

1. What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

2. Features of Python

Easy to Code:

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. As compared to other object-oriented programming languages like Java, C, C++, and C#, Python is one of the easiest to learn. Python is the easiest and uncomplicated language to use for the developers. It is easy to code and easy to read the language that does not take too much time.

Interpreted Language:

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners. Python is an interpreted language i.e. interpreter executes the code line by line at a time. When you use an interpreted language like Python, there is no separate compilation and execution steps. You just run the program from the source code. This makes debugging easy and thus suitable for beginners. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into

the native language of your specific computer and then runs it. You just run your programs and you never have to worry about linking and loading with libraries, etc.

Cross-platform Language:

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language. Python can run equally on different platforms such as Windows, Linux, Unix, Macintosh etc. A Python program written on a Macintosh computer will run on a Linux system and vice versa. Thus, Python is a portable language.

Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

Extensible:

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

Open Source and Free:

Python is an open-source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this Python is free to download and use in any operating system, be it Windows, Mac or Linux.

Python is readily available for everyone. It is an open-source language; its source code is accessible to the public so it can be download for free very easily. The source-code is also available. Therefore, it is open source.

Support for GUI:

GUI or Graphical User Interface is one of the key aspects of any programming language because it can add flair to code and make the results more visual. Python has support for a wide array of GUIs which can easily be imported to the interpreter, thus making this one of the most favorite languages for developers.

Object-Oriented Approach:

One of the key aspects of Python is its object-oriented approach. This basically means that Python recognizes the concept of class and object encapsulation thus allowing programs to be efficient in the long run. Python supports object oriented language and concepts of classes and objects come into existence. This programming language is an object-oriented language that focuses on objects and functions. It supports both procedures and objects-oriented programming to make the tasks easy to handle.

High-Level Language:

Python has been designed to be a high-level programming language, which means that when you code in Python you do not need to be aware of the coding structure, architecture as well as memory management.

Integrated by Nature:

Python is an integrated language by nature. This means that the python interpreter executes codes one line at a time. Unlike other object-oriented programming languages, we do not need to compile Python code thus making the debugging process much easier and efficient. Another advantage of

this is, that upon execution the Python code is immediately converted into an intermediate form also known as bytecode which makes it easier to execute and also saves runtime in the long run.

Highly Portable:

Suppose you are running Python on Windows and you need to shift the same to either a Mac or a Linux system, then you can easily achieve the same in Python without having to worry about changing the code. This is not possible in other programming languages, thus making Python one of the most portable languages available in the industry. Python is really a portable language that does not need to be changed while the transformation of coding between two systems. You can take one code to run it into different machines.

As mentioned in an earlier paragraph, Python is one of the most dynamic languages available in the industry today. What this basically means is that the type of a variable is decided at the run time and not in advance. Due to the presence of this feature, we do not need to specify the type of the variable during coding, thus saving time and increasing efficiency.

Extensive Array of Library:

Out of the box, Python comes inbuilt with a large number of libraries that can be imported at any instance and be used in a specific program. The presence of libraries also makes sure that you don't need to write all the code yourself and can import the same from those that already exist in the libraries. In python, the programmers don't have to write code for every single thing because it provides the enormously helpful feature of the large library with different functionalities.

The Python Standard Library is huge indeed. Python library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers , as well as modules written in Python that provide standardized solutions

for many problems that occur in everyday programming. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases etc.

Support for Other Languages:

Being coded in C, Python by default supports the execution of code written in other programming languages such as Java, C, and C#, thus making it one of the versatile in the industry.

It is really an active language that allows us to integrate scripting capabilities into our program of the other languages. There is also no need to specify data in python.

Python has a large standard library which provides rich set of module and functions, so you do not have to write your own code for everything. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

Dynamically Typed Language:

Python is dynamically typed language. That means the type (for example- int, double, long etc.) for a variable is decided at run time not in advance. Because of this feature we do not need to specify the type of variable.

3. Who are using Python?

The following industries are using Python:

- Industrial Light and Magic.
- Google.
- Facebook.
- Instagram.
- Spotify.
- Quora.
- Netflix.

- Dropbox

4. Where we can use python?

Python supports cross-platform operating systems which makes building applications with it more convenient. Some of the globally known applications such as YouTube, BitTorrent, DropBox, etc. use Python to achieve their functionality.

a) Web Development

Python can be used to make web-applications at a rapid rate. Why is that? It is because of the frameworks Python uses to create these applications. There is common-backend logic that goes into making these frameworks and a number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more. Some of the most well-known frameworks are Django, Flask, Pyramid. The frameworks are used for security, scalability, convenience that they provide is unparalleled compared to starting the development of a website from scratch.

b) Game Development

Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3, PyGame which provides functionality and a library for game development. Games such as Civilization-IV, Disney's Toontown Online, Vega Strike etc. have been built using Python.

c) Machine Learning and Artificial Intelligence

Machine Learning and Artificial Intelligence are the talks of the town as they yield the most promising careers for the future. We make the computer learn based on past experiences through the data stored or better yet, create algorithms which makes the computer learn by itself. The

programming language that mostly everyone chooses is Python because Support for these domains with the libraries that exist already such as Pandas, Scikit-Learn, NumPy and so many more.

d) Data Science and Data Visualization

Data is money if you know how to extract relevant information which can help you take calculated risks and increase profits. You study the data you have, perform operations, and extract the information required. Libraries such as Pandas, NumPy help you in extracting information. You can even visualize the data libraries such as Matplotlib, Seaborn, which are helpful in plotting graphs and much more. This is what Python offers you to become a Data Scientist.

e) Desktop GUI

Python can be used to program desktop applications. It provides the Tkinter library that can be used to develop user interfaces. There are some other useful toolkits such as the wxWidgets, Kivy, PYQT that can be used to create applications on several platforms.

You can start out with creating simple applications such as Calculators, To-Do apps and go ahead and create much more complicated applications.

f) Web Scraping Applications

Python can be used to pull a large amount of data from websites which can then be helpful in various real-world processes such as price comparison, job listings, research and development and much more.

5. One company use-case about python

ForecastWatch.com, a service of Intellovations, is in the business of rating the accuracy of weather reports from companies such as Accuweather, MyForecast.com, and The Weather Channel. Over 36,000 weather forecasts are collected every day for over 800 U.S. cities, and later compared with actual climatological data. These comparisons are used by meteorologists to improve their weather

forecasts, and to compare their forecasts with others. They are also used by consumers to better understand the probable accuracy of a forecast.

ForecastWatch.com is built from four major architectural components: An input process for acquiring forecasts, an input process for acquiring measured climatological data, the data aggregation engine, and the web application framework.

There are two main input processes in the system: The forecast parser, and the actuals parser. The forecast parser is responsible for requesting forecasts from the web for each of the forecast providers ForecastWatch.com tracks. It parses the forecast from the page and inserts the forecast data into a database until it can be compared to the actual data. The actuals parser takes actual data from the National Climatic Data Center of the National Weather Service, which provides high, low, precipitation, and significant weather events for over 800 United States cities and inserts the data into the database. This process also scores the forecasts with the actual weather data, and places that information in the database.

Once the data has been collected and scored, it is processed by the aggregation engine, which combines the scores into yearly and monthly blocks, sliced by provider, location, and the number of days into the future for which the forecasts were predicting. In its first year, 2003, the system only gathered forecasts for 20 U.S. cities, or about 250,000 individual forecasts, so most of the data output was based on the raw scoring data. The aggregation engine was added once the system was scaled up to 800 cities, increasing the data stream by almost 4000%. In the first half of 2004, the system has already scored over 4 million forecasts, all collected, parsed, and displayed on the web.

The last component in ForecastWatch.com's architecture is the website itself. This is the interface through which customers access the collected and aggregated forecast accuracy information.

ForecastWatch.com is a 100% pure Python solution. Python is used in all its components, from the back-end to the front-end, including also the more performance-critical portions of the system. Python was chosen initially because it comes with many standard libraries useful in collecting, parsing, and storing data from the web. Among those particularly useful in this application were the regular expression library, the thread library, the object serialization library, and gzip data compression library. Other libraries, such as an HTTP client capable of accepting cookies (ClientCookie), and an HTML table parser (ClientTable) were available as third party modules. These proved invaluable and were easy to use.

The threading library turned out to be very important in scaling ForecastWatch.com's coverage to over 800 cities. Grabbing web pages is a very I/O bound process and requesting a single page at a time for roughly 5000 web pages a day would have been prohibitively time-consuming. Using Python's threading library, the web page retrieval loop simply calls `thread.start_new()` for each request, passing in the necessary class instance method that retrieves and processes the web page, along with the parameters necessary to describe the city for the desired forecast. The request classes use a Python built-in Event class instance to communicate with the main controlling thread when processing is complete. Python made this application of threading incredibly easy.

Python is also used in the aggregation engine, which runs as a separate process to combine forecast accuracy scores into monthly and yearly slices. The aggregation process uses queries via MySQLdb to the MySQL database where the input modules have placed the forecast and climatological data they have harvested. Colorized maps, showing forecast accuracy by geographical area, are then generated for use on the web site and in printed reports.

ForecastWatch.com's web interface was originally written in PHP but later changed to Python to simplify the toolset and improve integration with the other components of the system. Quixote, a

Python web application framework, was selected as the basis for the entirely Python-based web front-end. The Quixote-based web application runs on Linux using Apache with mod_scgi, and was able to serve pages as fast as the PHP-based implementation. Python made it easier to make changes and add features than the PHP implementation.