

What is Python?

Python is a General-Purpose object-oriented programming language, which means that it can model real-world entities. It is also dynamically typed because it carries out type-checking at runtime.

It does so to make sure that the type of construct matches what we expect it to be.

The distinctive feature of Python is that it is an **interpreted language**.

The Python **IDLE** (Integrated Development Environment) executes instructions one line at a time. This also lets us use it as a calculator.

Why is it called Python?

Guido van Rossum named it after the comedy group Monty Python. That is why the *metasyntactic variables* (those we will often use to explain code syntax) used here are 'spam' and 'eggs' instead of 'foo' and 'bar'.

History of Python

- *Python* was conceived in the late **1980s** and was named after the *BBC TV show Monty Python's Flying Circus*.
- Guido van Rossum started implementing **Python at CWI in the Netherlands in December of 1989**.
- This was a successor to the ABC programming language which was capable of **exception handling and interfacing** with the Amoeba operating system.
- On **October 16 of 2000, Python 2.0 released** with many new features.
- Then **Python 3.0 released on December 3, 2008**.

Why Learn Python?

Python is the "most powerful language you can still read", Says Paul Dubois

Python is one of the richest Programming languages.

Going by the TIOBE Index, it is the **Second Most Popular Programming Language** in the world.

This makes a **career in Python** a great choice.

Python Features

Let us now see various features of Python that make it so powerful and popular:

a. Easy

Python is very easy to learn and understand; any beginner can learn Python easily. When writing code in Python, you need fewer lines of code compared to languages like Java.

b. Interpreted

It is interpreted(executed) line by line. This makes it easy to test and debug.

c. Object-Oriented

The Python programming language supports classes and objects and hence it is object-oriented.

d. Free and Open Source

The language and its source code are available to the public for free; there is no need to buy a costly license.

e. Portable

Since Python is open-source, you can run it on Windows, Mac, Linux or any other platform. Your programs will work without any need to change it for every machine.

f. GUI Programming

You can use it to develop a GUI (Graphical User Interface). One way to do this is through **Tkinter**.

g. Large Python Library

Python provides you with a large standard library.

You can use it to implement a variety of functions without the need to reinvent the wheel every time. Just pick the code you need and continue.

Applications of Python

Python is easy to pick-up even if you come from a non-programming background. You can look at the code and tell what's going on.

Talking of Python applications, some of the cool things that you can do are –

- Build a website using Python
- Develop a game in Python
- Perform Computer Vision (Facilities like face-detection and colour-detection)
- Implement Machine Learning (Give a computer the ability to learn)
- Enable Robotics with Python
- Perform Web Scraping (Harvest data from websites)
- Perform Data Analysis using Python
- Automate a web browser
- Perform Scripting in Python
- Perform Scientific Computing using Python
- Build Artificial Intelligence

Python is not limited to these applications. If you have ever used services from brands like YouTube, Dropbox, and Netflix, then you have been a consumer of Python.

The search-engine **Google** also made great use of this language in its initial stages.

Python Architecture and Workflow

Let us now talk about Python architecture and its usual flow –

a. Parser

It uses the source code to generate an abstract syntax tree.

b. Compiler

It turns the abstract syntax tree into Python bytecode.

c. Interpreter

It executes the code line by line in a REPL (*Read-Evaluate-Print-Loop*) *fashion*.

Python Constructs

a. Functions in Python

A **function in Python** is a collection of statements grouped under a name. You can use it whenever you want to execute all those statements at a time.

You can call it wherever you want and as many times as you want in a program. A function may return a value.

b. Classes in Python

As we discussed earlier, Python is an object-oriented language. It supports classes and objects.

A class is an abstract data type. In other words, it is a blueprint for an object of a certain kind. It holds no values.

An object is a real-world entity and an instance of a class.

c. Modules in Python

Python module is a collection of related classes and functions.

We have modules for mathematical calculations, string manipulations, web programming, and many more.

d. Packages in Python

Python package is a collection of related modules. You can either import a package or create your own.

e. List in Python

You can think of a **list** as a collection of values. Declared in the CSV (Comma-Separated Values) format and delimit using square brackets:

```
life = ['love', 'wisdom', 'anxiety']
```

```
arity = [1,2,3]
```

Notice that we do not declare the type for the list either. A list may also contain elements of different types, and the indexing begins at 0:

```
person = ['firstname', 21]
print(person[1])
```

Output: 21

You can also slice lists; slicing is a way of retrieving some values from it. We will learn more about it in further lessons.

f. Tuple in Python

A **tuple** is like a list, but it is immutable (you cannot change its values).

```
pizza = ('base', 'sauce', 'cheese', 'mushroom')
pizza[3] = 'jalapeno'
```

This raises a `TypeError`.

g. Dictionary in Python

A **dictionary** is a collection of key-value pairs. Declare it using curly braces, and commas to separate key-value pairs.

Also, separate values from keys using a colon (:).

```
student = {'Name': 'Abc', 'Age': 21}
print(student['Age'])
```

Output: 21

h. Comments and Docstrings in Python

Declare comments using an octothorpe (#). However, Python does not support multiline comments.

Docstrings are documentation strings that help explain the code.

```
#This is a comment
```

```
"""
```

```
This is a docstring
```

```
"""
```

Python has a lot of other constructs. These include control structures, functions, exceptions, etc.

Python vs Java vs C++



Python uses whitespace indentation to delimit code, you don't need to use curly braces for that.

```
a = 7
```

```
print(a)
```

While Java and C++ are statically typed, Python is dynamically typed. You also do not need to declare the type of a variable; you assign it:

```
life=42
```

Java is faster by a few seconds, but the difference does not invalidate Python's advantages over it.

Since you can interpret Python, the code is easier to test and debug.