

Full History of Python Language (Founder, Release Date, All Versions)

What is Python Programming Language?

Python is a high-level programming computer language that provides instructions to teach the computer how to perform a task.

It offers efficient high-level data structures and an object-oriented programming style that is simple yet effective.

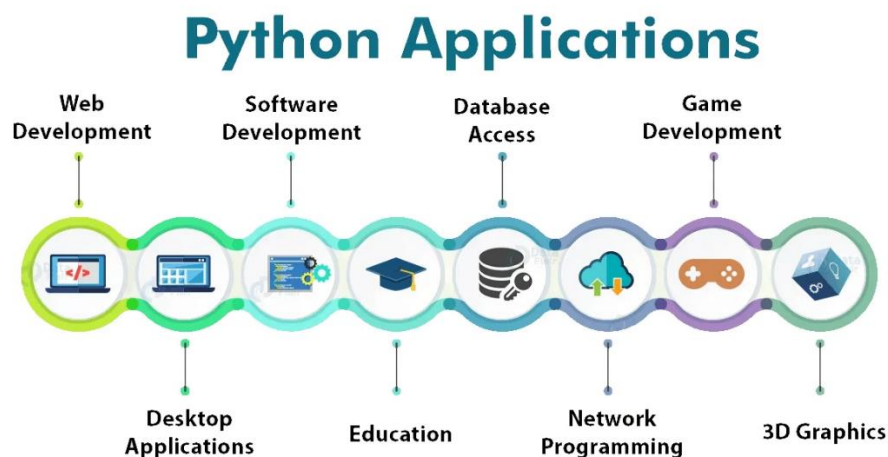
Python, a high-grade language, is a computer programming language that is meant to represent the needs of a problem and mimics natural language or mathematical notation.

It is a free and open-source language.



What is Python used for?

Python is commonly used for developing websites and software, task automation, data analysis, and data visualisation. Since it's relatively easy to learn, Python has been adopted by many non-programmers, such as accountants and scientists, for a variety of everyday tasks, like organising finances.



Fun Fact:

Guido van Rossum was reading the BBC's Monty Python's Flying Circus when developing Python. He named this language after Python, where he thinks the length is just right with a subtle mystery.

Who Developed Python Programming Language?

Python was created by Guido van Rossum, a Dutch programmer. He started working on Python in the late 1980s, and the first official release, Python 0.9.0, came out in February 1991.

Obviously, when talking about the history of Python language, the first question that arises is who developed Python. And, Guido van Rossum is known as the founder of Python.

Guido studied mathematics and computer science at the University of Amsterdam. His early exposure to programming languages like ABC influenced the development of Python.

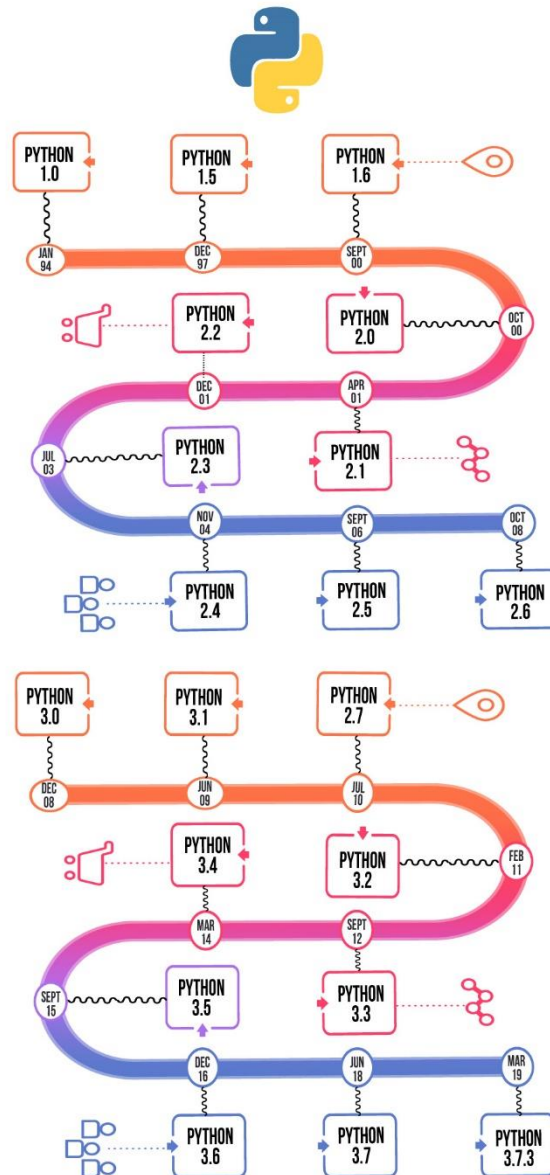
He began working on Python in the late 1980s while working at the Centrum Wiskunde & Informatica (CWI) in the Netherlands.

Guido assumed the role of BDFL, a title he gave himself, signifying his role as the ultimate decision-maker in the Python development process. This position allowed him to make final decisions on the language's features and design principles.

In July 2018, Guido van Rossum announced his decision to step down as the BDFL, citing the need for a more inclusive decision-making process. This marked a shift to a consensus-driven governance model through the Python Steering Council.

Python Language Over The Years

Python is an open-source programming language that is free to use and distribute. **Many developers worldwide utilized it to create GUI applications and web applications with fewer, easily readable codes.** Building large and complicated software applications is much faster with lesser time.



Below are all the versions of Python programming language:

- Python 0.9.0 (February 1991)

This is the first public release of Python by Guido van Rossum. It included features like **exception handling**, **functions**, and **modules**.

- Python 1.0 (January 1994)

The first official version included **lambda**, map, filter, and reduce functions. It also introduced list comprehensions.

- Python 2.0 (October 2000)

This version featured garbage collection, Unicode support, and introduced the list comprehensions feature. It marked a significant step forward in the language's development.

- Python 2.1 (April 2001)

This release focused on stability and improved Unicode support. It introduced a garbage collection system capable of collecting reference cycles.

- Python 2.2 (December 2001)

The major features included a garbage collector for memory management, improvements to Unicode support, and a new type system called "New Style Classes."

- Python 2.3 (July 2003)

This version introduced the `enumerate()` function, the set data type, and improvements to generators. It also included a new `type()` function.

- Python 2.4 (November 2004)

Key features included the addition of **decorators**, generator expressions, and the `with` statement for improved resource management.

- Python 2.5 (September 2006)

This version introduced the `with` statement, conditional expressions (PEP 308), and the `ctypes` module for interfacing with C libraries.

- Python 2.6 (October 2008)

This release focused on stability, performance improvements, and the introduction of the `collections` module. It also included an alpha version of the `io` module.

- Python 2.7 (July 2010)

This version was meant to be the last release in the 2.x series. It included features from Python 3, aiming to ease the transition for developers.

The transition from Python 2 to Python 3 was a major milestone in the language's history. Python 3 introduced several backward-incompatible changes, and development efforts were focused on encouraging the community to adopt Python 3.

- Python 3.0 (December 2008)

Also known as Python 3000 or Py3k, this version featured major changes to the language syntax and standard library to enhance clarity and remove redundancy.

- Python 3.1 (June 2009)

This release included bug fixes and improvements to the Python 3 series.

- Python 3.2 (February 2011)

Introduced the `concurrent.futures` module, improved the `argparse` module, and included several performance enhancements.

- Python 3.3 (September 2012)

Featured significant improvements, including the `yield from` expression, a new `ipaddress` module, and improvements to the `unittest` module.

- **Python 3.4 (March 2014)**

Introduced the asyncio module, improved performance with the enum module, and added a new statistics module.

- **Python 3.5 (September 2015)**

Introduced the async and await syntax for asynchronous programming, matrix multiplication operators, and the typing module for type hints.

- **Python 3.6 (December 2016)**

Introduced the formatted string literals (f-strings), the secrets module, and various syntax and library enhancements.

- **Python 3.7 (June 2018)**

Featured data classes, the asyncio.run() function, and improvements to the unittest and zipapp modules.

- **Python 3.8 (October 2019)**

Introduced the walrus operator (:=), positional-only parameters, and improvements to the datetime module.

- **Python 3.9 (October 2020)**

Featured the introduction of the zoneinfo module for time zone support, improved dictionary performance, and new syntax features.

- **Python 3.10 (October 2021)**

Included features like structural pattern matching, improved error messages, and enhancements to the zoneinfo module.

- **Python 3.11 (December 2022)**

This version of Python is claimed to be having faster performance than v3.10. It brought new syntax features like exception groups, and new built-in features like PEP 678 to allow exceptions to be enriched with notes. Additionally, there were new standard library modules to support parsing TOML.

- **Python 3.12 (October 2023)**

The latest version of Python language is Python 3.12. It was released on 2 October 2023.

This version brings updates to usability, correctness, and cleaning up deprecated APIs. In Python 3.12, there is no distutils package in the standard library.

Why Python is a Popular Programming Language?

The growing adoption of Python across various industries is supported by the following factors mentioned below:

1. Versatility

Python is a general-purpose programming language that can be used for a wide variety of tasks, ranging from **web development** to data analysis to **artificial intelligence (AI)**. The amount of versatility Python offers makes it the most used programming language.

2. Ease of Use

Python's syntax is designed to be simple and easy to understand, making it accessible for all levels of developers. It offers clear and readable code. This is the reason why developers' first language is Python nowadays.

3. Large Community

Python programming language has a large and active community. It has thousands of libraries and frameworks available. Resource finding becomes easy for developers due to high support from the community.

4. High Demand

Python has witnessed a high demand in the job market, across a range of industries, such as finance, healthcare, retail, and technology. These options make Python an easy language to learn and provide new learning opportunities to developers. That's the reason more aspirants are looking for the **best Python course online** to upskill and build their careers.

5. Open Source

Python is an open-source programming language, anyone can use, modify and distribute it for free. Many developers around the globe have shown great enthusiasm in contributing to the language, making Python the most efficient and powerful.

6. Powerful and Efficient Libraries

Python has a large number of libraries such as pandas, numpy, TensorFlow, etc. These libraries perform huge tasks making developers' lives a little easy.

Applications of Python Programming Language (Top Uses in 2023)

Now that we know what makes Python such a popular language, let's look at the top 10 uses of Python language with examples:

1. Web Development

The use of Python in web development is becoming prevalent in recent times. But why is that? It is because of the wide array of frameworks and Content Management System (CMS) that exists to simplify a developer's life.

Some of the well-known Python web development frameworks include Django, Pyramid, Flask, and more, while well-known content management systems include Plone CMS, Wagtail, etc.

These frameworks offer security, scalability, and convenience for the development of the website from scratch. More so, the Python framework comes with standard libraries that

allow for protocol integration such as HTML, SSL, FTP, etc. It also helps in processing JSON, XML, and even E-mail. It is one of the main uses of Python programming.

Furthermore, the development time is significantly reduced, allowing more time to market the application. The use of Python in web development has witnessed exponential growth.

Using Python for web development offers:

- Great Visualization
- Enhanced Security
- Efficient Development Process
- Easy to code

2. Data Analysis and Visualization

Python has been the most popular and preferred language when it comes to data analysis and visualization.

Data is the new age currency if you know how to extract relevant information that can help you take calculated risks and increase profits. Python offers in-built libraries such as NumPy, Pandas, TensorFlow, etc. to help extract valuable information from the data. Libraries like Matplotlib and Seaborn further allow us to visualize data through graphs and charts.

The functionalities that Python offers are a great help to data science professionals. Python handles everything from data collection, data sorting, and data analysis to data visualization. All the **top data science courses in India** include Python as an integral part of the curriculum because it is very vital for learners.

If you are still wondering what is the use of Python, then you must know that if you want to get into Data Science, it won't be an exaggeration to say that Python is the first thing that you need to know. Thinking to get into this stream, now is the right time!

3. Machine Learning and Artificial Intelligence (AI)

Artificial Intelligence and **Machine Learning** are, without a doubt, the most widely discussed and popular topics in the current decade. Python along with some other programming languages has seen a steep rise in their use for developing AI and ML-powered applications.

Python is a stable and secure language, making it a perfect fit for handling the intensive and complex computations that keep AI and Machine Learning systems running. Hence, these are other top applications of Python programming.

Some of the popular libraries to do the job are as follows:

- **SciPy:** SciPy is used for scientific and technical computing.
- **Keras:** It is used for Artificial Neural Networks.
- **Pandas:** Panda is used for the manipulation and analysis of data.
- **TensorFlow:** It is used for Machine Learning tasks.

- **NumPy:** It is used for complex mathematical functions and computation.
- **Scikit:** It is used for working with various Machine Learning Models.

4. Game Development

The gaming industry has to be mentioned when it comes to the impactful contribution and uses of Python programming language. It is used in the gaming industry for a variety of purposes, such as game development, game scripting, and game engine development.

Let's look at the top Python frameworks that are used in game Development:

1. PySoy

PySoy is a 3D game engine that supports Python 3.

2. Ren'Py

It is a versatile framework that can be used on any computer. It excels in handling intricate game logic. Its ease of handling text, images, and sounds makes it an ideal choice for digital storytelling and life simulation games.

3. Python-Ogre

It is a powerful rendering engine that offers speed and flexibility, making it a preferred option for game development's impressive feature set.

4. Cocos2d

It is used in building applications, games and some cross-platform GUI-based interactive programs.

5. Panda3D

It is used for real-time python applications in 3D games, simulations, visualizations, etc.

6. Pygame

It is used to build fully featured games and multimedia programs.

7. PyOpenGL

It is compatible with multiple GUI libraries such as PyGame, PyQt, etc.

8. Pyglet

It supports UI event handling, loading images, windows, joysticks, and playing sounds and music.

9. PyKya

It is known as the fast framework for game development. It is used for direct image reading, MPEG video, MP3, WAV, etc.

These libraries have made the game development process completely effortless. Some of your favorite games based on the Python programming language are Battlefield2, Pirates of the Caribbean, Civilization IV, etc.

5. Software Development

One of the core applications of Python is to simplify the software development process for complex apps. Popular apps such as Google, Netflix, and Reddit all use Python. It offers amazing features such as:

- Code reusability
- Platform Independence
- Inbuilt libraries and frameworks
- High compatibility
- Enhanced code readability

Apart from being a primary programming language in projects, software developers also use Python as a supporting language for project management, testing, and building control. It is among the top 10 uses of Python programming language.

6. Web Scraping Applications

The internet holds a vast amount of information that is ready to be utilized. Before getting into web scraping, let's understand what exactly it is.

Web scraping is a powerful tool for automatically extracting data from single or multiple websites using software or scripts and collecting it in one place. After the scraping is done, this data can be used by researchers, analysts, organizations, individuals, sentiment analysis of social media, and various ranges of tasks.

The whole process can be done using libraries and frameworks such as BeautifulSoup, Scrapy, Selenium, etc. So, web scraping is another top application of Python.

7. Embedded Applications

One of the most intriguing features of Python is its ability to run on embedded hardware. For people who are new to this, let's get a clear picture for you.

Embedded hardware, in simple language, is a tiny computer that is created to perform limited tasks. Embedded applications will drive this hardware. Popular Examples of embedded applications are MicroPython, PyMite, and EmbeddedPython.

Currently, there is a comprehensive list of embedded devices that can be found in almost every aspect of our lives. Products such as Digital Cameras, Smartphones, RaspberryPis, and Industrial Robots can be controlled by Python.

8. CAD Applications

CAD stands for Computer-Aided Design. It is a process of creating 2D and 3D models digitally. These are mostly used for product design by various industries such as aerospace, automotive, and many more.

Python is embedded with amazing applications like Blender, FreeCAD, and a lot more to efficiently design products. All these tools provide enhanced features such as technical drawing, recordings, file export, and import.

9. Audio and Video Applications

Python's efficiency is also evident in its application in the field of Audio and Video. When it comes to managing audio and video files, Python provides a wide range of tools and libraries to aid in the process, making it fully equipped with the task.

Tasks such as signal processing, audio recognition, and many more can be easily performed by libraries such as Pyo, pyAudioAnalysis, Dejavu, etc.

When it comes to video, Python offers various libraries such as OpenCV, Scikit-video, and SciPy, that can help you manipulate video for use in other applications.

All your favorite audio and video applications like Netflix, Spotify, and YouTube are written in Python! It has a fair share in the audio and video industry as well.

10. Enterprise Apps

Enterprise applications are different from our normal applications. First, they offer a specific set of features instead of numerous features. Second, their target group is a specific, close-knit group, typically an organization, unlike consumer applications that are created and designed for the general audience

Additional Uses of Python Language

Apart from the above-mentioned applications and uses of Python, there are more industries that have greatly benefited from this programming language. The industries are mentioned below:

1. Robotics

Robotics field has seen immense growth when Python's application is taken into account. The libraries backed up by Python have significantly improved this industry. Python provides libraries such as PyBullet and PyRobot which are highly efficient. This language can also be used to control low-level hardware and microcontrollers.

2. Automation

Python libraries such as Selenium, and Pytest, can be used to automate the testing of web applications and software. Automation is done by using scripts.

3. Image Processing

Due to the increasing use of **Machine Learning** and **Neural Networks**, the role of image processing tools has seen a steep rise in its popularity. To support this popularity, Python offers a list of libraries that simplify much of the initial work of a data scientist. Some of the famous preprocessing Python libraries are OpenCV, Scikit-Image, and Python Imaging Library (PIL).

4. Text Processing

Text processing allows the manipulation of large chunks of text and also gives the flexibility of structuring it. With Python, you can sort lines, extract text, tokenization,

stemming, sentiment analysis, part-of-speech tagging, reformat paragraphs, and many more.

Some of the famous text processing frameworks are NLTK (Natural Language Toolkit), spaCY, gensim, TextBlob, PyTorch, Pattern, etc.

5. Scientific and Numeric Applications

Python's collection of numeric and scientific tools and libraries has helped scientists and researchers solve countless number-crunching problems and uncover new findings. Python can be used excessively in research areas as well.

6. Operating Systems

Operating Systems are the brain and heart of computers. Python is one of the many programming languages that has contributed to the parts of Linux-based Operating systems.

Red Hat Enterprise's Anaconda Installer and Ubuntu's Ubiquity Installer are the two most famous instances of Python's application in the real world.

7. Desktop GUI

It goes without saying that Python's easy-to-understand syntax and modular programming approach are the main aspects of creating a super fast and responsive GUI. There are various tools available for GUI development using Python, like PyQt, Tkinter, and Kivy to name a few.

Why is Python So Popular?

Examples of Successful Projects Built with Python

Now that we have gone through the detailed history of Python programming, let's see which top brands are using this language:

Instagram:

Python played a pivotal role in the development of Instagram. The platform's backend is primarily powered by Django, a Python web framework, highlighting the language's capability to handle the massive scale of a social media giant.

Dropbox:

The cloud storage service Dropbox relies heavily on Python for its backend infrastructure. Python's simplicity and speed in development have been key factors in the success of Dropbox.

YouTube:

YouTube's video recommendation algorithm, a core component of the platform's success, is powered by Python. The language's versatility in data analysis and machine learning makes it a natural choice for such complex tasks.

Spotify:

Python is utilized in various aspects of Spotify's technology stack, including backend services and data analysis. Its ease of integration and robust libraries contribute to Spotify's seamless user experience.

Google:

Python is widely used at Google for various purposes, including web development, system administration, and **artificial intelligence**. Google's support for Python extends to libraries like TensorFlow, contributing to advancements in **machine learning**.

Facebook:

Python is employed extensively at Facebook, powering parts of the platform's backend and infrastructure. The language's versatility aligns with Facebook's diverse technical needs.

Netflix:

Python is an integral part of Netflix's technology stack, used for backend services and data analysis. The simplicity and readability of Python contribute to the maintainability of Netflix's vast codebase.

How Python Has Evolved to Overcome Challenges?

- ## Python 3 Adoption Efforts

The Python community, led by the Python Software Foundation, actively promoted the adoption of Python 3. The sunsetting of Python 2 in 2020 marked a significant milestone, encouraging developers to migrate and embrace the improvements and new features in Python 3.

- ## Performance Enhancements

Python has invested in performance improvements over the years. The introduction of optimizations like PEP 510 (Specialize functions with guards) and PEP 511 (API for code transformers) has contributed to enhancing the speed and efficiency of Python, addressing concerns about its performance.

- ## GIL Mitigation Strategies

While the Global Interpreter Lock remains a part of CPython (the reference implementation), efforts have been made to mitigate its impact. The introduction of asynchronous programming with the `asyncio` module and the promotion of multiprocessing over threading in certain scenarios have provided workarounds for GIL-related issues.

- ## Dependency Management Improvements

The introduction of tools like `pip` and virtual environments has significantly improved Python's dependency management. The adoption of the Python Package Index (PyPI)

as the central repository for **Python packages**, along with the introduction of tools like pipenv and poetry, has streamlined the process of managing project dependencies.

Future Trends and Developments in Python Ecosystem

Based on current trends and insights by experts in Python, here are some potential future developments:

1. Continued Growth in Data Science and Machine Learning

Python's dominance in data science and machine learning is likely to persist. The ecosystem may witness the emergence of new libraries, tools, and frameworks, making it even more accessible for data scientists and machine learning practitioners.

2. Enhancements in Performance

Efforts to improve Python's performance are likely to continue. The community may explore optimizations, compiler advancements, or even alternative implementations to address performance concerns and make Python more competitive in high-performance computing scenarios.

3. Advancements in Web Development

The web development landscape will likely see further innovations with Python. Frameworks like Django and Flask may evolve to embrace new web standards, and new frameworks may emerge to cater to specific needs, fostering diversity and flexibility in the Python web development ecosystem.

4. Increased Emphasis on Sustainability

The Python community may place a greater emphasis on sustainability and maintenance. This could involve improving tools for code documentation, testing, and ensuring long-term support for packages, addressing concerns about the longevity of Python projects.

5. Further Integration with DevOps and Cloud Technologies

Python's role in DevOps and cloud technologies may expand. As cloud-native development and containerization become more prevalent, Python is likely to play a crucial role in building and managing cloud-native applications.

6. Evolution of the Python Language

The Python language itself may undergo further enhancements. Future versions of Python could introduce new features, syntax improvements, or language constructs to address evolving programming paradigms and community needs.

7. Increased Focus on Education

Python's popularity in education is likely to grow. Efforts to enhance educational resources, develop more engaging learning materials, and foster collaboration between educational institutions and the Python community may be on the horizon.

8. Advancements in AI Ethics and Responsible AI

With the increasing adoption of Python in AI and machine learning, there may be a heightened focus on ethical AI practices. The Python community may actively contribute to the development of frameworks and tools that promote responsible and ethical use of AI technologies.

9. Community-Driven Initiatives

The community's role in shaping the Python ecosystem will likely strengthen. Initiatives to promote diversity, inclusion, and community-driven decision-making may become more prominent, ensuring that the Python ecosystem remains vibrant and accessible to a diverse range of developers.