Introduction to Python

What is Python

- Python is programming language
- Python is scripting language
- Python is interpreted language
- Python is object oriented programming language
- Python is high level language

History

- Developed by Guido vano rossum in 1991
- Version 1.0 in 1994
- Version 2.0 in 2000
- Version 3.0 in 2008
- Latest version 3.13.5 released on June 11, 2025.

The Python Software Foundation (PSF) is
a nonprofit organization devoted to
the Python Programming language.

Feature	Python	Java	C	C++
Level	High-level	High-level	Low-level	Mid to high-level
Typing	Dynamically typed	Statically typed	Statically typed	Statically typed
Speed	Slowest of the four	Faster than Python, slower than C/C++	Very fast	Fast (close to C)
Memory Management	Automatic (Garbage Collection)	Automatic (Garbage Collection)	Manual	Manual or smart pointers (modern C++)
Syntax Simplicity	Very simple and readable	Verbose but clean	Very low-level and complex	Complex (especially with OOP, templates)

Object-Oriented	**Yes**	**Yes (strictly OOP)**	**No (procedural only)**	**Yes (multi-paradigm)**
Compilation	Interpreted	Compiled to byte code (JVM)	Compiled to machine code	Compiled to machine code
Portability	Very portable	Very portable (Write Once, Run Anywhere)	Low (platform dependent)	Medium to high (with effort)
Use in AI/ML	Excellent	Limited	Not suitable	Rarely used
Best For	AI/ML, data science, scripting	Web, mobile (Android), enterprise	Embedded, OS development	Game engines, performance-critical apps

Usability

Domain	Best Language
AI / Machine Learning	Python
Web development (backend)	Python or Java
Android app development	Java
System/Kernel development	C
Game development	C++
High-performance computation	C or C++
Rapid prototyping	Python
Enterprise software	Java
Learning fundamentals of programming	C or Python

other Languages for AI and Data Science

Language	Used For
R	Statistics-heavy tasks, academia
SQL	Data querying, databases
Julia	High-performance numerical computing (still niche)
Java/Scala	Some big data tools (Spark), but not for modeling
C/C++	Rare; used when performance is critical

Why python Is the First Choice for ML, AI & Data Science

1. Simplicity and Readability

Tasks that take dozens of lines in other languages often take just a few in Python.

2. Rich Ecosystem of Libraries and Frameworks

Python has a vast array of libraries that simplify complex tasks:

For Data Manipulation & Analysis:

NumPy, Pandas

For Visualization:

Matplotlib, Seaborn, Plotly

For Machine Learning:

scikit-learn, XGBoost, LightGBM

For Deep Learning:

TensorFlow, PyTorch, Keras

For Natural Language Processing (NLP):

NLTK, spaCy, Transformers (Hugging Face)

3. Community Support

Python has a massive, active community.

You’ll find thousands of tutorials, forums (like Stack Overflow), GitHub projects, and Q&A threads—essential for solving issues quickly.

4. Platform Independence

Python works on all major platforms (Windows, Linux, macOS).

Code written in Python can often run without modification across platforms, making it ideal for research and production.

5. Integration and Flexibility

Python integrates easily with other languages (C, C++, Java) and tools (Spark, Hadoop).

Supports web development, scripting, automation, and even IoT—making it a flexible all-in-one solution.

6. Adoption by Industry and Academia

Academia: Many AI/ML courses (including from Stanford, MIT, and Coursera) use Python.

Industry: Companies like Google, Facebook, Netflix, and Uber use Python extensively for AI/ML applications.

7. Rapid Prototyping

Python allows fast prototyping, which is essential in ML/AI where experimentation and iteration are key.

8. Tooling and Development Environment

Powerful IDEs and notebooks like **Jupyter**, **VSCode**, **Google Colab**, and **PyCharm** make experimentation and visualization easier.

Why python

- The IEEE ranked Python as the #1 top programming language since 2017.
- Python as a Technology Choice for Fintech company which are mostly working on big data and data science field.
- Python has a Healthy, Active and Supportive Community
- Python has Some Great Corporate Sponsors

  Google

  Red Hat

  Bloomberg

  Microsoft

- Integration with different programming language like java, C, C++
- Integration with multiple database like MySQL, Oracle, postgresql,Mongodb.
- Used for Web and general programming like software testing, networking and admin domain.
- Big data and Data science
- Business application like odoo.

Applications of Python

1. GUI-Based Desktop Applications:

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

- **Image Processing and Graphic Design Applications:**

Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Lightwave and Maya, also use Python in variable proportions.

- **Scientific and Computational Applications:**

  The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as FreeCAD, and finite element method software, such as Abaqus, are coded in Python.

- **Games:**

  Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney’s Toontown Online, Vega Strike etc.

2. Web Frameworks and Web Applications:

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP,FTP and POP. Plone-a content management system;Odoo–a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

3. Enterprise and Business Applications:

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lisp, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

4. Operating Systems:

Python is often an integral part of Linux distributions. For instance, Ubuntu’s Ubiquity Installer, and Fedora’s and Red Hat Enterprise Linux’s Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

5. Language Development:

Python’s design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple’s Swift, CoffeeScript, Cobra, and OCaml all share similarity with Python.

Where exactly is Python used at Google?

- **Google App Engine** - Python was the language where Google App Engine was originally designed for. Google App Engine is an eminent sample of Python-written application, it allows building web applications with Python programming language, using its rich collection of libraries, tools and frameworks.
- **YouTube** - is a big user of Python, the entire site uses Python for different purposes: view video, control templates for website, administer video, access to canonical data, and many more. Python is everywhere at YouTube.
- **code.google.com** - main website for Google developers.
- **open source libraries**:
  - **Google Data Python Client Library** - provides a library and source code that make it easy to access data through Google Data APIs.
  - **Google APIs Client Library for Python** - small, flexible, and powerful Python client library for accessing Google APIs.
  - **Google AdWords API Python Client Library** - makes it easier to write Python clients to programmatically access AdWords accounts.

  and many others.

Python use case in following area :-

- Testing – selenium
- Business - oodo
- Web development – django, flask
- Data science- numpy,scipy,matplotlib
- Devops –Jenkins

Python Features

- Easy to learn
- Portable and cross platform
- Object oriented
- Broad standard library
- Extendable with modules and packages
- GUI programming
- Scalable
- Multiple database support

C , Java and python code comparison

C code -

```
#include <stdio.h>
int main()
{
printf("Hello, World!");
return 0;
}
```

Java code-

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Python code -

```
print(“Hello, World!”)
```

2013–2015: Deep Learning Breakthroughs

- **Key events:**
  - Rise of **deep neural networks**, especially **convolutional neural networks (CNNs)** for image tasks.
  - **AlexNet (2012)**, although slightly earlier, sparked renewed interest in deep learning.
  - Tools like **TensorFlow** (2015) and **PyTorch** (2016) made deep learning more accessible.
- **Impact:** AI began outperforming humans in narrow tasks like image classification (e.g., ImageNet).

2016–2018: Reinforcement Learning and Language Models

- **AlphaGo (2016)** defeated human champions in Go, showcasing **reinforcement learning**.
- **Transformers (2017)** introduced by Vaswani et al. changed the future of NLP (e.g., BERT in 2018).
- **Pretrained language models** started becoming mainstream.

2019–2020: Scaling Laws and GPT Emergence

GPT-2 (2019) showed impressive text generation but was initially held back due to “misuse” concerns.

GPT-3 (2020) showed that **scaling model size** led to dramatic improvements in generalization.

Zero-shot and few-shot learning became viable without fine-tuning.

Rise in **AI ethics discussions**, bias concerns, and regulatory debates.

2021–2022: Multimodal and Commercial AI

Models like **CLIP** and **DALL·E** from OpenAI integrated **text + image understanding**.

Codex led to tools like **GitHub Copilot**, bringing AI to programming.

AI became **widely commercialized**: chatbots, voice assistants, image generators, etc.

Concerns about **misinformation**, **deepfakes**, and **AI safety** intensified.

2023–2024: Generative AI Explosion

ChatGPT, powered by GPT-3.5 and later **GPT-4**, brought conversational AI to the mainstream.

Open-source LLMs like LLaMA and Mistral surged in popularity.

Multimodal AI advanced—handling text, image, audio, and even video.

Use of AI in **education, law, healthcare, and creative arts** accelerated.

Synthetic media, **autonomous agents**, and **AI copilots** became everyday tools.

2025: Present Day

AI assistants are becoming **Popular**, integrated into apps, OSs, and workflows.

Focus is shifting toward **agentic AI**, autonomy, **real-time multimodal interaction**, and **AI alignment**.

AI is increasingly embedded in **robots, vehicles, enterprise tools**, and **content creation**.

Regulation and governance are now core global discussions (e.g., EU AI Act, U.S. policy frameworks).