Python (programming language)

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/Python_(programming_language)#mw-head)[Jump to search](https://en.wikipedia.org/wiki/Python_(programming_language)#searchInput)

|  |  |
| --- | --- |
| Python | |
|  | |
| [Paradigm](https://en.wikipedia.org/wiki/Programming_paradigm) | [Multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language): [functional](https://en.wikipedia.org/wiki/Functional_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [structured](https://en.wikipedia.org/wiki/Structured_programming), [reflective](https://en.wikipedia.org/wiki/Reflective_programming) |
| [Designed by](https://en.wikipedia.org/wiki/Software_design) | [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) |
| [Developer](https://en.wikipedia.org/wiki/Software_developer) | [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation) |
| First appeared | 1991; 30 years ago[[1]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-guttag-1) |
|  | |
| [Stable release](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 3.9.1[[2]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-wikidata-ba26ae53d9121f3dd8ba5c1091f58b764835bd74-v3-2) [Edit this on Wikidata](https://www.wikidata.org/wiki/Q28865?uselang=en#P348) / 7 December 2020; 44 days ago |
| [Preview release](https://en.wikipedia.org/wiki/Software_release_life_cycle#BETA) | 3.10.0a4[[3]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-wikidata-e4828aaecf90577d574c483675454ffe8394560b-v3-3) [Edit this on Wikidata](https://www.wikidata.org/wiki/Q28865?uselang=en#P348) / 4 January 2021; 16 days ago |
| [Typing discipline](https://en.wikipedia.org/wiki/Type_system) | [Duck](https://en.wikipedia.org/wiki/Duck_typing), [dynamic](https://en.wikipedia.org/wiki/Dynamic_typing), [gradual](https://en.wikipedia.org/wiki/Gradual_typing) (since 3.5)[[4]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-4) |
| [OS](https://en.wikipedia.org/wiki/Operating_system) | [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS), [Windows 8](https://en.wikipedia.org/wiki/Windows_8) and later and more |
| [License](https://en.wikipedia.org/wiki/Software_license) | [Python Software Foundation License](https://en.wikipedia.org/wiki/Python_Software_Foundation_License) |
| [Filename extensions](https://en.wikipedia.org/wiki/Filename_extension) | .py, .pyi, .pyc, .pyd, .pyo (prior to 3.5),[[5]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-5) .pyw, .pyz (since 3.5)[[6]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-6) |
| Website | [www.python.org](https://www.python.org/) |
| Major [implementations](https://en.wikipedia.org/wiki/Programming_language_implementation) | |
| [CPython](https://en.wikipedia.org/wiki/CPython), [PyPy](https://en.wikipedia.org/wiki/PyPy" \o "PyPy), [Stackless Python](https://en.wikipedia.org/wiki/Stackless_Python" \o "Stackless Python), [MicroPython](https://en.wikipedia.org/wiki/MicroPython" \o "MicroPython), [CircuitPython](https://en.wikipedia.org/wiki/CircuitPython" \o "CircuitPython), [IronPython](https://en.wikipedia.org/wiki/IronPython" \o "IronPython), [Jython](https://en.wikipedia.org/wiki/Jython" \o "Jython), RustPython | |
| [Dialects](https://en.wikipedia.org/wiki/Programming_language#Dialects,_flavors_and_implementations) | |
| [Cython](https://en.wikipedia.org/wiki/Cython), [RPython](https://en.wikipedia.org/wiki/PyPy" \l "RPython" \o "PyPy), [Starlark](https://en.wikipedia.org/wiki/Bazel_(software)" \o "Bazel (software))[[7]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-7) | |
| Influenced by | |
| [ABC](https://en.wikipedia.org/wiki/ABC_(programming_language)),[[8]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-faq-created-8) [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language)),[[9]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-9) [ALGOL 68](https://en.wikipedia.org/wiki/ALGOL_68),[[10]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-98-interview-10) [APL](https://en.wikipedia.org/wiki/APL_(programming_language)),[[11]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-python.org-11) [C](https://en.wikipedia.org/wiki/C_(programming_language)),[[12]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-1-12) [C++](https://en.wikipedia.org/wiki/C%2B%2B),[[13]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-classmix-13) [CLU](https://en.wikipedia.org/wiki/CLU_(programming_language)),[[14]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-effbot-call-by-object-14) [Dylan](https://en.wikipedia.org/wiki/Dylan_(programming_language)),[[15]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-2-15) [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)),[[16]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-3-16) [Icon](https://en.wikipedia.org/wiki/Icon_(programming_language)),[[17]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-4-17) [Java](https://en.wikipedia.org/wiki/Java_(programming_language)),[[18]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-5-18) [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)),[[19]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-6-19) [Modula-3](https://en.wikipedia.org/wiki/Modula-3),[[13]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-classmix-13) [Perl](https://en.wikipedia.org/wiki/Perl), [Standard ML](https://en.wikipedia.org/wiki/Standard_ML)[[11]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-python.org-11) | |
| Influenced | |
| [Apache Groovy](https://en.wikipedia.org/wiki/Apache_Groovy), [CoffeeScript](https://en.wikipedia.org/wiki/CoffeeScript),[[20]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-20) [D](https://en.wikipedia.org/wiki/D_(programming_language)), [F#](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)), [Genie](https://en.wikipedia.org/wiki/Genie_(programming_language)),[[21]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-21) [Go](https://en.wikipedia.org/wiki/Go_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript),[[22]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-22)[[23]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-23) [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)),[[24]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Julia-24) [Nim](https://en.wikipedia.org/wiki/Nim_(programming_language)), Ring,[[25]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-The_Ring_programming_language_and_other_languages-25) [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)),[[26]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-bini-26) [Swift](https://en.wikipedia.org/wiki/Swift_(programming_language))[[27]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-lattner2014-27) | |
| * [Python Programming](https://en.wikibooks.org/wiki/Python_Programming) at Wikibooks | |

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) and [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.[[28]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-7-28)

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).[[29]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-About-29)

Python was created in the late 1980s, and first released in 1991, by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)). Python 2.0, released in 2000, introduced new features, such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), and a garbage collection system with [reference counting](https://en.wikipedia.org/wiki/Reference_counting), and was discontinued with version 2.7 in 2020.[[30]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-30) Python 3.0, released in 2008, was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) and much Python 2 code does not run unmodified on Python 3. With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)), only Python 3.6.x[[31]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-31) and later are supported, with older versions still supporting e.g. [Windows 7](https://en.wikipedia.org/wiki/Windows_7) (and old installers not restricted to 64-bit Windows).

Python [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) are supported for mainstream [operating systems](https://en.wikipedia.org/wiki/Operating_system) and available for a few more (and in the past supported many more). A global community of programmers develops and maintains [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython), a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software)[[32]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-32) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation). A non-profit organization, the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation), manages and directs resources for Python and CPython development.

As of December 2020 Python ranked third in [TIOBE](https://en.wikipedia.org/wiki/TIOBE_index)’s index of most popular programming languages, behind [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [Java](https://en.wikipedia.org/wiki/Java_(programming_language)).[[33]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-33)



Contents

* [1History](https://en.wikipedia.org/wiki/Python_(programming_language)#History)
* [2Design philosophy and features](https://en.wikipedia.org/wiki/Python_(programming_language)#Design_philosophy_and_features)
* [3Syntax and semantics](https://en.wikipedia.org/wiki/Python_(programming_language)#Syntax_and_semantics)
  + [3.1Indentation](https://en.wikipedia.org/wiki/Python_(programming_language)#Indentation)
  + [3.2Statements and control flow](https://en.wikipedia.org/wiki/Python_(programming_language)#Statements_and_control_flow)
  + [3.3Expressions](https://en.wikipedia.org/wiki/Python_(programming_language)#Expressions)
  + [3.4Methods](https://en.wikipedia.org/wiki/Python_(programming_language)#Methods)
  + [3.5Typing](https://en.wikipedia.org/wiki/Python_(programming_language)#Typing)
  + [3.6Arithmetic operations](https://en.wikipedia.org/wiki/Python_(programming_language)#Arithmetic_operations)
* [4Programming examples](https://en.wikipedia.org/wiki/Python_(programming_language)#Programming_examples)
* [5Libraries](https://en.wikipedia.org/wiki/Python_(programming_language)#Libraries)
* [6Development environments](https://en.wikipedia.org/wiki/Python_(programming_language)#Development_environments)
* [7Implementations](https://en.wikipedia.org/wiki/Python_(programming_language)#Implementations)
  + [7.1Reference implementation](https://en.wikipedia.org/wiki/Python_(programming_language)#Reference_implementation)
  + [7.2Other implementations](https://en.wikipedia.org/wiki/Python_(programming_language)#Other_implementations)
  + [7.3Unsupported implementations](https://en.wikipedia.org/wiki/Python_(programming_language)#Unsupported_implementations)
  + [7.4Cross-compilers to other languages](https://en.wikipedia.org/wiki/Python_(programming_language)#Cross-compilers_to_other_languages)
  + [7.5Performance](https://en.wikipedia.org/wiki/Python_(programming_language)#Performance)
* [8Development](https://en.wikipedia.org/wiki/Python_(programming_language)#Development)
* [9API documentation generators](https://en.wikipedia.org/wiki/Python_(programming_language)#API_documentation_generators)
* [10Naming](https://en.wikipedia.org/wiki/Python_(programming_language)#Naming)
* [11Uses](https://en.wikipedia.org/wiki/Python_(programming_language)#Uses)
* [12Languages influenced by Python](https://en.wikipedia.org/wiki/Python_(programming_language)#Languages_influenced_by_Python)
* [13See also](https://en.wikipedia.org/wiki/Python_(programming_language)#See_also)
* [14References](https://en.wikipedia.org/wiki/Python_(programming_language)#References)
  + [14.1Sources](https://en.wikipedia.org/wiki/Python_(programming_language)#Sources)
* [15Further reading](https://en.wikipedia.org/wiki/Python_(programming_language)#Further_reading)
* [16External links](https://en.wikipedia.org/wiki/Python_(programming_language)#External_links)

History[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=1)]

[](https://en.wikipedia.org/wiki/File:Guido_van_Rossum_OSCON_2006_cropped.png)

[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) at OSCON 2006

*Main article:*[*History of Python*](https://en.wikipedia.org/wiki/History_of_Python)

Python was conceived in the late 1980s[[34]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-venners-interview-pt-1-34) by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) at [Centrum Wiskunde & Informatica](https://en.wikipedia.org/wiki/Centrum_Wiskunde_%26_Informatica) (CWI) in the [Netherlands](https://en.wikipedia.org/wiki/Netherlands) as a successor to [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)), which was inspired by [SETL](https://en.wikipedia.org/wiki/SETL),[[35]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-12-35) capable of [exception handling](https://en.wikipedia.org/wiki/Exception_handling) and interfacing with the [Amoeba](https://en.wikipedia.org/wiki/Amoeba_(operating_system)) operating system.[[8]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-faq-created-8) Its implementation began in December 1989.[[36]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-timeline-of-python-36) 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*](https://en.wikipedia.org/wiki/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.[[37]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-lj-bdfl-resignation-37) He now shares his leadership as a member of a five-person steering council.[[38]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-38)[[39]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-39)[[40]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-40) In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project.[[41]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-41) Guido van Rossum has since then withdrawn his nomination for the 2020 Steering council.[[42]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-42)

Python 2.0 was released on 16 October 2000 with many major new features, including a [cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) [garbage collector](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) and support for [Unicode](https://en.wikipedia.org/wiki/Unicode).[[43]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-newin-2.0-43)

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility).[[44]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-3.0-release-44) Many of its major features were [backported](https://en.wikipedia.org/wiki/Backporting) to Python 2.6.x[[45]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep-3000-45) and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.[[46]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-46)

Python 2.7's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)) date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.[[47]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-47)[[48]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-48) No more security patches or other improvements will be released for it.[[49]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-49)[[50]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-50) With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)), only Python 3.6.x[[51]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-51) and later are supported.

Design philosophy and features[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=2)]

Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by [metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming)[[52]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-13-52) and [metaobjects](https://en.wikipedia.org/wiki/Metaobject) (magic methods)).[[53]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-14-53) Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract)[[54]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-15-54)[[55]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-16-55) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).[[56]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-17-56)

Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management).[[57]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Reference_counting-57) It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Python's design offers some support for [functional programming](https://en.wikipedia.org/wiki/Functional_programming) in the [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) tradition. It has filter, map, and reduce functions; [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [dictionaries](https://en.wikipedia.org/wiki/Associative_array), sets, and [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)) expressions.[[58]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-59-58) The standard library has two modules (itertools and functools) that implement functional tools borrowed from [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) and [Standard ML](https://en.wikipedia.org/wiki/Standard_ML).[[59]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-18-59)

The language's core philosophy is summarized in the document *The*[*Zen of Python*](https://en.wikipedia.org/wiki/Zen_of_Python) (*PEP 20*), which includes [aphorisms](https://en.wikipedia.org/wiki/Aphorism) such as:[[60]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PEP20-60)

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

Rather than having all of its functionality built into its core, Python was designed to be highly [extensible](https://en.wikipedia.org/wiki/Extensibility). 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](https://en.wikipedia.org/wiki/ABC_(programming_language)), which espoused the opposite approach.[[34]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-venners-interview-pt-1-34)

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to [Perl](https://en.wikipedia.org/wiki/Perl)'s "[there is more than one way to do it](https://en.wikipedia.org/wiki/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" design philosophy.[[60]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PEP20-60) [Alex Martelli](https://en.wikipedia.org/wiki/Alex_Martelli), a [Fellow](https://en.wikipedia.org/wiki/Fellow) at the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation) and Python book author, writes that "To describe something as 'clever' is *not* considered a compliment in the Python culture."[[61]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-19-61)

Python's developers strive to avoid [premature optimization](https://en.wikipedia.org/wiki/Premature_optimization), and reject patches to non-critical parts of the [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython) reference implementation that would offer marginal increases in speed at the cost of clarity.[[62]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-20-62) When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use [PyPy](https://en.wikipedia.org/wiki/PyPy" \o "PyPy), a [just-in-time compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation). [Cython](https://en.wikipedia.org/wiki/Cython" \o "Cython) is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group [Monty Python](https://en.wikipedia.org/wiki/Monty_Python)[[63]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-24-63)—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a [famous Monty Python sketch](https://en.wikipedia.org/wiki/Spam_(Monty_Python))) instead of the standard [foo and bar](https://en.wikipedia.org/wiki/Foobar).[[64]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-64)[[65]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-65)

A common [neologism](https://en.wikipedia.org/wiki/Neologism) in the Python community is *pythonic*, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called *unpythonic*.[[66]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-66)[[67]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-67)

Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as *Pythonistas*.[[68]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-27-68)[[69]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-25-69)

Syntax and semantics[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=3)]

*Main article:*[*Python syntax and semantics*](https://en.wikipedia.org/wiki/Python_syntax_and_semantics)

|  |  |
| --- | --- |
|  | It has been suggested that this article be [merged](https://en.wikipedia.org/wiki/Wikipedia:Merging) with [*Python syntax and semantics*](https://en.wikipedia.org/wiki/Python_syntax_and_semantics). ([Discuss](https://en.wikipedia.org/wiki/Talk:Python_(programming_language)))*Proposed since January 2021.* |

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than [C](https://en.wikipedia.org/wiki/C_(programming_language)) or [Pascal](https://en.wikipedia.org/wiki/Pascal_(programming_language)).[[70]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-52-70)

Indentation[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=4)]

*Main article:*[*Python syntax and semantics § Indentation*](https://en.wikipedia.org/wiki/Python_syntax_and_semantics#Indentation)

Python uses [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation, rather than [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) or keywords, to delimit [blocks](https://en.wikipedia.org/wiki/Block_(programming)). An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.[[71]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-53-71) Thus, the program's visual structure accurately represents the program's semantic structure.[[1]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-guttag-1) This feature is sometimes termed the [off-side rule](https://en.wikipedia.org/wiki/Off-side_rule), which some other languages share, but in most languages indentation doesn't have any semantic meaning.

Statements and control flow[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=5)]

Python's [statements](https://en.wikipedia.org/wiki/Statement_(computer_science)) include (among others):

* The [assignment](https://en.wikipedia.org/wiki/Assignment_(computer_science)) statement (token '=', the equals sign).
* The [if](https://en.wikipedia.org/wiki/If-then-else) statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
* The [for](https://en.wikipedia.org/wiki/Foreach#Python) statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
* The [while](https://en.wikipedia.org/wiki/While_loop#Python) statement, which executes a block of code as long as its condition is true.
* The [try](https://en.wikipedia.org/wiki/Exception_handling_syntax#Python) statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
* The raise statement, used to raise a specified exception or re-raise a caught exception.
* The class statement, which executes a block of code and attaches its local namespace to a [class](https://en.wikipedia.org/wiki/Class_(computer_science)), for use in [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming).
* The def statement, which defines a [function](https://en.wikipedia.org/wiki/Function_(computing)) or [method](https://en.wikipedia.org/wiki/Method_(computing)).
* The with statement, from Python 2.5 released in September 2006,[[72]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-72) which encloses a code block within a context manager (for example, acquiring a [lock](https://en.wikipedia.org/wiki/Lock_(computer_science)) before the block of code is run and releasing the lock afterwards, or opening a [file](https://en.wikipedia.org/wiki/Computer_file) and then closing it), allowing [Resource Acquisition Is Initialization](https://en.wikipedia.org/wiki/Resource_acquisition_is_initialization) (RAII)-like behavior and replaces a common try/finally idiom.[[73]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-73)
* The [break](https://en.wikipedia.org/wiki/Break_statement) statement, exits from the loop.
* The continue statement, skips this iteration and continues with the next item.
* The del statement, removes a variable, which means the reference from the name to the value is deleted and trying to use that variable cause an error. Deleted variable can be reassigned to later as normal.
* The pass statement, which serves as a [NOP](https://en.wikipedia.org/wiki/NOP_(code)). It is syntactically needed to create an empty code block.
* The [assert](https://en.wikipedia.org/wiki/Assertion_(programming)) statement, used during debugging to check for conditions that ought to apply.
* The yield statement, which returns a value from a [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)#Python) function. From Python 2.5, yield is also an operator. This form is used to implement [coroutines](https://en.wikipedia.org/wiki/Coroutine).
* The return statement, used to return a value from a function.
* The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as <alias>] or from <module name> import \* or from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>], ....
* The print statement was changed to the print() function in Python 3.

The assignment statement ('=') operates by binding a name as a [reference](https://en.wikipedia.org/wiki/Pointer_(computer_programming)) to a separate, dynamically allocated [object](https://en.wikipedia.org/wiki/Object_(computer_science)). Since the name's storage location doesn't *contain* the indicated value, it is improper to call it a *variable*. Names may be subsequently rebound at any time to any object. Since a name is a generic reference holder it is unreasonable to associate a fixed [data type](https://en.wikipedia.org/wiki/Type_system) with it. However at a given time, a name will be bound to *some* object, which will have a type. This association is referred to as [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_type).

Python does not support [tail call](https://en.wikipedia.org/wiki/Tail_call) optimization or [first-class continuations](https://en.wikipedia.org/wiki/First-class_continuations), and, according to Guido van Rossum, it never will.[[74]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-55-74)[[75]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-56-75) However, better support for [coroutine](https://en.wikipedia.org/wiki/Coroutine)-like functionality is provided in 2.5, by extending Python's [generators](https://en.wikipedia.org/wiki/Generator_(computer_programming)).[[76]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-57-76) Before 2.5, generators were [lazy](https://en.wikipedia.org/wiki/Lazy_evaluation) [iterators](https://en.wikipedia.org/wiki/Iterator); information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.[[77]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-58-77)

Expressions[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=6)]

Some Python [expressions](https://en.wikipedia.org/wiki/Expression_(computer_science)) are similar to languages such as [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), while some are not:

* Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division (or integer division) // and floating point/division.[[78]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-78) Python also added the \*\* operator for exponentiation.
* From Python 3.5, the new @ infix operator was introduced. It is intended to be used by libraries such as [NumPy](https://en.wikipedia.org/wiki/NumPy) for [matrix multiplication](https://en.wikipedia.org/wiki/Matrix_multiplication).[[79]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PEP465-79)[[80]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Python3.5Changelog-80)
* From Python 3.8, the syntax :=, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.[[81]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Python3.8Changelog-81)
* In Python, == compares by value, versus Java, which compares numerics by value[[82]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-82) and objects by reference.[[83]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-83) (Value comparisons in Java on objects can be performed with the equals() method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example a <= b <= c.
* Python uses the words and, or, not for its boolean operators rather than the symbolic &&, ||, ! used in Java and C.
* Python has a type of expression termed a [*list comprehension*](https://en.wikipedia.org/wiki/List_comprehension#Python). Python 2.4 extended list comprehensions into a more general expression termed a [*generator*](https://en.wikipedia.org/wiki/Generator_(computer_programming))*expression*.[[58]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-59-58)
* [Anonymous functions](https://en.wikipedia.org/wiki/Anonymous_function) are implemented using [lambda expressions](https://en.wikipedia.org/wiki/Lambda_(programming)); however, these are limited in that the body can only be one expression.
* Conditional expressions in Python are written as x if c else y[[84]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-60-84) (different in order of operands from the [c ? x : y](https://en.wikipedia.org/wiki/%3F:) operator common to many other languages).
* Python makes a distinction between [lists](https://en.wikipedia.org/wiki/List_(computer_science)) and [tuples](https://en.wikipedia.org/wiki/Tuple). Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be [immutable](https://en.wikipedia.org/wiki/Immutable) in Python). Tuples are written as (1, 2, 3), are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The + operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable t initially equal to (1, 2, 3), executing t = t + (4, 5) first evaluates t + (4, 5), which yields (1, 2, 3, 4, 5), which is then assigned back to t, thereby effectively "modifying the contents" of t, while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.[[85]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-85)
* Python features *sequence unpacking* wherein multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in an identical manner to that forming tuple literals and, as a whole, are put on the left-hand side of the equal sign in an assignment statement. The statement expects an *iterable* object on the right-hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through and will iterate through it, assigning each of the produced values to the corresponding expression on the left.[[86]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-86)
* Python has a "string format" operator %. This functions analogously to [printf](https://en.wikipedia.org/wiki/Printf_format_string" \o "Printf format string) format strings in [C](https://en.wikipedia.org/wiki/C_(programming_language)), e.g. "spam=%s eggs=%d" % ("blah", 2) evaluates to "spam=blah eggs=2". In Python 3 and 2.6+, this was supplemented by the format() method of the str class, e.g. "spam={0} eggs={1}".format("blah", 2). Python 3.6 added "f-strings": blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'.[[87]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep-0498-87)
* Strings in Python can be [concatenated](https://en.wikipedia.org/wiki/Concatenation), by "adding" them (same operator as for adding integers and floats). E.g. "spam" + "eggs" returns "spameggs". Even if your strings contain numbers, they are still added as strings rather than integers. E.g. "2" + "2" returns "22".
* Python has various kinds of [string literals](https://en.wikipedia.org/wiki/String_literal):
  + Strings delimited by single or double quote marks. Unlike in [Unix shells](https://en.wikipedia.org/wiki/Unix_shell), [Perl](https://en.wikipedia.org/wiki/Perl) and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (\) as an [escape character](https://en.wikipedia.org/wiki/Escape_character). [String interpolation](https://en.wikipedia.org/wiki/String_interpolation) became available in Python 3.6 as "formatted string literals".[[87]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep-0498-87)
  + Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like [here documents](https://en.wikipedia.org/wiki/Here_document) in shells, Perl and [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)).
  + [Raw string](https://en.wikipedia.org/wiki/Raw_string) varieties, denoted by prefixing the string literal with an r. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as [regular expressions](https://en.wikipedia.org/wiki/Regular_expression) and [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows)-style paths. Compare "@-quoting" in [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)).
* Python has [array index](https://en.wikipedia.org/wiki/Array_index) and [array slicing](https://en.wikipedia.org/wiki/Array_slicing) expressions on lists, denoted as a[key], a[start:stop] or a[start:stop:step]. Indexes are [zero-based](https://en.wikipedia.org/wiki/Zero-based), and negative indexes are relative to the end. Slices take elements from the *start* index up to, but not including, the *stop* index. The third slice parameter, called *step* or *stride*, allows elements to be skipped and reversed. Slice indexes may be omitted, for example a[:] returns a copy of the entire list. Each element of a slice is a [shallow copy](https://en.wikipedia.org/wiki/Shallow_copy).

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as [Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp), [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)), or [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)). This leads to duplicating some functionality. For example:

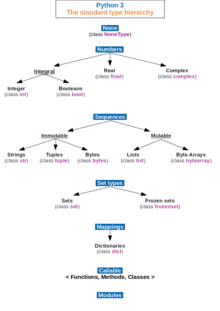
* [List comprehensions](https://en.wikipedia.org/wiki/List_comprehensions) vs. for-loops
* [Conditional](https://en.wikipedia.org/wiki/Conditional_(programming)) expressions vs. if blocks
* The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or [lambda expressions](https://en.wikipedia.org/wiki/Lambda_(programming)), all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as a = 1 cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c = 1) { ... } is syntactically valid (but probably unintended) C code but if c = 1: ... causes a syntax error in Python.

Methods[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=7)]

[Methods](https://en.wikipedia.org/wiki/Method_(programming)) on objects are [functions](https://en.wikipedia.org/wiki/Function_(programming)) attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, [syntactic sugar](https://en.wikipedia.org/wiki/Syntactic_sugar) for Class.method(instance, argument). Python methods have an explicit [self](https://en.wikipedia.org/wiki/This_(computer_programming)" \o "This (computer programming)) parameter to access [instance data](https://en.wikipedia.org/wiki/Instance_data), in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., [C++](https://en.wikipedia.org/wiki/C%2B%2B), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [Objective-C](https://en.wikipedia.org/wiki/Objective-C), or [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language))).[[88]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-61-88)

Typing[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=8)]

[](https://en.wikipedia.org/wiki/File:Python_3._The_standard_type_hierarchy.png)

The standard type hierarchy in Python 3

Python uses [duck typing](https://en.wikipedia.org/wiki/Duck_typing) and has typed objects but untyped variable names. Type constraints are not checked at [compile time](https://en.wikipedia.org/wiki/Compile_time); rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being [dynamically typed](https://en.wikipedia.org/wiki/Type_system#Dynamic_type_checking_and_runtime_type_information), Python is [strongly typed](https://en.wikipedia.org/wiki/Strongly_typed_programming_language), forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

Python allows programmers to define their own types using [classes](https://en.wikipedia.org/wiki/Class_(computer_science)), which are most often used for [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming). New [instances](https://en.wikipedia.org/wiki/Object_(computer_science)) of classes are constructed by calling the class (for example, SpamClass() or EggsClass()), and the classes are instances of the [metaclass](https://en.wikipedia.org/wiki/Metaclass" \o "Metaclass) type (itself an instance of itself), allowing [metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming) and [reflection](https://en.wikipedia.org/wiki/Reflection_(computer_science)).

Before version 3.0, Python had two kinds of classes: *old-style* and *new-style*.[[89]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-classy-89) The syntax of both styles is the same, the difference being whether the class object is inherited from, directly or indirectly (all new-style classes inherit from object and are instances of type). In versions of Python 2 from Python 2.2 onwards, both kinds of classes can be used. Old-style classes were eliminated in Python 3.0.

The long-term plan is to support [gradual typing](https://en.wikipedia.org/wiki/Gradual_typing)[[90]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-90) and from Python 3.5, the syntax of the language allows specifying static types but they are not checked in the default implementation, CPython. An experimental optional static type checker named *mypy* supports compile-time type checking.[[91]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-91)

|  |  |  |  |
| --- | --- | --- | --- |
| Summary of Python 3's built-in types | | | |
| Type | [Mutability](https://en.wikipedia.org/wiki/Immutable_object) | Description | Syntax examples |
| bool | immutable | [Boolean value](https://en.wikipedia.org/wiki/Boolean_value) | True False |
| bytearray | mutable | Sequence of [bytes](https://en.wikipedia.org/wiki/Byte) | bytearray(b'Some ASCII') bytearray(b"Some ASCII") bytearray([119, 105, 107, 105]) |
| bytes | immutable | Sequence of bytes | b'Some ASCII' b"Some ASCII" bytes([119, 105, 107, 105]) |
| complex | immutable | [Complex number](https://en.wikipedia.org/wiki/Complex_number) with real and imaginary parts | 3+2.7j |
| dict | mutable | [Associative array](https://en.wikipedia.org/wiki/Associative_array) (or dictionary) of key and value pairs; can contain mixed types (keys and values), keys must be a hashable type | {'key1': 1.0, 3: False} {} |
| ellipsis[a](https://en.wikipedia.org/wiki/Python_(programming_language)#endnote_inaccessible-type) | immutable | An [ellipsis](https://en.wikipedia.org/wiki/Ellipsis_(programming_operator)) placeholder to be used as an index in [NumPy](https://en.wikipedia.org/wiki/NumPy) arrays | ... Ellipsis |
| float | immutable | [Double precision](https://en.wikipedia.org/wiki/Double_precision) [floating point](https://en.wikipedia.org/wiki/Floating_point) number. The precision is machine dependent but in practice is generally implemented as a 64-bit [IEEE 754](https://en.wikipedia.org/wiki/IEEE_754) number with 53 bits of precision[[92]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-92) | 1.414 |
| frozenset | immutable | Unordered [set](https://en.wikipedia.org/wiki/Set_(computer_science)), contains no duplicates; can contain mixed types, if hashable | frozenset([4.0, 'string', True]) |
| int | immutable | [Integer](https://en.wikipedia.org/wiki/Integer_(computer_science)) of unlimited magnitude[[93]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep0237-93) | 42 |
| list | mutable | [List](https://en.wikipedia.org/wiki/List_(computer_science)), can contain mixed types | [4.0, 'string', True] [] |
| NoneType[a](https://en.wikipedia.org/wiki/Python_(programming_language)#endnote_inaccessible-type) | immutable | An object representing the absence of a value, often called [null](https://en.wikipedia.org/wiki/Null_pointer) in other languages | None |
| NotImplementedType[a](https://en.wikipedia.org/wiki/Python_(programming_language)#endnote_inaccessible-type) | immutable | A placeholder that can be returned from [overloaded operators](https://en.wikipedia.org/wiki/Operator_overloading) to indicate unsupported operand types. | NotImplemented |
| range | immutable | A Sequence of numbers commonly used for looping specific number of times in for loops[[94]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-94) | range(1, 10) range(10, -5, -2) |
| set | mutable | Unordered [set](https://en.wikipedia.org/wiki/Set_(computer_science)), contains no duplicates; can contain mixed types, if hashable | {4.0, 'string', True} set() |
| str | immutable | A [character string](https://en.wikipedia.org/wiki/String_(computer_science)): sequence of Unicode codepoints | 'Wikipedia' "Wikipedia"  *"""Spanning*  *multiple*  *lines"""* |
| tuple | immutable | Can contain mixed types | (4.0, 'string', True) ('single element',) () |

[^a](https://en.wikipedia.org/wiki/Python_(programming_language)#ref_inaccessible-type) Not directly accessible by name

Arithmetic operations[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=9)]

Python has the usual symbols for arithmetic operators (+, -, \*, /), the floor division operator // and the [modulo operation](https://en.wikipedia.org/wiki/Modulo_operation) % (where the remainder can be negative, e.g. 4 % -3 == -2). It also has \*\* for [exponentiation](https://en.wikipedia.org/wiki/Exponentiation), e.g. 5\*\*3 == 125 and 9\*\*0.5 == 3.0, and a matrix multiply operator @ .[[95]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-95) These operators work like in traditional math; with the same [precedence rules](https://en.wikipedia.org/wiki/Order_of_operations), the operators [infix](https://en.wikipedia.org/wiki/Infix_notation) ( + and - can also be [unary](https://en.wikipedia.org/wiki/Unary_operation) to represent positive and negative numbers respectively).

The division between integers produces floating-point results. The behavior of division has changed significantly over time:[[96]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep0238-96)

* Python 2.1 and earlier used C's division behavior. The / operator is integer division if both operands are integers, and floating-point division otherwise. Integer division rounds towards 0, e.g. 7/3 == 2 and -7/3 == -2.
* Python 2.2 changed integer division to round towards negative infinity, e.g. 7/3 == 2 and -7/3 == -3. The floor division // operator was introduced. So 7//3 == 2, -7//3 == -3, 7.5//3 == 2.0 and -7.5//3 == -3.0. Adding from \_\_future\_\_ import division causes a module to use Python 3.0 rules for division (see next).
* Python 3.0 changed / to always be floating-point division, e.g. 5/2 == 2.5.

In Python terms, / is *true division* (or simply *division*), and // is *floor division.* / before version 3.0 is *classic division*.[[96]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep0238-96)

Rounding towards negative infinity, though different from most languages, adds consistency. For instance, it means that the equation (a + b)//b == a//b + 1 is always true. It also means that the equation b\*(a//b) + a%b == a is valid for both positive and negative values of a. However, maintaining the validity of this equation means that while the result of a%b is, as expected, in the [half-open interval](https://en.wikipedia.org/wiki/Half-open_interval) [0, *b*), where b is a positive integer, it has to lie in the interval (*b*, 0] when b is negative.[[97]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-62-97)

Python provides a round function for [rounding](https://en.wikipedia.org/wiki/Rounding) a float to the nearest integer. For [tie-breaking](https://en.wikipedia.org/wiki/Rounding#Tie-breaking), Python 3 uses [round to even](https://en.wikipedia.org/wiki/Round_to_even): round(1.5) and round(2.5) both produce 2.[[98]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-64-98) Versions before 3 used [round-away-from-zero](https://en.wikipedia.org/wiki/Rounding#Rounding_away_from_zero): round(0.5) is 1.0, round(-0.5) is −1.0.[[99]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-63-99)

Python allows boolean expressions with multiple equality relations in a manner that is consistent with general use in mathematics. For example, the expression a < b < c tests whether a is less than b and b is less than c.[[100]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-65-100) C-derived languages interpret this expression differently: in C, the expression would first evaluate a < b, resulting in 0 or 1, and that result would then be compared with c.[[101]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-CPL-101)

Python uses [arbitrary-precision arithmetic](https://en.wikipedia.org/wiki/Arbitrary-precision_arithmetic) for all integer operations. The Decimal type/class in the decimal module provides decimal floating point numbers to a pre-defined arbitrary precision and several rounding modes.[[102]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-102) The Fraction class in the fractions module provides arbitrary precision for [rational numbers](https://en.wikipedia.org/wiki/Rational_number).[[103]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-103)

Due to Python's extensive mathematics library, and the third-party library [NumPy](https://en.wikipedia.org/wiki/NumPy) that further extends the native capabilities, it is frequently used as a scientific scripting language to aid in problems such as numerical data processing and manipulation.[[104]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-104)[[105]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-105)

Programming examples[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=10)]

[Hello world](https://en.wikipedia.org/wiki/%22Hello,_World!%22_program) program:

print('Hello, world!')

Program to calculate the [factorial](https://en.wikipedia.org/wiki/Factorial) of a positive integer:

n = int(input('Type a number, and its factorial will be printed: '))

if n < 0:

raise ValueError('You must enter a non negative integer')

fact = 1

for i in range(2, n + 1):

fact \*= i

print(fact)

Libraries[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=11)]

Python's large [standard library](https://en.wikipedia.org/wiki/Standard_library), commonly cited as one of its greatest strengths,[[106]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-86-106) provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as [MIME](https://en.wikipedia.org/wiki/MIME) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) are supported. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary-precision decimals,[[107]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-88-107) manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing).

Some parts of the standard library are covered by specifications (for example, the [Web Server Gateway Interface](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface) (WSGI) implementation wsgiref follows PEP 333[[108]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-89-108)), but most modules are not. They are specified by their code, internal documentation, and [test suites](https://en.wikipedia.org/wiki/Test_suite). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of November 2019, the [Python Package Index](https://en.wikipedia.org/wiki/Python_Package_Index) (PyPI), the official repository for third-party Python software, contains over 200,000[[109]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-109) packages with a wide range of functionality, including:

* [Automation](https://en.wikipedia.org/wiki/Automation)
* [Data analytics](https://en.wikipedia.org/wiki/Data_analytics)
* [Databases](https://en.wikipedia.org/wiki/Databases)
* [Documentation](https://en.wikipedia.org/wiki/Documentation)
* [Graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interfaces)
* [Image processing](https://en.wikipedia.org/wiki/Image_processing)
* [Machine learning](https://en.wikipedia.org/wiki/Machine_learning)
* [Mobile App](https://en.wikipedia.org/wiki/Mobile_App)
* [Multimedia](https://en.wikipedia.org/wiki/Multimedia)
* [Computer Networking](https://en.wikipedia.org/wiki/Computer_networking)
* [Scientific computing](https://en.wikipedia.org/wiki/Scientific_computing)
* [System administration](https://en.wikipedia.org/wiki/System_administration)
* [Test frameworks](https://en.wikipedia.org/wiki/Test_framework)
* [Text processing](https://en.wikipedia.org/wiki/Text_processing)
* [Web frameworks](https://en.wikipedia.org/wiki/Web_framework)
* [Web scraping](https://en.wikipedia.org/wiki/Web_scraping)[[110]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-110)

Development environments[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=12)]

*See also:*[*Comparison of integrated development environments § Python*](https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments#Python)

Most Python implementations (including CPython) include a [read–eval–print loop](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) (REPL), permitting them to function as a [command line interpreter](https://en.wikipedia.org/wiki/Command_line_interpreter) for which the user enters statements sequentially and receives results immediately.

Other shells, including [IDLE](https://en.wikipedia.org/wiki/IDLE) and [IPython](https://en.wikipedia.org/wiki/IPython" \o "IPython), add further abilities such as improved auto-completion, session state retention and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting).

As well as standard desktop [integrated development environments](https://en.wikipedia.org/wiki/Integrated_development_environment), there are [Web browser](https://en.wikipedia.org/wiki/Web_browser)-based IDEs; [SageMath](https://en.wikipedia.org/wiki/SageMath" \o "SageMath) (intended for developing science and math-related Python programs); [PythonAnywhere](https://en.wikipedia.org/wiki/PythonAnywhere), a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing).[[111]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-111)

Implementations[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=13)]

*See also:*[*List of Python software § Python implementations*](https://en.wikipedia.org/wiki/List_of_Python_software#Python_implementations)

Reference implementation[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=14)]

[CPython](https://en.wikipedia.org/wiki/CPython) is the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python. It is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)), meeting the [C89](https://en.wikipedia.org/wiki/C89_(C_version)) standard with several select [C99](https://en.wikipedia.org/wiki/C99) features.[[112]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-66-112) It compiles Python programs into an intermediate [bytecode](https://en.wikipedia.org/wiki/Bytecode)[[113]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-67-113) which is then executed by its [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine).[[114]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-68-114) CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) (since Python 3.9.1 version, Python "actively disallows installation on [Windows 7](https://en.wikipedia.org/wiki/Windows_7)";[[115]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-115) and did support [Windows XP](https://en.wikipedia.org/wiki/Windows_XP) and older, with by now unsupported Python 2.7) and most modern [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems, including macOS (and [Apple M1](https://en.wikipedia.org/wiki/Apple_M1) Macs, since Python 3.9.1, with experimental installer) and unofficial support for e.g. [VMS](https://en.wikipedia.org/wiki/OpenVMS).[[116]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-116) Platform portability was one of its earliest priorities,[[117]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-69-117) in Python 1 and 2 time-frame, even supporting [OS/2](https://en.wikipedia.org/wiki/OS/2) and [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system));[[118]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-118) support has since been dropped for a lot of platforms.

Other implementations[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=15)]

[PyPy](https://en.wikipedia.org/wiki/PyPy) is a fast, compliant interpreter of Python 2.7 and 3.6.[[119]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-70-119) Its [just-in-time compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation) brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.[[120]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-71-120)[[121]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-121)

[Stackless Python](https://en.wikipedia.org/wiki/Stackless_Python) is a significant fork of CPython that implements [microthreads](https://en.wikipedia.org/wiki/Microthread" \o "Microthread); it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless version.[[122]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-73-122)

[MicroPython](https://en.wikipedia.org/wiki/MicroPython) and [CircuitPython](https://en.wikipedia.org/wiki/CircuitPython" \o "CircuitPython) are Python 3 variants optimized for [microcontrollers](https://en.wikipedia.org/wiki/Microcontroller). This includes [Lego Mindstorms EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3).[[123]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-123)

Unsupported implementations[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=16)]

Other just-in-time Python compilers have been developed, but are now unsupported:

* Google began a project named [Unladen Swallow](https://en.wikipedia.org/wiki/Unladen_Swallow) in 2009, with the aim of speeding up the Python interpreter fivefold by using the [LLVM](https://en.wikipedia.org/wiki/LLVM), and of improving its multithreading ability to scale to thousands of cores,[[124]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-74-124) while ordinary implementations suffer from the [global interpreter lock](https://en.wikipedia.org/wiki/Global_interpreter_lock).
* [Psyco](https://en.wikipedia.org/wiki/Psyco) was a [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) [specializing](https://en.wikipedia.org/wiki/Run-time_algorithm_specialization) compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain [data types](https://en.wikipedia.org/wiki/Data_type) and is faster than the standard Python code.

In 2005, [Nokia](https://en.wikipedia.org/wiki/Nokia) released a Python interpreter for the [Series 60](https://en.wikipedia.org/wiki/Series_60) mobile phones named [PyS60](https://en.wikipedia.org/wiki/PyS60). It includes many of the modules from the CPython implementations and some additional modules to integrate with the [Symbian](https://en.wikipedia.org/wiki/Symbian) operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia [N900](https://en.wikipedia.org/wiki/N900) also supports Python with [GTK](https://en.wikipedia.org/wiki/GTK) widget libraries, enabling programs to be written and run on the target device.[[125]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-125)

Cross-compilers to other languages[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=17)]

There are several compilers to high-level [object languages](https://en.wikipedia.org/wiki/Object_language), with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

* [Cython](https://en.wikipedia.org/wiki/Cython) compiles Python to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B).
* Google's Grumpy (latest release in 2017) compiles Python 2 to [Go](https://en.wikipedia.org/wiki/Go_(programming_language)).[[126]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-126)[[127]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-127)[[128]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-128)
* [IronPython](https://en.wikipedia.org/wiki/IronPython) follows a similar approach in order to run Python programs on the .NET [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime).
* [Jython](https://en.wikipedia.org/wiki/Jython) enables the use of the Java class library from a Python program.
* [MyHDL](https://en.wikipedia.org/wiki/MyHDL) compiles Python to [VHDL](https://en.wikipedia.org/wiki/VHDL).
* [Nuitka](https://en.wikipedia.org/wiki/Nuitka) compiles Python into C++.[[129]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-129)
* [Numba](https://en.wikipedia.org/wiki/Numba) uses [LLVM](https://en.wikipedia.org/wiki/LLVM) to compile Python to machine code.
* [Pyjs](https://en.wikipedia.org/wiki/Pyjs) (latest release in 2012) compiles Python to [JavaScript](https://en.wikipedia.org/wiki/JavaScript).
* [Pyrex](https://en.wikipedia.org/wiki/Pyrex_(programming_language)) (latest release in 2010) and [Shed Skin](https://en.wikipedia.org/wiki/Shed_Skin) (latest release in 2013) compile to C and C++ respectively.
* Pythran compiles Python to [C++](https://en.wikipedia.org/wiki/C%2B%2B).[[130]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-130)[[131]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-131)[[132]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-132)
* [RPython](https://en.wikipedia.org/wiki/RPython) can be compiled to [C](https://en.wikipedia.org/wiki/C_(programming_language)), and is used to build the PyPy interpreter of Python.
* Transcrypt compiles Python to JavaScript.[[133]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-133)[[134]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-134)

Performance[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=18)]

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.[[135]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-135) Python's performance compared to other programming languages has also been benchmarked by [The Computer Language Benchmarks Game](https://en.wikipedia.org/wiki/The_Computer_Language_Benchmarks_Game).[[136]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-136)

Development[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=19)]

Python's development is conducted largely through the *Python Enhancement Proposal* (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues and documenting Python design decisions.[[137]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PepCite000-137) Python coding style is covered in PEP 8.[[138]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-138) Outstanding PEPs are reviewed and commented on by the Python community and the steering council.[[137]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PepCite000-137)

Enhancement of the language corresponds with development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are discussed in the [Roundup](https://en.wikipedia.org/wiki/Roundup_(issue_tracker)) [bug tracker](https://en.wikipedia.org/wiki/Bug_tracker) hosted at [bugs.python.org](https://bugs.python.org/).[[139]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-21-139) Development originally took place on a [self-hosted](https://en.wikipedia.org/wiki/Self-hosting_(web_services)) source-code repository running [Mercurial](https://en.wikipedia.org/wiki/Mercurial), until Python moved to [GitHub](https://en.wikipedia.org/wiki/GitHub) in January 2017.[[140]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-py_dev_guide-140)

CPython's public releases come in three types, distinguished by which part of the version number is incremented:

* Backward-incompatible versions, where code is expected to break and need to be manually [ported](https://en.wikipedia.org/wiki/Ported). The first part of the version number is incremented. These releases happen infrequently—for example, version 3.0 was released 8 years after 2.0.
* Major or "feature" releases, about every 18 months, are largely compatible but introduce new features. The second part of the version number is incremented. Each major version is supported by bugfixes for several years after its release.[[141]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-release-schedule-141)
* Bugfix releases,[[142]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-22-142) which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release. Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented.[[142]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-22-142)

Python 3.9 alpha1 was announced in November 2019[[143]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-143) and with the adoption of a new yearly release cadence,[[144]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-144)[[145]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-145) the first release of 3.9 is slated for November 2020.[[146]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-146)

Many [alpha, beta, and release-candidates](https://en.wikipedia.org/wiki/Beta_release) are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large [unit test](https://en.wikipedia.org/wiki/Unit_test) suite during development, and using the [BuildBot](https://en.wikipedia.org/wiki/BuildBot" \o "BuildBot) [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration) system.[[147]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-23-147)

The major [academic conference](https://en.wikipedia.org/wiki/Academic_conference) on Python is [PyCon](https://en.wikipedia.org/wiki/PyCon" \o "PyCon). There are also special Python mentoring programmes, such as [Pyladies](https://en.wikipedia.org/wiki/Pyladies" \o "Pyladies).

API documentation generators[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=20)]

Python API documentation generators include:

* [pydoc](https://en.wikipedia.org/wiki/Pydoc)
* [Sphinx](https://en.wikipedia.org/wiki/Sphinx_(documentation_generator))

Naming[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=21)]

Python's name is derived from the British comedy group [Monty Python](https://en.wikipedia.org/wiki/Monty_Python), whom Python creator Guido van Rossum enjoyed while developing the language. Monty Python references appear frequently in Python code and culture;[[148]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-tutorial-chapter1-148) for example, the [metasyntactic variables](https://en.wikipedia.org/wiki/Metasyntactic_variable) often used in Python literature are [*spam* and *eggs*](https://en.wikipedia.org/wiki/Spam_(Monty_Python)) instead of the traditional [*foo* and *bar*](https://en.wikipedia.org/wiki/Foobar).[[148]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-tutorial-chapter1-148)[[149]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-26-149) The official Python documentation also contains various references to Monty Python routines.[[150]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-150)[[151]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-151)

The prefix *Py-* is used to show that something is related to Python. Examples of the use of this prefix in names of Python applications or libraries include [Pygame](https://en.wikipedia.org/wiki/Pygame" \o "Pygame), a [binding](https://en.wikipedia.org/wiki/Language_binding) of [SDL](https://en.wikipedia.org/wiki/Simple_DirectMedia_Layer) to Python (commonly used to create games); [PyQt](https://en.wikipedia.org/wiki/PyQt" \o "PyQt) and [PyGTK](https://en.wikipedia.org/wiki/PyGTK" \o "PyGTK), which bind [Qt](https://en.wikipedia.org/wiki/Qt_(software)) and [GTK](https://en.wikipedia.org/wiki/GTK) to Python respectively; and [PyPy](https://en.wikipedia.org/wiki/PyPy" \o "PyPy), a Python implementation originally written in Python.

Uses[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=22)]

*Main article:*[*List of Python software*](https://en.wikipedia.org/wiki/List_of_Python_software)

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the [TIOBE Programming Community Index](https://en.wikipedia.org/wiki/TIOBE_Programming_Community_Index) where, as of February 2020, it is the third most popular language (behind [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), and [C](https://en.wikipedia.org/wiki/C_(programming_language))).[[152]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-152) It was selected Programming Language of the Year in 2007, 2010, and 2018.[[153]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-34-153)

An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".[[154]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-28-154)

Large organizations that use Python include [Wikipedia](https://en.wikipedia.org/wiki/Wikipedia), [Google](https://en.wikipedia.org/wiki/Google),[[155]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-quotes-about-python-155) [Yahoo!](https://en.wikipedia.org/wiki/Yahoo!),[[156]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-29-156) [CERN](https://en.wikipedia.org/wiki/CERN),[[157]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-30-157) [NASA](https://en.wikipedia.org/wiki/NASA),[[158]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-31-158) [Facebook](https://en.wikipedia.org/wiki/Facebook),[[159]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-159) [Amazon](https://en.wikipedia.org/wiki/Amazon_(company)), [Instagram](https://en.wikipedia.org/wiki/Instagram),[[160]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-160) [Spotify](https://en.wikipedia.org/wiki/Spotify)[[161]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-161) and some smaller entities like [ILM](https://en.wikipedia.org/wiki/Industrial_Light_%26_Magic)[[162]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-32-162) and [ITA](https://en.wikipedia.org/wiki/ITA_Software).[[163]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-33-163) The social news networking site [Reddit](https://en.wikipedia.org/wiki/Reddit) is written entirely in Python.[[164]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-164)

Python can serve as a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) for [web applications](https://en.wikipedia.org/wiki/Web_application), e.g., via [mod\_wsgi](https://en.wikipedia.org/wiki/Mod_wsgi" \o "Mod wsgi) for the [Apache web server](https://en.wikipedia.org/wiki/Apache_web_server).[[165]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-35-165) With [Web Server Gateway Interface](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface), a standard API has evolved to facilitate these applications. [Web frameworks](https://en.wikipedia.org/wiki/Web_framework) like [Django](https://en.wikipedia.org/wiki/Django_(web_framework)), [Pylons](https://en.wikipedia.org/wiki/Pylons_(web_framework)), [Pyramid](https://en.wikipedia.org/wiki/Pyramid_(web_framework)), [TurboGears](https://en.wikipedia.org/wiki/TurboGears), [web2py](https://en.wikipedia.org/wiki/Web2py), [Tornado](https://en.wikipedia.org/wiki/Tornado_(web_server)), [Flask](https://en.wikipedia.org/wiki/Flask_(web_framework)), [Bottle](https://en.wikipedia.org/wiki/Bottle_(web_framework)) and [Zope](https://en.wikipedia.org/wiki/Zope) support developers in the design and maintenance of complex applications. [Pyjs](https://en.wikipedia.org/wiki/Pyjs" \o "Pyjs) and [IronPython](https://en.wikipedia.org/wiki/IronPython" \o "IronPython) can be used to develop the client-side of Ajax-based applications. [SQLAlchemy](https://en.wikipedia.org/wiki/SQLAlchemy" \o "SQLAlchemy) can be used as a [data mapper](https://en.wikipedia.org/wiki/Data_mapper_pattern) to a relational database. [Twisted](https://en.wikipedia.org/wiki/Twisted_(software)) is a framework to program communications between computers, and is used (for example) by [Dropbox](https://en.wikipedia.org/wiki/Dropbox_(service)).

Libraries such as [NumPy](https://en.wikipedia.org/wiki/NumPy), [SciPy](https://en.wikipedia.org/wiki/SciPy) and [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) allow the effective use of Python in [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing),[[166]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-cise-166)[[167]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-millman-167) with specialized libraries such as [Biopython](https://en.wikipedia.org/wiki/Biopython" \o "Biopython) and [Astropy](https://en.wikipedia.org/wiki/Astropy" \o "Astropy) providing domain-specific functionality. [SageMath](https://en.wikipedia.org/wiki/SageMath" \o "SageMath) is a [mathematical software](https://en.wikipedia.org/wiki/Mathematical_software) with a [notebook interface](https://en.wikipedia.org/wiki/Notebook_interface) programmable in Python: its library covers many aspects of [mathematics](https://en.wikipedia.org/wiki/Mathematics), including [algebra](https://en.wikipedia.org/wiki/Algebra), [combinatorics](https://en.wikipedia.org/wiki/Combinatorics), [numerical mathematics](https://en.wikipedia.org/wiki/Numerical_mathematics), [number theory](https://en.wikipedia.org/wiki/Number_theory), and [calculus](https://en.wikipedia.org/wiki/Calculus).[[168]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-ICSE-168) [OpenCV](https://en.wikipedia.org/wiki/OpenCV) has python bindings with a rich set of features for [computer vision](https://en.wikipedia.org/wiki/Computer_vision) and [image processing](https://en.wikipedia.org/wiki/Digital_image_processing).[[169]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-169)

Python has been successfully embedded in many software products as a scripting language, including in [finite element method](https://en.wikipedia.org/wiki/Finite_element_method) software such as [Abaqus](https://en.wikipedia.org/wiki/Abaqus), 3D parametric modeler like [FreeCAD](https://en.wikipedia.org/wiki/FreeCAD" \o "FreeCAD), 3D animation packages such as [3ds Max](https://en.wikipedia.org/wiki/3ds_Max), [Blender](https://en.wikipedia.org/wiki/Blender_(software)), [Cinema 4D](https://en.wikipedia.org/wiki/Cinema_4D), [Lightwave](https://en.wikipedia.org/wiki/Lightwave), [Houdini](https://en.wikipedia.org/wiki/Houdini_(software)), [Maya](https://en.wikipedia.org/wiki/Maya_(software)), [modo](https://en.wikipedia.org/wiki/Modo_(software)), [MotionBuilder](https://en.wikipedia.org/wiki/MotionBuilder" \o "MotionBuilder), [Softimage](https://en.wikipedia.org/wiki/Autodesk_Softimage), the visual effects compositor [Nuke](https://en.wikipedia.org/wiki/Nuke_(software)), 2D imaging programs like [GIMP](https://en.wikipedia.org/wiki/GIMP),[[170]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-170) [Inkscape](https://en.wikipedia.org/wiki/Inkscape), [Scribus](https://en.wikipedia.org/wiki/Scribus" \o "Scribus) and [Paint Shop Pro](https://en.wikipedia.org/wiki/Paint_Shop_Pro),[[171]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-38-171) and [musical notation](https://en.wikipedia.org/wiki/Musical_notation) programs like [scorewriter](https://en.wikipedia.org/wiki/Scorewriter" \o "Scorewriter) and [capella](https://en.wikipedia.org/wiki/Capella_(notation_program)" \o "Capella (notation program)). [GNU Debugger](https://en.wikipedia.org/wiki/GNU_Debugger) uses Python as a [pretty printer](https://en.wikipedia.org/wiki/Prettyprint) to show complex structures such as C++ containers. [Esri](https://en.wikipedia.org/wiki/Esri) promotes Python as the best choice for writing scripts in [ArcGIS](https://en.wikipedia.org/wiki/ArcGIS).[[172]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-39-172) It has also been used in several video games,[[173]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-40-173)[[174]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-41-174) and has been adopted as first of the three available [programming languages](https://en.wikipedia.org/wiki/Programming_language) in [Google App Engine](https://en.wikipedia.org/wiki/Google_App_Engine), the other two being [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) and [Go](https://en.wikipedia.org/wiki/Go_(programming_language)).[[175]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-42-175)

Python is commonly used in [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) projects and machine learning projects with the help of libraries like [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Keras](https://en.wikipedia.org/wiki/Keras), [Pytorch](https://en.wikipedia.org/wiki/PyTorch) and [Scikit-learn](https://en.wikipedia.org/wiki/Scikit-learn).[[176]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-whitepaper2015-176)[[177]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-177)[[178]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-178)[[179]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-179) As a scripting language with [modular architecture](https://en.wikipedia.org/wiki/Modular_programming), simple syntax and rich text processing tools, Python is often used for [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing).[[180]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-47-180)

Many operating systems include Python as a standard component. It ships with most [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution),[[181]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-181) [AmigaOS 4](https://en.wikipedia.org/wiki/AmigaOS_4" \o "AmigaOS 4) (using Python 2.7), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) (as a package), [NetBSD](https://en.wikipedia.org/wiki/NetBSD), [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD) (as a package) and [macOS](https://en.wikipedia.org/wiki/MacOS) and can be used from the command line (terminal). Many Linux distributions use installers written in Python: [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) uses the [Ubiquity](https://en.wikipedia.org/wiki/Ubiquity_(software)) installer, while [Red Hat Linux](https://en.wikipedia.org/wiki/Red_Hat_Linux) and [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system)) use the [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(installer)) installer. [Gentoo Linux](https://en.wikipedia.org/wiki/Gentoo_Linux) uses Python in its [package management system](https://en.wikipedia.org/wiki/Package_management_system), [Portage](https://en.wikipedia.org/wiki/Portage_(software)).

Python is used extensively in the [information security](https://en.wikipedia.org/wiki/Information_security) industry, including in exploit development.[[182]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-49-182)[[183]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-50-183)

Most of the [Sugar](https://en.wikipedia.org/wiki/Sugar_(software)) software for the [One Laptop per Child](https://en.wikipedia.org/wiki/One_Laptop_per_Child) XO, now developed at [Sugar Labs](https://en.wikipedia.org/wiki/Sugar_Labs), is written in Python.[[184]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-51-184) The [Raspberry Pi](https://en.wikipedia.org/wiki/Raspberry_Pi) [single-board computer](https://en.wikipedia.org/wiki/Single-board_computer) project has adopted Python as its main user-programming language.

[LibreOffice](https://en.wikipedia.org/wiki/LibreOffice) includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature[[185]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-185) since Version 4.0 from 7 February 2013.

Languages influenced by Python[[edit](https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit&section=23)]

Python's design and philosophy have influenced many other programming languages:

* [CoffeeScript](https://en.wikipedia.org/wiki/CoffeeScript), a [programming language](https://en.wikipedia.org/wiki/Programming_language) that cross-compiles to JavaScript, has Python-inspired syntax.
* [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript)/[JavaScript](https://en.wikipedia.org/wiki/JavaScript) borrowed [iterators](https://en.wikipedia.org/wiki/Iterator) and [generators](https://en.wikipedia.org/wiki/Generator_(computer_science)) from Python.[[186]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-93-186)
* [GDScript](https://en.wikipedia.org/wiki/GDScript), a scripting language very similar to Python, built-in to the [Godot](https://en.wikipedia.org/wiki/Godot_(game_engine)) game engine.[[187]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-187)
* [Go](https://en.wikipedia.org/wiki/Go_(programming_language)) is designed for the "speed of working in a dynamic language like Python"[[188]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-94-188) and shares the same syntax for slicing arrays.
* [Groovy](https://en.wikipedia.org/wiki/Groovy_(programming_language)) was motivated by the desire to bring the Python design philosophy to [Java](https://en.wikipedia.org/wiki/Java_(programming_language)).[[189]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-95-189)
* [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)) was designed to be "as usable for general programming as Python".[[24]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Julia-24)
* [Nim](https://en.wikipedia.org/wiki/Nim_(programming_language)) uses indentation and similar syntax.[[190]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-190)
* [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language))'s creator, [Yukihiro Matsumoto](https://en.wikipedia.org/wiki/Yukihiro_Matsumoto), has said: "I wanted a scripting language that was more powerful than Perl, and more object-oriented than Python. That's why I decided to design my own language."[[191]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-linuxdevcenter-191)
* [Swift](https://en.wikipedia.org/wiki/Swift_(programming_language)), a programming language developed by Apple, has some Python-inspired syntax.[[192]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-192)

Python's development practices have also been emulated by other languages. For example, the practice of requiring a document describing the rationale for, and issues surrounding, a change to the language (in Python, a PEP) is also used in [Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl),[[193]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-99-193) [Erlang](https://en.wikipedia.org/wiki/Erlang_(programming_language)),[[194]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-100-194) and Swift.[[195]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-195)