

Network Programming in Python I

Justin Ellis MBA

Any Questions?

Python History

- Implementation in 1989
- Named after BBC show Monty Python's Flying Circus

Python History

Version ↕	Latest micro version ↕	Release date ↕	End of full support ↕	End of security fixes ↕
0.9	0.9.9 ^[2]	1991-02-20 ^[2]	1993-07-29 ^{[a][2]}	
1.0	1.0.4 ^[2]	1994-01-26 ^[2]	1994-02-15 ^{[a][2]}	
1.1	1.1.1 ^[2]	1994-10-11 ^[2]	1994-11-10 ^{[a][2]}	
1.2		1995-04-13 ^[2]	Unsupported	
1.3		1995-10-13 ^[2]	Unsupported	
1.4		1996-10-25 ^[2]	Unsupported	
1.5	1.5.2 ^[39]	1998-01-03 ^[2]	1999-04-13 ^{[a][2]}	
1.6	1.6.1 ^[39]	2000-09-05 ^[40]	2000-09 ^{[a][39]}	
2.0	2.0.1 ^[41]	2000-10-16 ^[42]	2001-06-22 ^{[a][41]}	
2.1	2.1.3 ^[41]	2001-04-15 ^[43]	2002-04-09 ^{[a][41]}	
2.2	2.2.3 ^[41]	2001-12-21 ^[44]	2003-05-30 ^{[a][41]}	
2.3	2.3.7 ^[41]	2003-06-29 ^[45]	2008-03-11 ^{[a][41]}	
2.4	2.4.6 ^[41]	2004-11-30 ^[46]	2008-12-19 ^{[a][41]}	
2.5	2.5.6 ^[41]	2006-09-19 ^[47]	2011-05-26 ^{[a][41]}	
2.6	2.6.9 ^[26]	2008-10-01 ^[26]	2010-08-24 ^{[b][26]}	2013-10-29 ^[26]
2.7	2.7.18 ^[31]	2010-07-03 ^[31]	2020-01-01 ^{[c][31]}	
3.0	3.0.1 ^[41]	2008-12-03 ^[26]	2009-06-27 ^[48]	
3.1	3.1.5 ^[49]	2009-06-27 ^[49]	2011-06-12 ^[50]	2012-06 ^[49]
3.2	3.2.6 ^[51]	2011-02-20 ^[51]	2013-05-13 ^{[b][51]}	2016-02-20 ^[51]
3.3	3.3.7 ^[52]	2012-09-29 ^[52]	2014-03-08 ^{[b][52]}	2017-09-29 ^[52]
3.4	3.4.10 ^[53]	2014-03-16 ^[53]	2017-08-09 ^[54]	2019-03-18 ^{[a][53]}
3.5	3.5.10 ^[55]	2015-09-13 ^[55]	2017-08-08 ^[56]	2020-09-30 ^[55]
3.6	3.6.13 ^[57]	2016-12-23 ^[57]	2018-12-24 ^{[b][57]}	2021-12 ^[57]
3.7	3.7.10 ^[58]	2018-06-27 ^[58]	2020-06-27 ^{[b][58]}	2023-06 ^[58]
3.8	3.8.9 ^[59]	2019-10-14 ^[59]	2021-05-03 ^[59]	2024-10 ^[59]
3.9	3.9.4 ^[60]	2020-10-05 ^[60]	2022-05 ^[61]	2025-10 ^{[60][61]}
3.10		2021-10-04 ^[62]	2023-05 ^[62]	2026-10 ^[62]

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

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

Python Data Types

- [Python Data Types \(w3schools.com\)](https://www.w3schools.com/python/python_data_types.asp)

How We run Python

Integrated development environment

From Wikipedia, the free encyclopedia

An **integrated development environment** (IDE) is a [software application](#) that provides comprehensive facilities to [computer programmers](#) for [software development](#). An IDE normally consists of at least a [source code editor](#), [build automation](#) tools and a [debugger](#). Some IDEs, such as [NetBeans](#) and [Eclipse](#), contain the necessary [compiler](#), [interpreter](#), or both; others, such as [SharpDevelop](#) and [Lazarus](#), do not.

The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a [version control system](#) or various tools to simplify the construction of a [graphical user interface](#) (GUI) are integrated. Many modern IDEs also have a [class browser](#), an [object browser](#), and a [class hierarchy diagram](#) for use in [object-oriented software development](#).

Contents [\[hide\]](#)

- 1 [Overview](#)
- 2 [History](#)
- 3 [Topics](#)
 - 3.1 [Syntax highlighting](#)
 - 3.2 [Code completion](#)
 - 3.3 [Refactoring](#)
 - 3.4 [Version control](#)
 - 3.5 [Debugging](#)
 - 3.6 [Code search](#)
 - 3.7 [Visual programming](#)
 - 3.8 [Language support](#)
 - 3.9 [Attitudes across different computing platforms](#)
- 4 [Artificial intelligence](#)
- 5 [Web integrated development environment](#)
- 6 [See also](#)
- 7 [References](#)

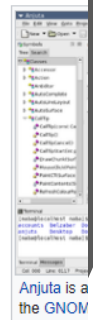
Overview [\[edit \]](#)

Integrated development environments are designed to maximize programmer productivity by providing tight-knit components with similar [user interfaces](#). IDEs present a single program in which all development is done. This program typically provides many features for authoring, modifying, compiling, deploying and debugging software. This contrasts with software development using unrelated tools, such as [vi](#), [GCC](#) or [make](#).

One aim of the IDE is to reduce the configuration necessary to piece together multiple development utilities, instead, it provides the same set of capabilities as one cohesive unit. Reducing setup time can increase developer productivity, especially in cases where learning to use the IDE is faster than manually integrating and learning all of the individual tools. Tighter integration of all development tasks has the potential to improve overall productivity beyond just helping with setup tasks. For example, code can be continuously parsed while it is being edited, providing instant feedback when syntax errors are introduced, thus allowing developers to debug code much faster and more easily with an IDE.

Some IDEs are dedicated to a specific [programming language](#), allowing a feature set that most closely matches the [programming paradigms](#) of the language. However, there are many multiple-language IDEs.

While most modern IDEs are graphical, text-based IDEs such as [Turbo Pascal](#) were in popular use before the availability of windowing systems like [Microsoft Windows](#) and the [X Window System](#) (X11). They commonly use function keys or [hotkeys](#) to execute frequently used commands or macros.




Different Types of IDE

Top Python IDEs And Code Editors Comparison

There are several Python IDE and Code editors that are discussed in this article and all the information that is required to choose the best IDE for your organization are explained here.

Comparison Table

IDE	User Rating	Size in MB	Developed in
PyCharm 	4.5/5	BIG	JAVA, PYTHON
Spyder 	May 4, 2018	BIG	PYTHON
PyDev 	4.6/5	MEDIUM	JAVA, PYTHON
Idle 	4.2/5	MEDIUM	PYTHON
Wing 	May 4, 2018	BIG	C, C++, PYTHON

A large white circle is centered on a dark gray background. The circle is thick and its edges are slightly irregular, giving it a hand-drawn or stylized appearance. The word "Runestone" is written in white, sans-serif font in the upper right quadrant of the circle.

Runestone