**KEY FEATURES OF PYTHON :**

* Easy to code and learn
* Free and Open Source with a[Python Software Foundation License](https://en.wikipedia.org/wiki/Python_Software_Foundation_License)
* Object-Oriented Language
* Dynamically Typed Language
* GUI Programming Support
* High-Level Language
* Extensible Language
* Portable Language
* Multi-platform Language
* Interpreted Language
* Large Standard Library
* Documentation readability is supported
* Novice or non-coding background friendly
* Less complex as compared to C++ and Java
* Readability of Python code is more since it resembles actual English
* Less lines of code and time efficient compared to other programming or scripting languages
* Python shines as a glue language, used to combine components written in C++/Java.
* Python can be used as a scripting language as well as a programming language

**LIMITATIONS :**

* Python is way *slower than Go(Golang)* as a backend tech because the “Go” supports concurrency as well as would work with GRPC protocol on http2.
* Despite of using Python for developing math models, C++’s Dlib could be used for custom neural networks since C++ holds the title for the “fastest programming language.
* Unlike C++ and Java, python works with the help of an interpreter which makes compilation and execution slower in AI development.
* Not suitable for mobile computing. For AI meant for mobile applications, Python unsuitable due to its weak language for mobile computing.
* Python code is typically 3-5 times shorter than equivalent Java code, it is often 5-10 times shorter than equivalent C++ code
* One who is custom code builder would rarely opt python libs because it reduces the logic level of coders since it provides great functionality and libraries to interact with, with few lines of code.
* Python's structures demand more memory space. This language is not suitable to use for development under limited memory restrictions.
* Developer's Restrictions - Once a developer gets used to the ease and simplicity of this language, it becomes difficult for them to switch back to other languages.
* As compared to other popular technologies such as JDBC and ODBC, the Python database access layer is a little underdeveloped and primitive. It is, therefore, not considered suitable if developers are looking for a smooth interaction of complex legacy data.
* Problems with threading - Threading is not really good in Python due to the Global Interpreter Lock (GIL). GIL is simply a mutex that allows only one thread to execute at a time. As a result, multi-threaded CPU-bound programs may be slower than single-threaded ones. Luckily there’s a solution for this problem. - We need to implement multiprocessing programs instead of multithreaded ones. That's what we often do for data processing.

**CONCLUSION :**

* Hence, the real world solutions are build using many different popular libraries and languages linked to combine and give away the efficient output of the projects that are under construction.