has two modules (`itertools` and `functools`) that implement functional tools borrowed from Haskell and Standard ML.[72]

Its core philosophy is summarized in the document *The Zen of Python* (*PEP 20*), which includes aphorisms such as:[73]

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

Rather than building all of its functionality into its core, Python was designed to be highly extensible via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.[41]

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" philosophy.[73] Alex Martelli, a Fellow at the Python Software Foundation and Python book author, wrote: "To describe something as 'clever' is *not* considered a compliment in the Python culture."[74]

Python's developers strive to avoid premature optimization and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity.[75] When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C; or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

Python's developers aim for it to be fun to use. This is reflected in its name—a tribute to the British comedy group Monty Python[76]—and in occasionally playful approaches to tutorials and reference materials, such as the use of the terms "spam", and "eggs" (a reference to a Monty Python sketch) in examples, instead of the often-used "foo", and "bar".[77][78]

A common neologism in the Python community is *pythonic*, which has a wide range of meanings related to program style. "Pythonic" code may use Python idioms well, be natural or show fluency in the language, or conform with Python's minimalist philosophy and emphasis on readability. Code that is difficult to understand or reads like a rough transcription from another programming language is called *unpythonic*.[79][80]

Python users and admirers, especially those considered knowledgeable or experienced, are often referred to as *Pythonistas*.[81][82]

Performance

Performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.[160] Python's performance compared to other programming languages is also benchmarked by The Computer Language Benchmarks Game.[161]

All this information from "Python (programming language) - Wikipedia"