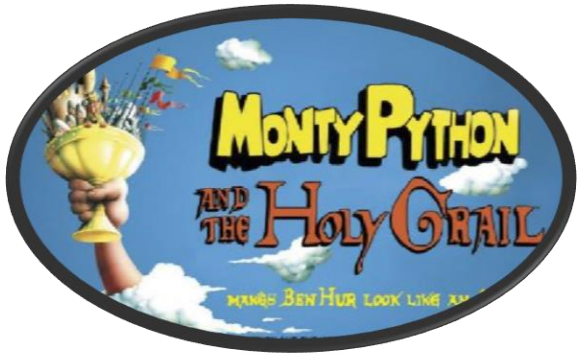
CMPE 102
Programming Logic and Design

Module 3
Welcome to
Python Programming

What is Python?

Python is a powerful multi paradigm computer programming language, optimized for programmer productivity, code readability, and software quality.

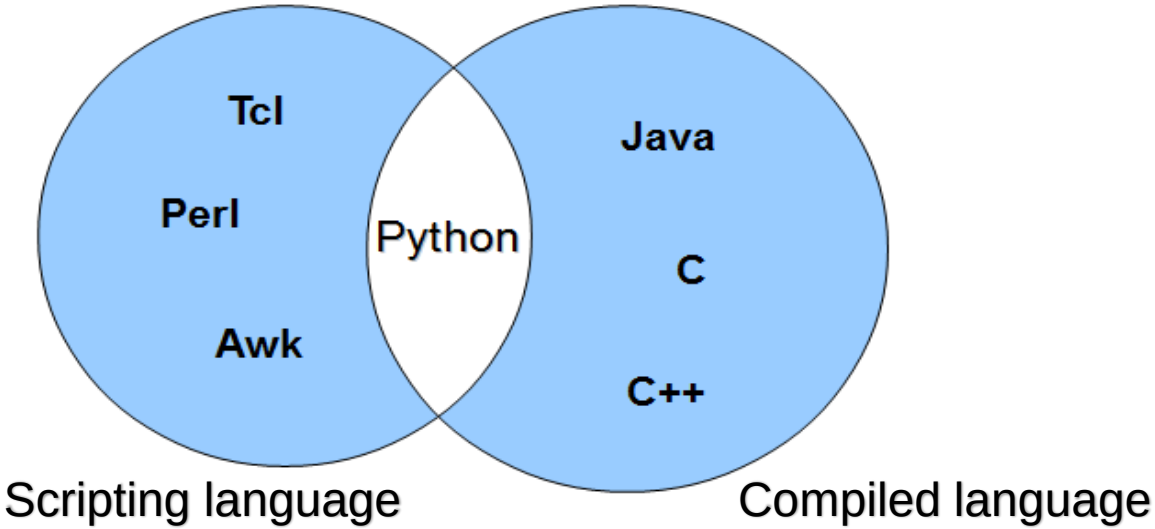
Guido Van Rossum



Python is named after famous British comedy group

Source code is available under the GNU General Public License (GPL).

Python: a Complete Language



Python provides simplicity and ease of use of scripting Language, along with more advanced software- engineering tools typically found in compiled languages which makes Python useful for large-scale development projects.

Features of Python

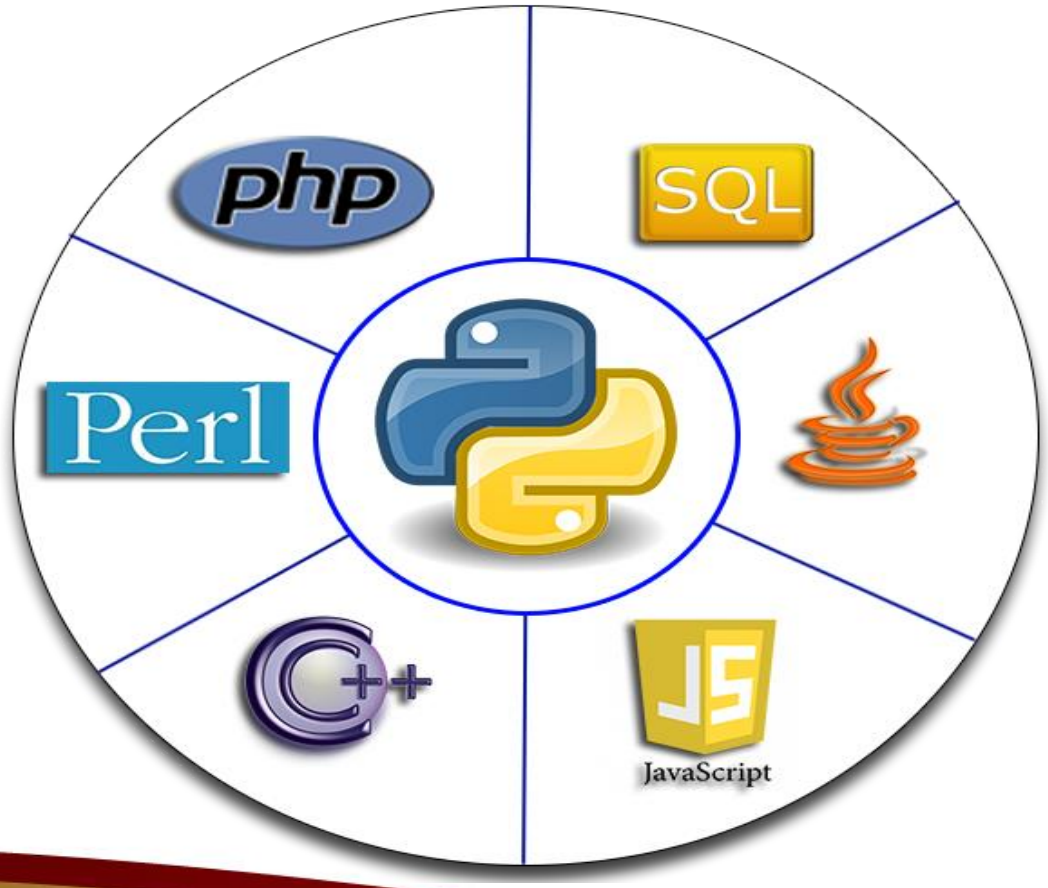
Portable

Python compiles and runs on every major platform currently in use. Python can be used on 21 different operating systems and environments.

Powerful

Python's toolset places it between traditional scripting languages (Tcl, Scheme and Perl) and systems development languages (C ,C++ and Java)

Mixable



Python programs can be easily glued to components written in other languages.

Ease of Syntax

– vs –

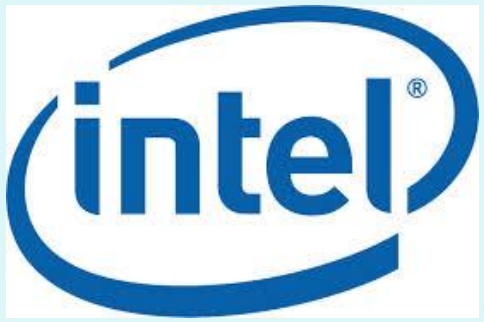
Java Code to print “Hello World” :

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

Python code to print “Hello World”:

```
print  “Hello, World”
```

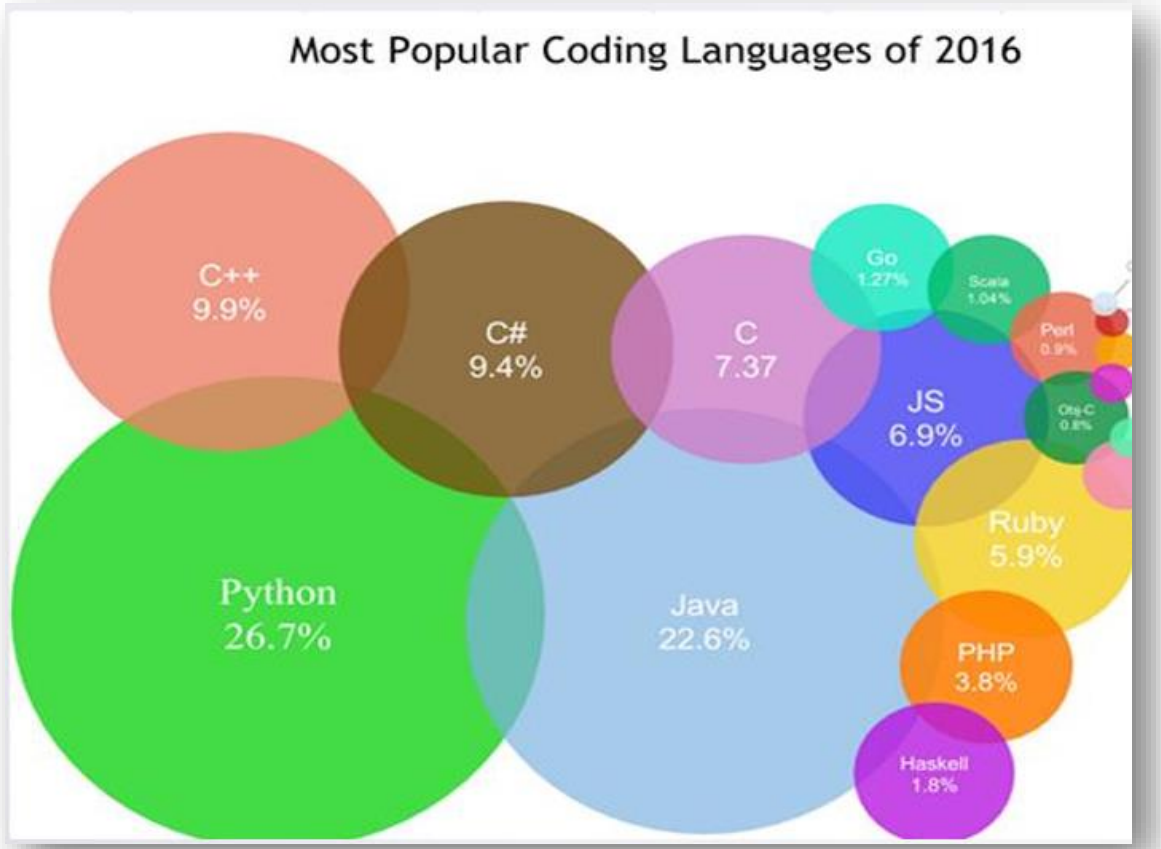
Big Names in Python's Users



Why Python?

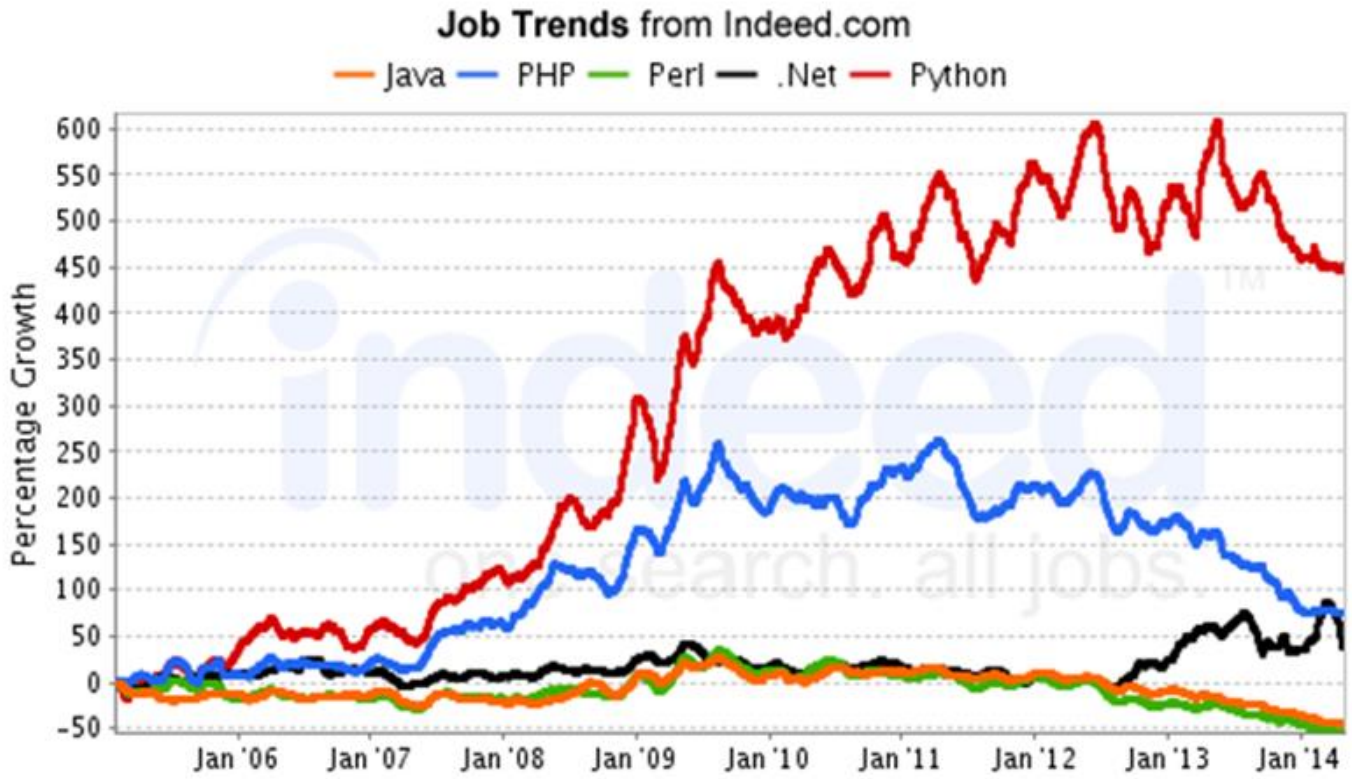
	C	C++	Java	Python
Object oriented	No	Yes	Yes	Yes
Functional	No	Yes	No	Yes
Type safety	Unsafe	Unsafe	Safe	Safe
Type expression	Explicit	Explicit	Explicit	Implicit
Type checking	Static	Static	Static	Dynamic
Failsafe I/O	No	No	Yes	Yes
Readability	Difficult	Difficult	Difficult	Easy
Learning	Difficult	Difficult	Difficult	Easy
Language	Programming	Programming	Programming	Programming and scripting
Length of code	5-10 times greater than python	5-10 times greater than python	3-5 times greater than python	Small and manageable codes

Scope of Python



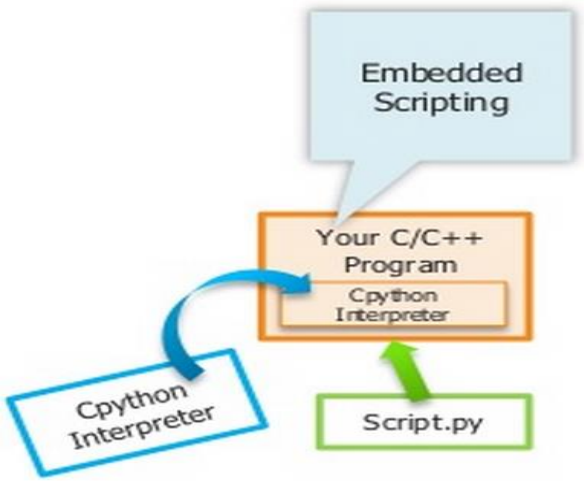
Python is most demanded language currently as shown by surveys.

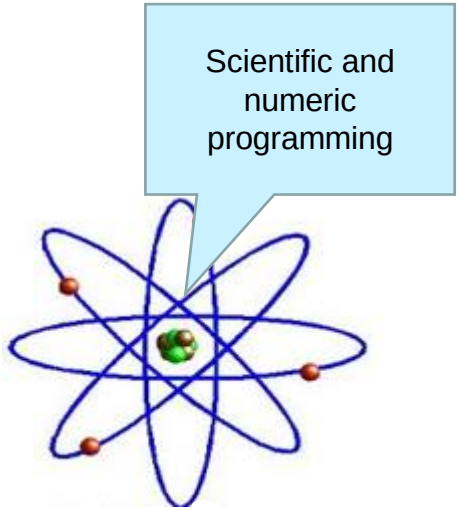
What Job Trends Say



The graph here shows the job trends in different languages with Python having the maximum percentage growth.

Major uses of Python





GUIs

Camelot provides components for building applications on top of Python, SQLAlchemy and Qt. inspired by the Django admin interface

GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off toolsto complete application suites.

Kivy is a Python library for development of multi-touch enabled media rich applications. The aim is to allow for quick and easy interaction design and rapid prototyping, while making your code reusable and deployable.

Tkinter has the advantage of being included with the Python standard library, making it the most convenient and compatible toolkit to program with.

PyQt is a set of Python v2 and v3 bindings for Digia's Qt application framework and runs on all platforms supported by Qt including Windows, MacOS/X and Linux.

pyjs is a Rich Internet Application (RIA) Development Platform for both Web and Desktop. With pyjs you can write your JavaScript-powered web applications entirely in Python.

wxPython is a GUI toolkit for the Python programming language. It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily

What Is Python?

- Python is an elegant and robust programming language that delivers both the power and general applicability of traditional compiled languages with the ease of use of simpler scripting and interpreted languages.
- Python originated in late 1989 at the National Research Institute for Mathematics and Computer Science in the Netherlands by Guido van Rossum, with its first public release coming in early 1991.
- Van Rossum was doing research in system administration at the time and found most conventional programming languages too cumbersome or incomplete to perform the work he envisioned. He was working on automating system administration tasks and thus needed access to the power of system level calls.

What Is Python?

- Van Rossum was working on an Amoeba distributed operating system at the time and gave serious consideration to developing an Amoeba-specific language to further his research. In the end he decided to go with a generalized language and Python was born.
- Although Python has now been around for about 20 years, many people feel that it is still relatively new to the general software development industry.
- The next few pages will illustrate some of the primary features of Python and why it has become a valuable tool for system administrators.

History of Python

- Created in 1989 by Guido van Rossum
  - Created as a scripting language for administrative tasks
  - Based on *All Basic Code (ABC)* and *Modula-3*
    - Added extensibility
  - Named after comic troupe Monty Python
- Released publicly in 1991
  - Growing community of Python developers
  - Evolved into well-supported programming language

History of Python

- Modules
  - Reusable pieces of software
  - Can be written by any Python developer
  - Extend Python's capabilities
- Python Web site at www.python.org
  - Primary distribution center for Python source code, modules and documentation

Downloading And Installing Python

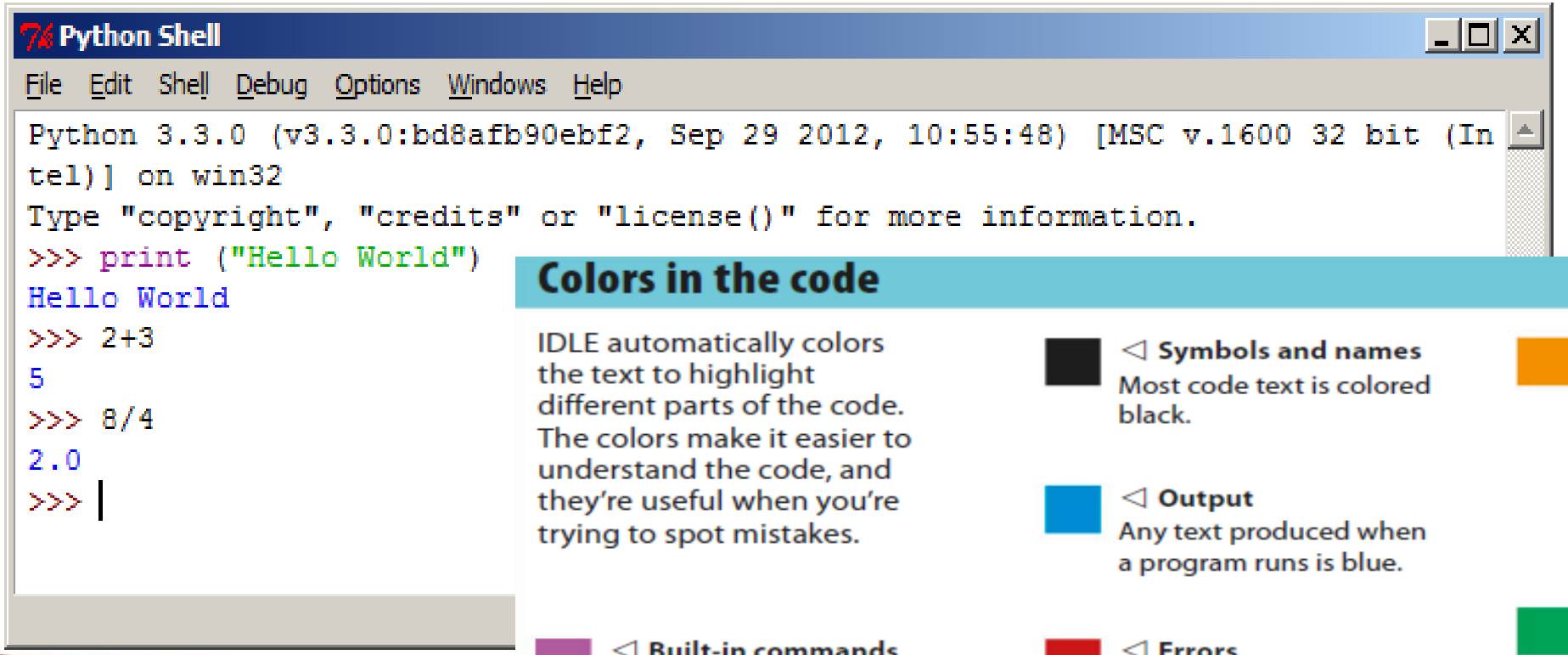
- The most obvious place to get all Python-related software is at http://python.org.
- Right now there are two current production versions of Python available, versions 2.7.3 and 3.3.0. This is a common occurrence with Python and if in doubt about which version to download, the older version will almost always have more compatibility with third party software than the newer version which requires some lead time before third party vendors can catch up with new developments in the language.

Running Python

- Download your preferred version of Python and install it accordingly on your system.
- Once you've downloaded Python there are several different ways to start it running.
- The simplest way is to start the interpreter interactively, entering one line of Python at a time for execution
- The next way is to run a script written in Python by invoking the interpreter on the script. To do this, first create the script file using a text editor, then simply click on the script file to execute it.

Running Python

- Another way to run Python is through an IDE.
- Current versions of Python come with an IDE called IDLE (Python GUI), which looks like:



Colors in the code

IDLE automatically colors the text to highlight different parts of the code. The colors make it easier to understand the code, and they're useful when you're trying to spot mistakes.

◁ **Symbols and names**
Most code text is colored black.

◁ **Keywords**
Certain words, such as "`if`" and "`else`", are special words that Python uses. They are called keywords and are shown in orange.

◁ **Output**
Any text produced when a program runs is blue.

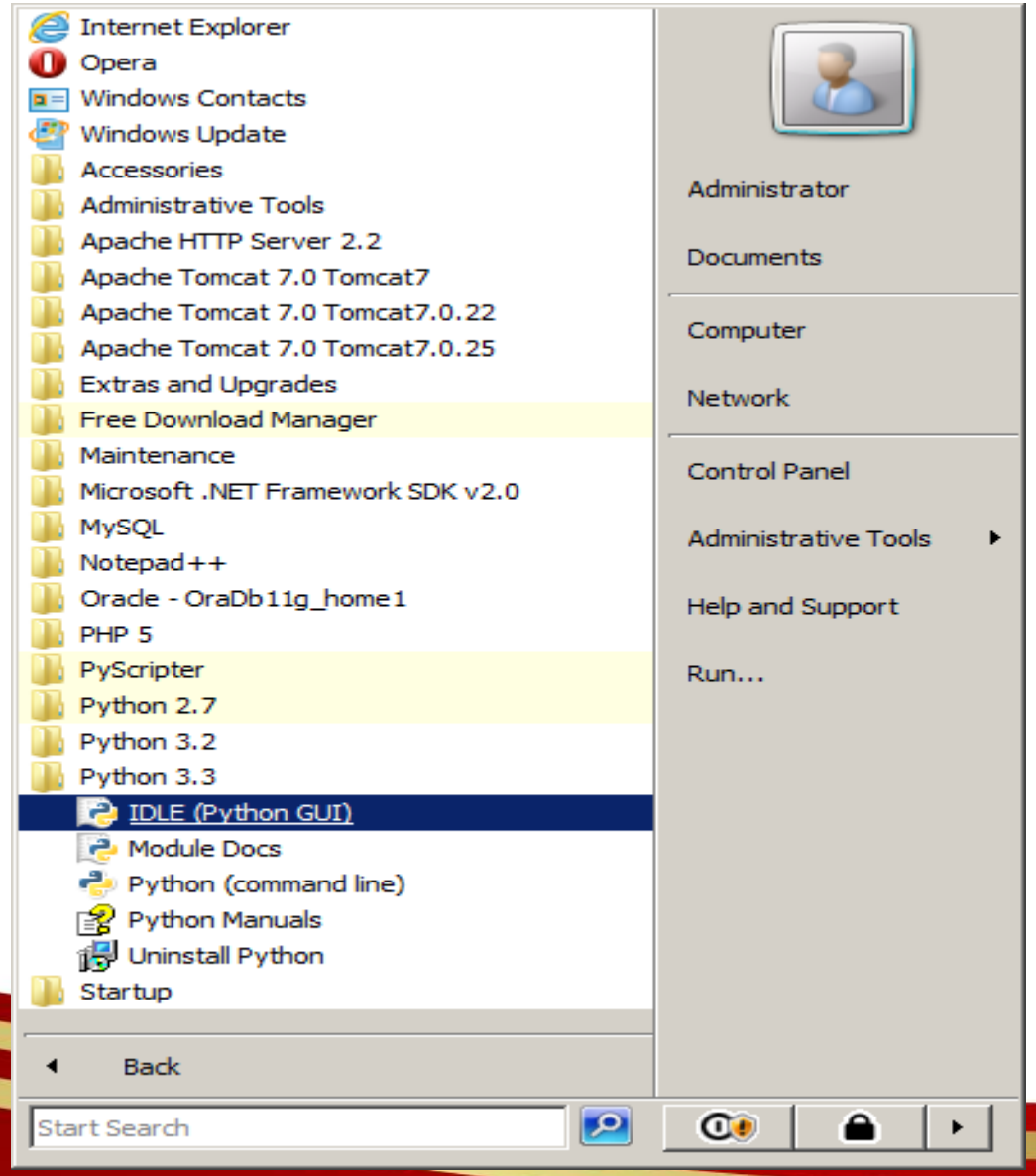
◁ **Built-in commands**
Python commands, such as "`print`", are shown in purple.

◁ **Errors**
Python uses red to alert you to any errors in your code.

◁ **Text in quotes**
Text in quote marks is green. A green bracket around text shows you're missing a quote mark.

Running Python

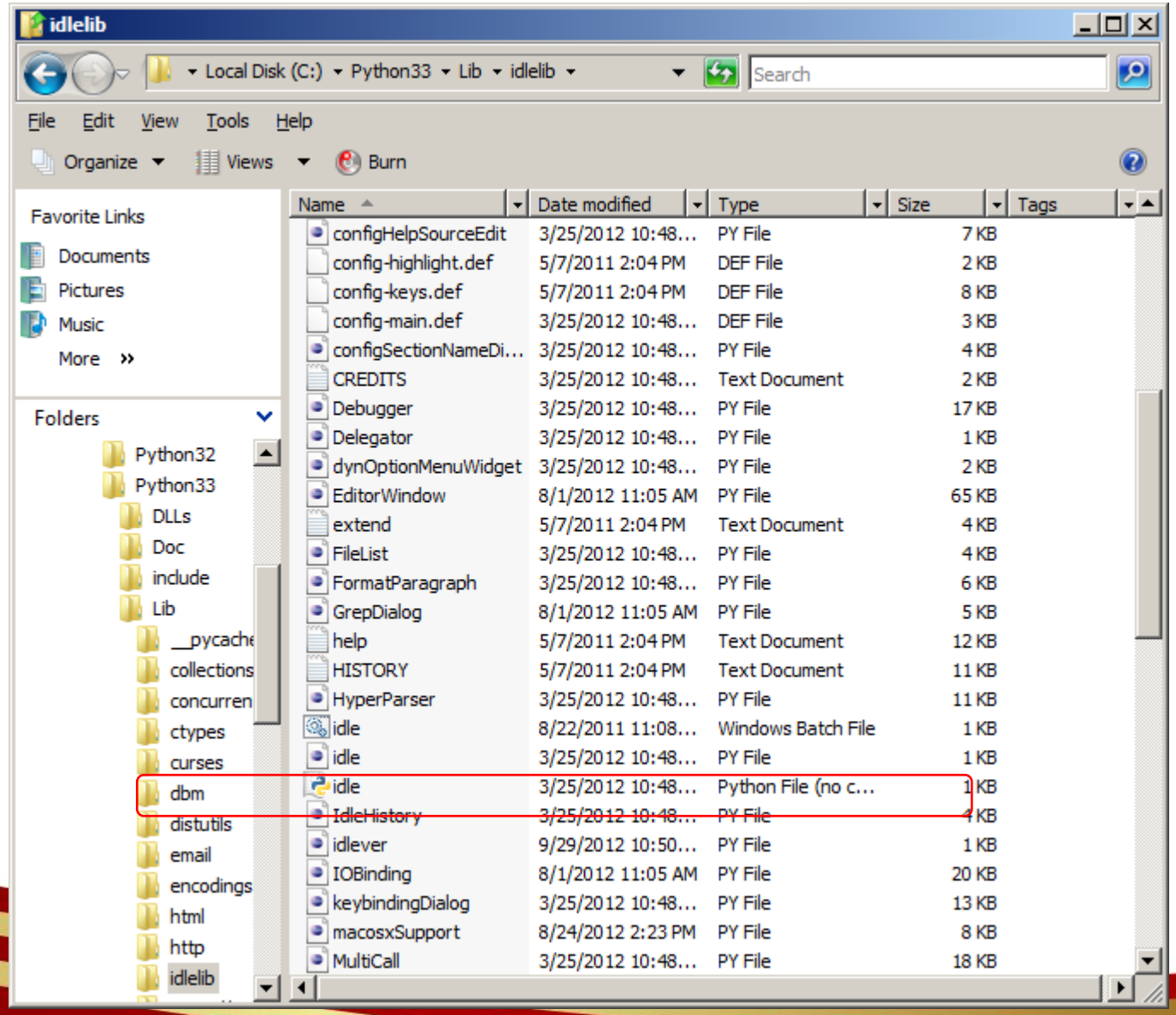
The Python IDLE should be available from your start menu under Python 3.3, simply click on it to run.



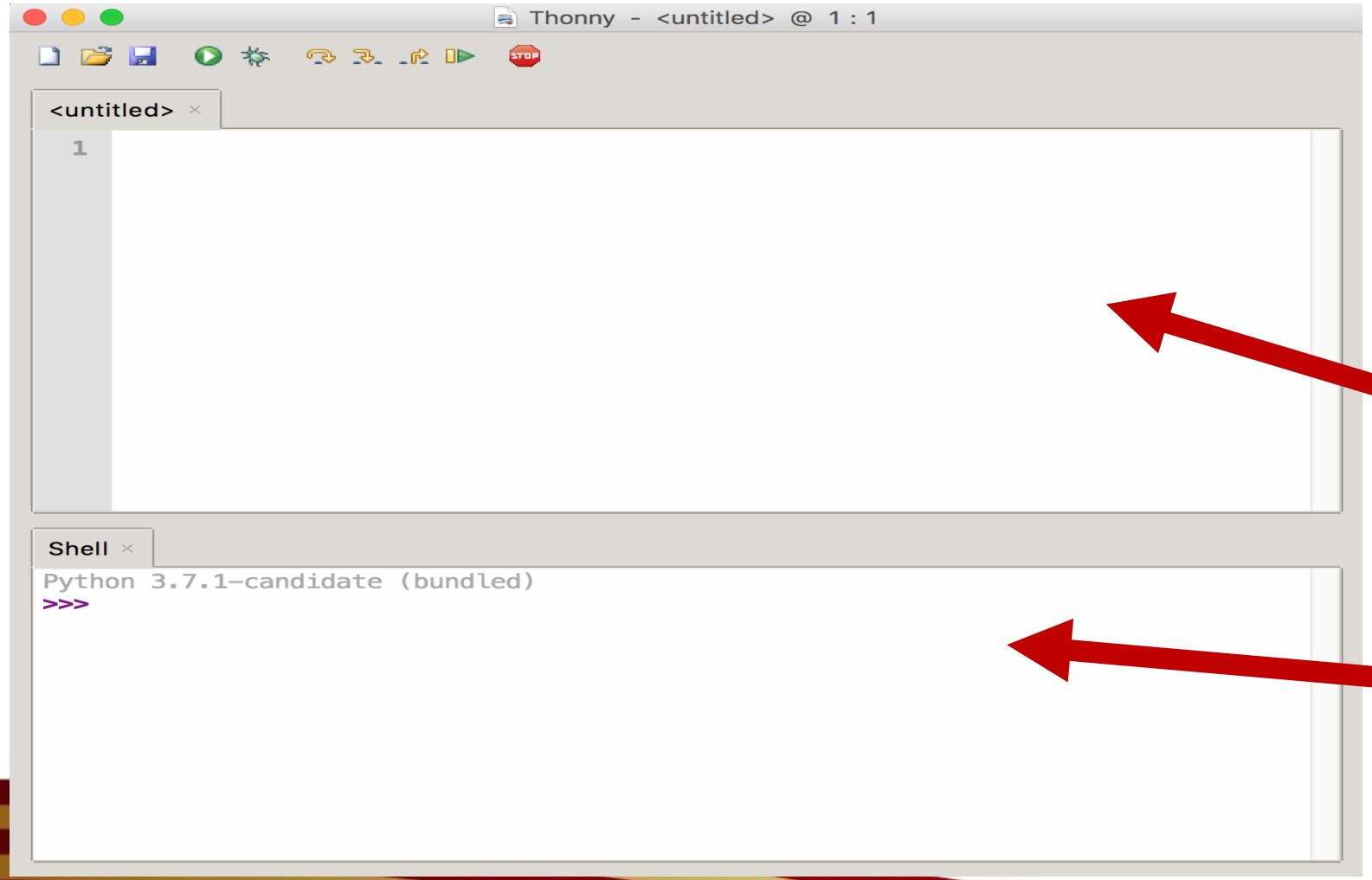
Running Python

The Python IDLE can be found in your `PythonXX/Lib/idlelib` folder as shown below.

Double click on the batch file to run the IDE.



Running Python using Thonny IDE



Code Editor Screen

Shell Screen

Thonny IDE's ICONs

A: The paper icon allows you to create a new file.
B: The open folder icon allows you to open a file that already exists on your computer.
C: The floppy disk icon allows you to save your code. Press this early and often. Y
D: The play icon allows you to run your code. Remember that the code you write is meant to be executed.
E: The bug icon allows you to debug your code.
F-H: The arrow icons allow you to run your programs step by step. This can be very useful when you're debugging or, in other words, trying to find those nasty bugs in your code. The **F** arrow tells Python to take a big step, meaning jumping to the next line or block of code.
The **G** arrow tells Python to take a small step, meaning diving deep into each component of an expression.
The **H** arrow tells Python to exit out of the debugger.
I: The resume icon allows you to return to play mode from debug mode.
J: The stop icon allows you to stop running your code.