Python Programming

Introduction
to
Python Programming

Outline

- History of Python
- Features of Python
- Applications of Python
- Limitations of Python
- Comparison of Python and JAVA
- Installing Python
- Executing Python Programs
- Internal Working of Python
- Python Implementations

History of Python

- Python is a general-purpose interpreted, interactive, object-oriented and high-level programming language.
- It is created by Guido van Rossum and first released in 1991 at Centrum Wiskunde & Informatica (CWI), the National Research Institute for Mathematics and Computer Science in the Netherlands.
- Rossum named it python inspired by British comedy series “Monty Python’s Flying Circus”.
- Rossum developed Python as a hobby.

History of Python

- Python is a real-world, production language that is freely available for most computers.
- Python source code is available under the GNU General Public License (GPL).
- Python uses an interpreter in a statement by statement mode enabling users to experiment quite easily
- Python has become a popular programming language widely used in industry and academia due to its simple, concise, and intuitive syntax and extensive library.
- Official Website for python: http:www.python.org

History of Python

- Python is now being developed and maintained by a large team of volunteers and is available for free from the Python Software Foundation.
- Two versions of Python are currently coexistent: Python 2 and Python 3.
- The programs written in Python 3 will not run in Python 2.
- Python 3 is a newer version, but it is not backward-compatible with Python 2. This means that if you write a program using the Python 2 syntax, it may not work with a Python 3 interpreter.
- Python provides a tool that automatically converts code written in Python 2 into syntax Python 3 can use.
- Python 2 will eventually be replaced by Python 3.
- Python releases by version number :https://www.python.org/downloads/

Features of Python

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes. It is easier for others to understand your code faster. Readability usually helps a language easy to learn.
- **Easy-to-maintain** – Maintaining source code is part of the software development lifecycle. Your software usually continues to evolve until it is replaced or obsolete. Python's source code is fairly easy-to-maintain.
- **Object Oriented** - It supports functional and structured programming methods as well as OOP. Python is not *just* an OO language like Java. It is actually a pleasant mix of multiple programming paradigms.

Features of Python

- **Interpreted**
- Python is Interpreted. It is processed at runtime by the interpreter. You do not need to compile your program before executing it. the interpreter processes the code line by line and creates a byte code. Bytecode is an in-between "language" that is not quite machine code but it is not the source code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms. Bytecode is more transferable between operating systems than machine code, this helps Python be cross-platform.

Features of Python

- **Robust**
- Python provides safe exits on errors. When your Python crashes due to errors, the interpreter dumps out a "stack trace" full of useful information such as why your program crashed and where in the code (file name, line number, function call, etc.) the error took place. Python even gives you the ability to monitor for errors and take an action if an error occur during runtime.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting. It can be used as a scripting language or can be compiled to byte-code for building large applications. Python encourages clean code design, high-level structure and "packaging" of multiple components, all of which deliver the flexibility, consistency and faster development time required as projects expand in breadth and scope.

Features of Python

- **Extensible**
- Extensibility in a language provides engineers with the flexibility to add-on or customize their tools to be more productive and to develop in a shorter period of time. Python extensions can be written in C and C++ for the standard implementation of Python in C (CPython). Java language implementation of Python is called Jython, so extensions would be written using Java. There is IronPython, the C# mplementation for the .NET or Mono platforms. You can also extend IronPython in C# or Visual Basic.NET.
- **Interactive Mode**
- Python has support for an interactive mode which allows interactive testing and debugging of snippets of code. It provides an interactive command line and interpreter shell.

Features of Python

- **GUI Programming**
- Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix. It has vast, powerful standard library.
- **Garbage Collection**
- It supports automatic garbage collection. This leads to fewer bugs, a more robust application and shorter overall development time.
- **Developer Productivity**
- It provides very high-level dynamic data types and supports dynamic type checking. Types are bound to values, not to variables. The size of code is reduced quite a bit so debugging is very easy.
- **Databases**
- Python provides interfaces to all major commercial databases.

Applications of Python

- Python is used in many application domains.
- **Web Development**
- Python is an easily extensible language that provides good integration with database and other web standards. Python's standard library supports many Internet protocols.
- **Scientific and Numeric Computing**
- Features like high speed, productivity and availability of tools, such as Scientific Python and Numeric Python, have made Python a preferred language to perform computation and processing of scientific data. 3D modeling software, such as FreeCAD and finite element method software, like Abaqus, are coded in Python.

Applications of Python

- **Education**
- Python is a superb language for teaching programming, both at the introductory level and in more advanced courses.
- **Entertainment**
- Python has various modules, libraries, and platforms that support development of games. Games like Civilization-IV, Disney's Toontown Online, Vega Strike, etc. are coded using Python.
- **Desktop GUI**
- Simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems makes Python a preferred choice for developing desktop-based applications.

Applications of Python

- **Software Development**
- Python is often used as a support language for software developers, for build control and management, testing, and in many other ways. Python is used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. It is also used to make 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya.
- **Business Applications**
- Python is also used to build ERP and e-commerce systems. It is a suitable coding language for customizing larger applications. A large part of Youtube code is also written in Python. Python also forms an integral part of Linux distributions

Limitations of Python

- Python has varied advantageous features and programmers prefer this language to other programming languages because it is easy to learn and code too. limitations of Python are:
- **Weak in Mobile Computing**
- Python has made its presence on many desktop and server platforms, but it is seen as a weak language for mobile computing. Python is not a very good language for mobile development**.**
- **Speed**
- Python executes with the help of an interpreter instead of the compiler, which causes it to slow down because compilation and execution help it to work normally. On the other hand, it can be seen that it is fast for many web applications too.

Limitations of Python

- **Memory Consumption**
- Python is not a good choice for memory intensive tasks. Due to the flexibility of the data-types, Python's memory consumption is also high. It is not a great choice for a high-graphic3d game that takes up a lot of CPU.
- **Runtime Errors**
- Python programmers cited several issues with the design of the language. Because the language is dynamically typed, it requires more testing and has errors that only show up at runtime.
- **Underdeveloped Database Access Layers**
- As compared to the popular technologies like JDBC and ODBC, the Python’s database access layer is found to be bit underdeveloped and primitive. However, it cannot be applied in the enterprises that need smooth interaction of complex legacy data.

Comparison of Python and Java

- **Similarities to Java**
- Object Oriented Programming languages.
- Compiled to byte code in Java and interpreted to Bytecode in python
- Vast, Powerful Standard Library
- Case sensitive
- Portability
- Garbage Collection
- Exception Handling
- Multithreading
- Serialization

Comparison of Python and Java

- **Difference Between Python and Java**
- Java supports only object-oriented programming. It is possible to write a Python program without making any use of OO concepts.

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

The same python program looks like this:

```
print("Hello World!")
```

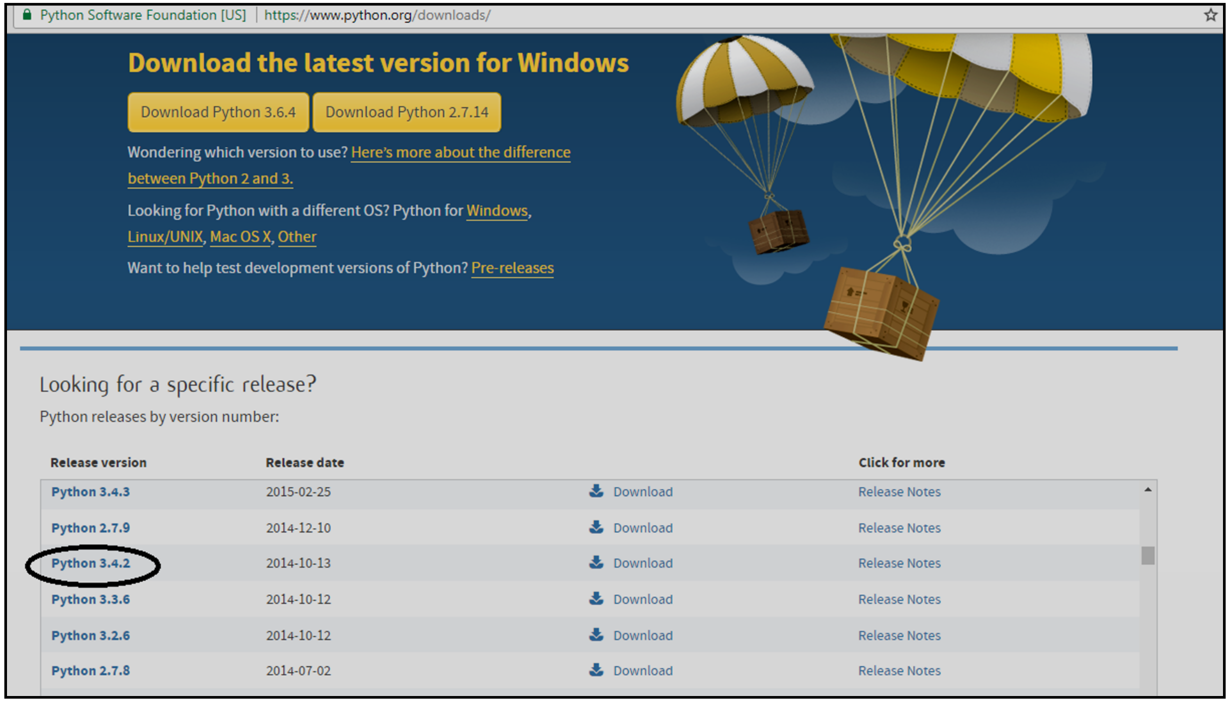
- Statements in Java always end with a semicolon (;). Python is line-oriented: statements end at the end of a line unless the line break is explicitly escaped with \.

Comparison of Python and Java

- Java is statically typed. i.e. int a = 42;
  Python is dynamically typed i.e. a=42.
- Programs written in Java must be explicitly compiled into bytecodes (.class files), though an IDE may do this automatically in a way that is transparent to the user. Python is designed to be used interpretively. A Python statement may be entered at the interpreter prompt (>>>), and will be executed immediately.
- Java uses traditional braces to start and end blocks, while Python uses indentation.
- Most programmers agree that Python is an easier language for novice programmers to learn. Python is simpler and more compact compared to Java.

Installing Python

- The Python programming language runs on almost any system such as Linux, Macintosh, and Windows. The distributions are made available by the core team on the main download page of the Python website at: https://www.python.org/downloads/



Python Software Foundation [US] | https://www.python.org/downloads/

Download the latest version for Windows

Download Python 3.6.4
Download Python 2.7.14

Wondering which version to use? Here's more about the difference between Python 2 and 3.

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases

Looking for a specific release?

Python releases by version number:

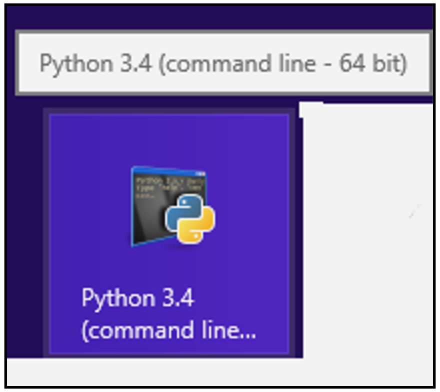
Release version	Release date		Click for more
Python 3.4.3	2015-02-25	Download	Release Notes
Python 2.7.9	2014-12-10	Download	Release Notes
Python 3.4.2	2014-10-13	Download	Release Notes
Python 3.3.6	2014-10-12	Download	Release Notes
Python 3.2.6	2014-10-12	Download	Release Notes
Python 2.7.8	2014-07-02	Download	Release Notes

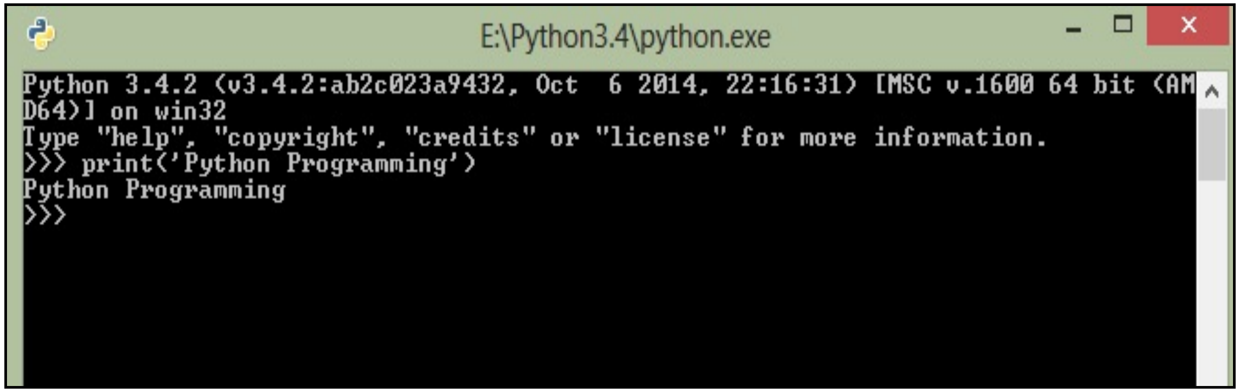
Installing Python

- Download the .msi file from python.org for Windows or a stable version of python i.e. Python 3.4.2
- If you have a Mac or Linux, you may already have Python on your computer, though it may be an older version.
- Python is usually installed in a standard location so you can find it rather easily.
- Python also has mechanisms to allow a python program to act both as a script and as a module to be imported and used by another python program

Executing Python Programs

- There are different ways to start executing Python.
- **Interactive Interpreter from the Command Line (Python- command line)**
- The simplest way is by starting the interpreter interactively, entering one line of Python at a time for execution.



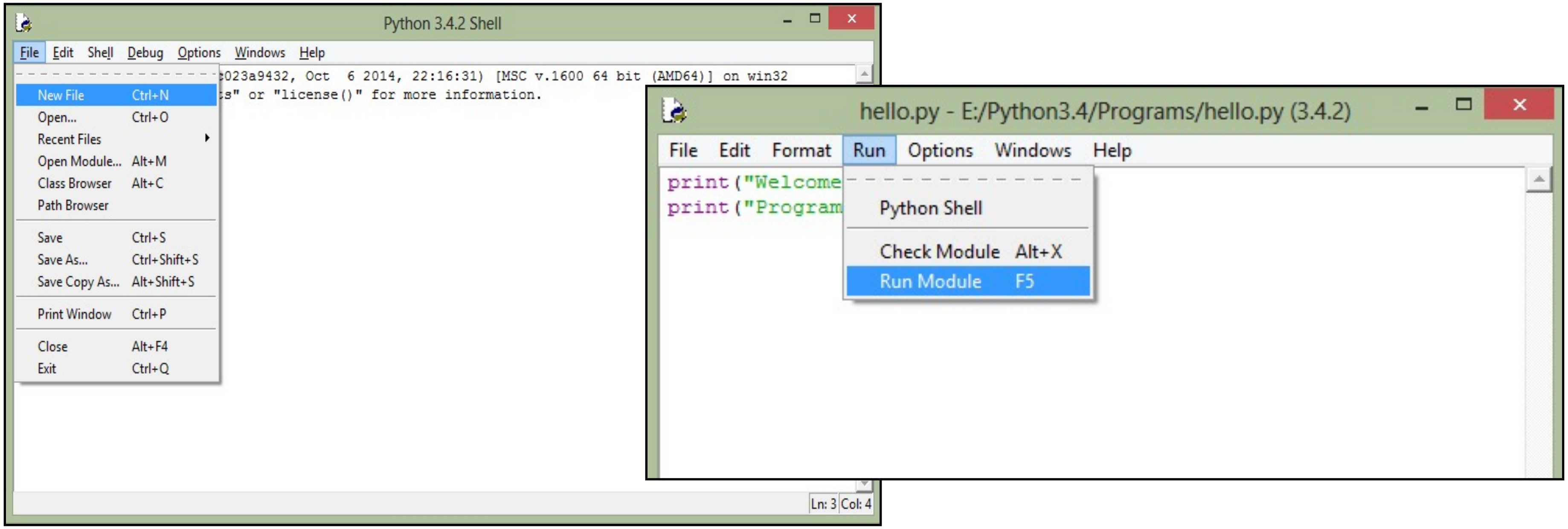


```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  6 2014, 22:16:31) [MSC v.1600 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Python Programming')
Python Programming
>>>
```

Executing Python Programs

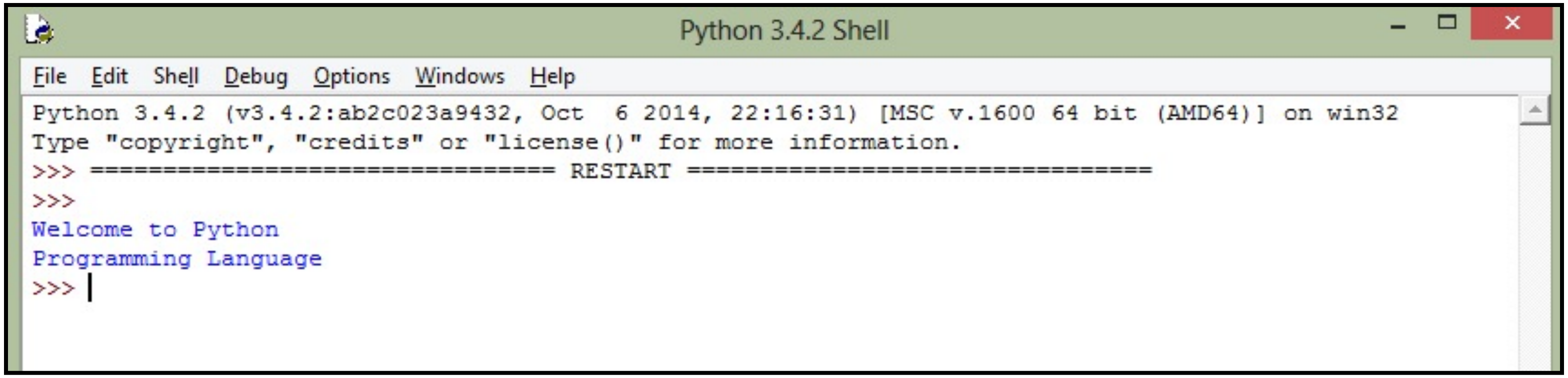
- **Integrated Development Environment (IDLE -Python GUI)**
- You can run Python from a graphical user interface (GUI) environment as well.
- *IDLE(Interactive Development Environment) is an* integrated development environment (IDE) for Python. You can create, open, save, edit,and run Python programs in IDLE.
- Both the command-line Python interpreter and IDLE are available after Python is installed on your machine.Save it as a file with the filename having the extension .py.
- Run the interpreter with the command python program_name.py or use IDLE to run the programs.If you want to execute the program in Python shell, then just press F5 key or click on Run Menu and then select Run Module.

Executing Python Programs



```
:023a9432, Oct  6 2014, 22:16:31) [MSC v.1600 64 bit (AMD64)] on win32
s" or "license()" for more information.
```

```
print("Welcome
print("Program
```



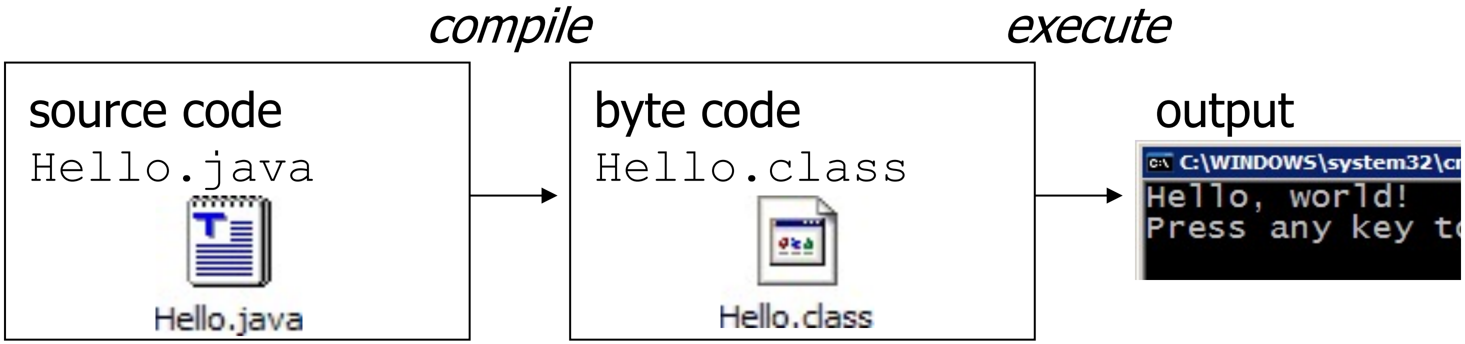
```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  6 2014, 22:16:31) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Welcome to Python
Programming Language
>>> 
```

Executing Python Programs

- Most common syntax colors and their meanings in IDLE.
- **Common Python syntax colors:**
  - Keywords : orange
  - Strings : green
  - Comments : red
  - Definitions: blue
  - Misc. Words: black

Internal Working of Python

- Many languages require you to *compile* (translate) your program into a form that the machine understands.



compile *execute*

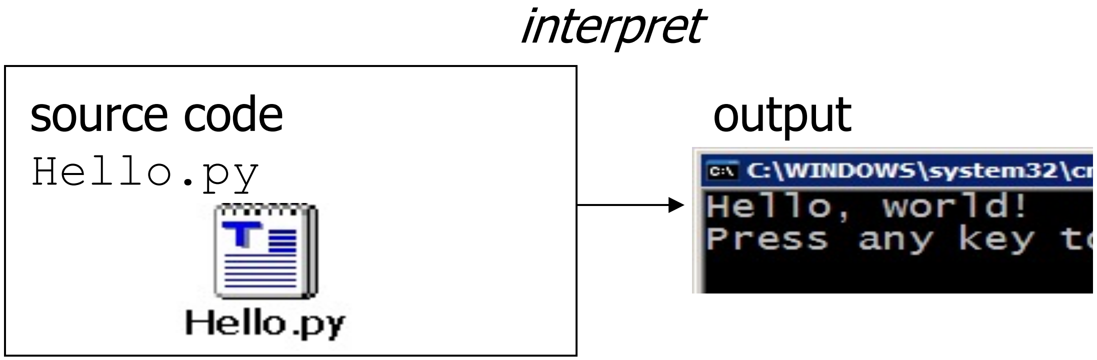
source code
`Hello.java`

byte code
`Hello.class`

output

```
Hello, world!
Press any key to
```

- Python is instead directly *interpreted* into machine instructions and the interpreter actually has the structure of a classic compiler.



interpret

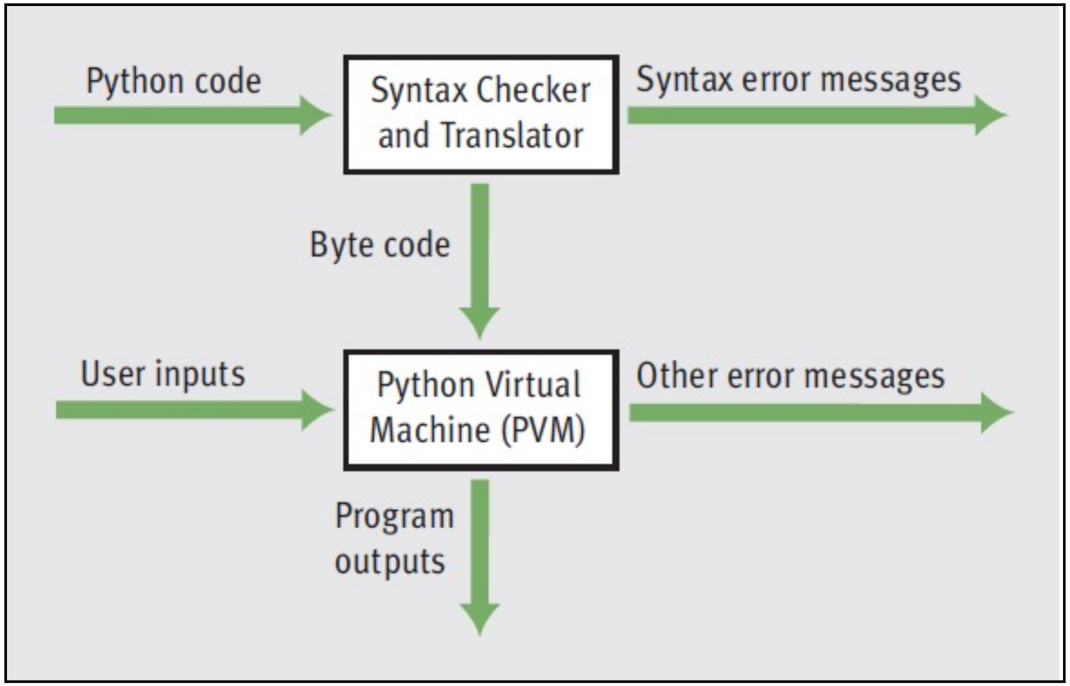
source code
`Hello.py`

output

```
Hello, world!
Press any key to
```

Internal Working of Python

- Whether you are running Python code as a script or interactively in a shell, the Python interpreter does a great deal of work to carry out the instructions in your program. This work can be broken into a series of steps as



Internal Working of Python

- Interpreter reads a Python expression or statement also called the source code and verifies that it is well formed. In this step, the interpreter behaves like a strict English teacher who rejects any sentence that does not adhere to the grammar rules, or syntax, of the language. As soon as the interpreter encounters such an error, it halts translation with an error message.
- If a Python expression is well formed, the interpreter then translates it to an equivalent form in a low-level language called byte code. When the interpreter runs a script, it completely translates it to byte code.
- Byte code is next sent to another software component, called the Python virtual machine (PVM), where it is executed. If another error occurs during this step, execution also halts with an error message.
- If there is no error, byte code is executed by the machine.

Python Implementations

- *Python* is actually a specification for a language that can be implemented in many different ways.
- **CPython**
- The main Python implementation is written in the C language and is called **CPython.** It is the one that majority of people refer to, when they talk about Python. CPython provides the highest level of compatibility with Python packages and C extension modules.
- If you are writing open-source Python code and want to reach the widest possible audience, targeting CPython is best.

Python Implementations

- **Jython**
- Jython is a Java implementation of the language. Jython is a Python implementation that compiles Python code to Java bytecode which is then executed by the JVM (Java Virtual Machine). Additionally, it is able to import and use any Java class like a Python module.
- If you need to interface with an existing Java codebase or have other reasons to need to write Python code for the JVM, Jython is the best choice.

Python Implementations

- **IronPython**
- IronPython brings Python into .NET.
- It is an implementation of Python for the .NET framework. It can use both Python and .NET framework libraries, and can also expose Python code to other languages in the .NET framework.
- Python Tools for Visual Studio integrates IronPython directly into the Visual Studio development environment, making it an ideal choice for Windows developers.

Python Implementations

- **PyPy**
- PyPy is probably the most exciting implementation, as its goal is to rewrite Python into Python. In PyPy, the Python interpreter is itself written in Python.
- PyPy is a Python interpreter implemented in a restricted statically-typed subset of the Python language called RPython. The interpreter features a just-in-time compiler and supports multiple back-ends (C, CLI, JVM).
- PyPy aims for maximum compatibility with the reference CPython implementation while improving performance.
- If you are looking to increase performance of your Python code, it’s worth giving PyPy a try. On a suite of benchmarks, it’s currently over 5 times faster than CPython.

Python Implementations

- Alternative Python Implementations
- **Stackless Python** (Branch of CPython supporting microthreads, thread-based programming)
- **MicroPython** (Python running on micro controllers)
- **PythonAnywhere** (To host and run Python in the cloud, freemium hosted Python installation which lets you run Python in the browser)
- **Anaconda Python** (a full Python distribution for data management, analysis and visualization of large data sets)
- **winpython** (WinPython is a portable scientific Python distribution for Windows)
- **Conceptive Python SDK** (targets business, desktop and database applications)

THANKS.....