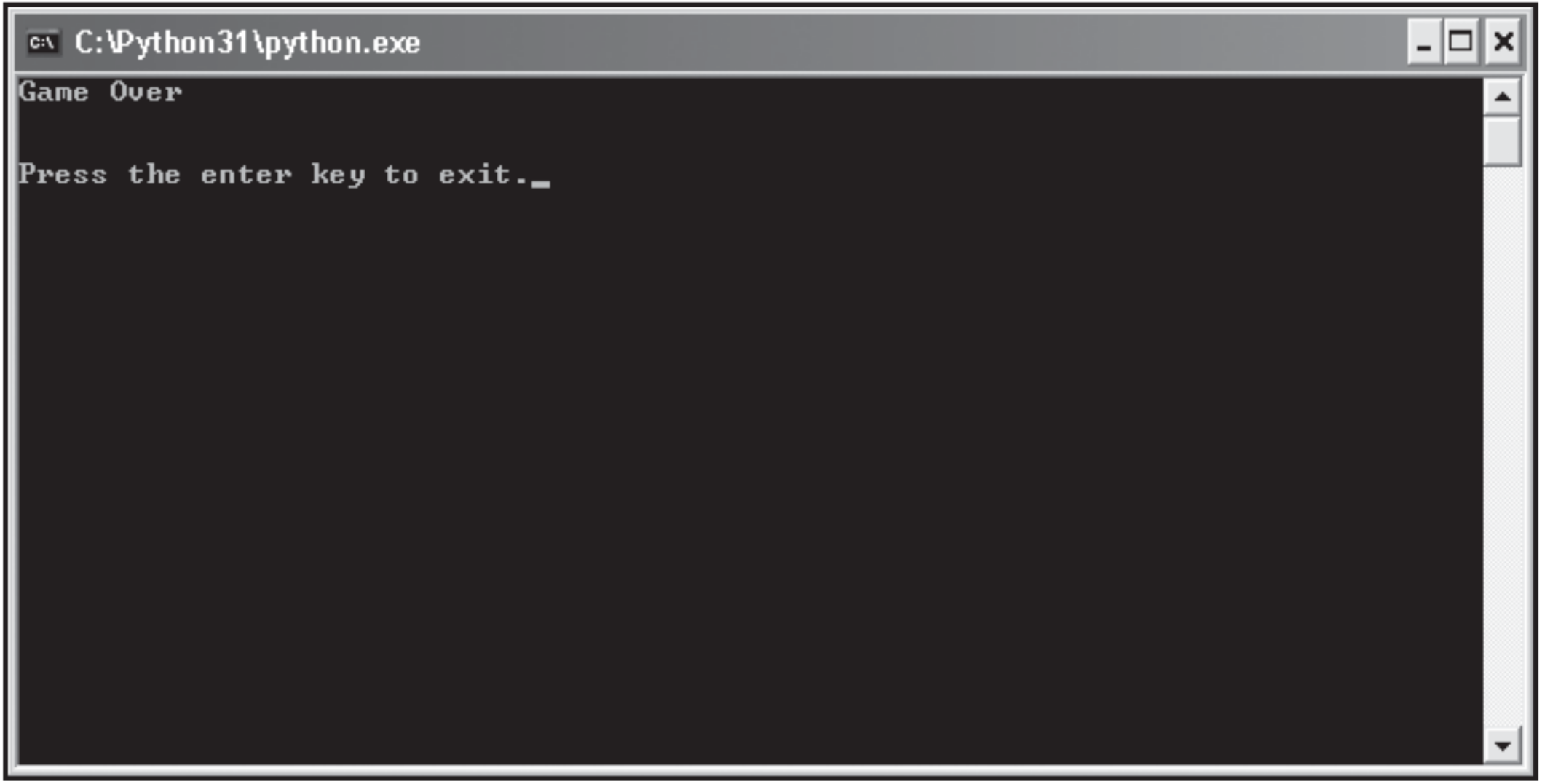
Chapter 1

Getting Started: The Game Over Program

Examining the Game Over Program



- **console window**: a window that can display only text. Though not as nice as windows in a Graphical User Interface (GUI), console applications are easier to write and a good place for the beginning programmer to start.

- The Game Over program is really just a variation of the traditional Hello World program, which displays the words “Hello World” on the screen.

- The Hello World program is often the first program a beginner writes in order to dip his toe in a new language.

Introducing Python

- Python is a powerful easy-to-use programming language developed by Guido van Rossum, first released in 1991.

- With Python, you can quickly write a small project.

- But Python also scales up nicely and can be used for mission-critical, commercial applications.

Python Is Easy to Use

- The major goal of any programming language is to bridge the gap between the programmer's brain and the computer.

- Most of the popular languages you've probably heard of, like Visual Basic, C#, and Java, are considered *high-level languages*, which means that they're closer to human language than machine language.

- Python, with its clear and simple rules, is even closer to English.

- Creating Python programs is so straightforward that it's been called "programming at the speed of thought."

- Python programs are shorter and take less time to create than programs in many other popular languages.

Python Is Powerful

- Python has all the power you'd expect from a modern programming language.

- Python is powerful enough to attract developers from around the world as well as companies.

- Python is also used as a tool by professional game programmers.

Python Is Object-Oriented

- *Object-oriented programming* (OOP) is a modern approach to solving problems with computers.

- It embodies an intuitive method of representing information and actions in a program.

- In Python, using OOP techniques is optional. You have all of OOP’s power at your disposal, but you can use it when you need it.

Python Is a “Glue” Language

- Python can be integrated with other languages such as C, C++, and Java.

- Thus a programmer can take advantage of work already done in another language while using Python.

- One can leverage the strengths of other languages, such as the extra speed that C/C++ might offer, while still enjoying the ease of development in Python.

Python Runs Everywhere

- Python runs on everything from a Palm to a supercomputer, from Windows, Macintosh, to Linux machines.

- Python programs are *platform independent*, which means that regardless of the operating system you use to create your program, it’ll run on any other computer with Python.

Python Has a Strong Community

- Python has something called the Python Tutor mailing list, a more informal way for beginning programmers to ask those first questions.

- The list is at http://mail.python.org/mailman/listinfo/tutor

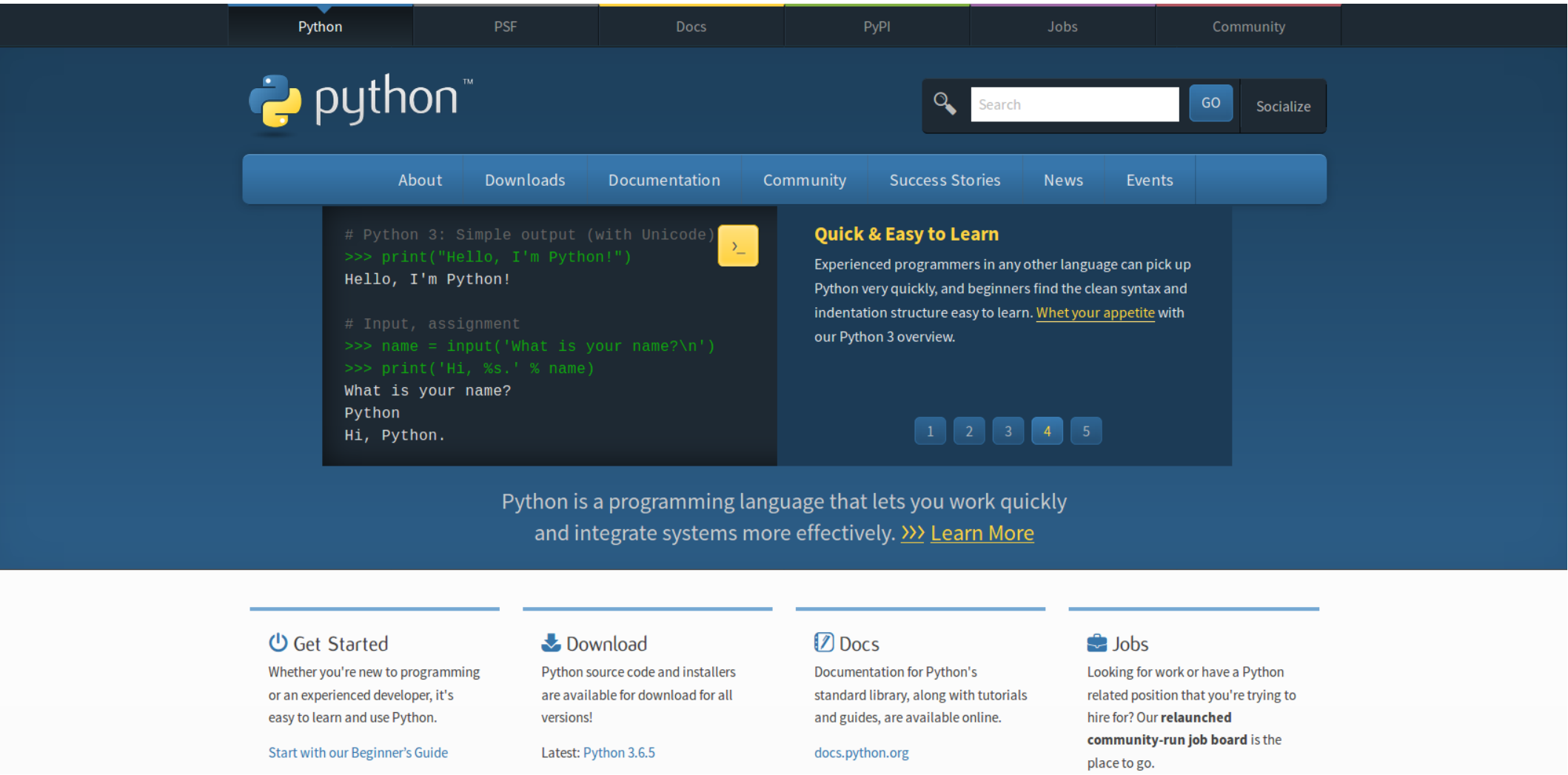
- There are many other Python communities focused on different areas. It is because Python itself is so approachable for beginners.

Python Is Free and Open Source

- Python is free.

- Embracing open-source ideals like this is part of what makes Python popular and successful.

Installing Python



- Visit the official Python website, http://www.python.org, to download the latest version of the language for your machine, and install it.

Introducing IDLE

- Python comes with an integrated development environment called IDLE.

- A *development environment* is a set of tools that makes writing programs easier.

- IDLE provides two modes in which to work: an **interactive** mode and a **script** mode.

Another Installation

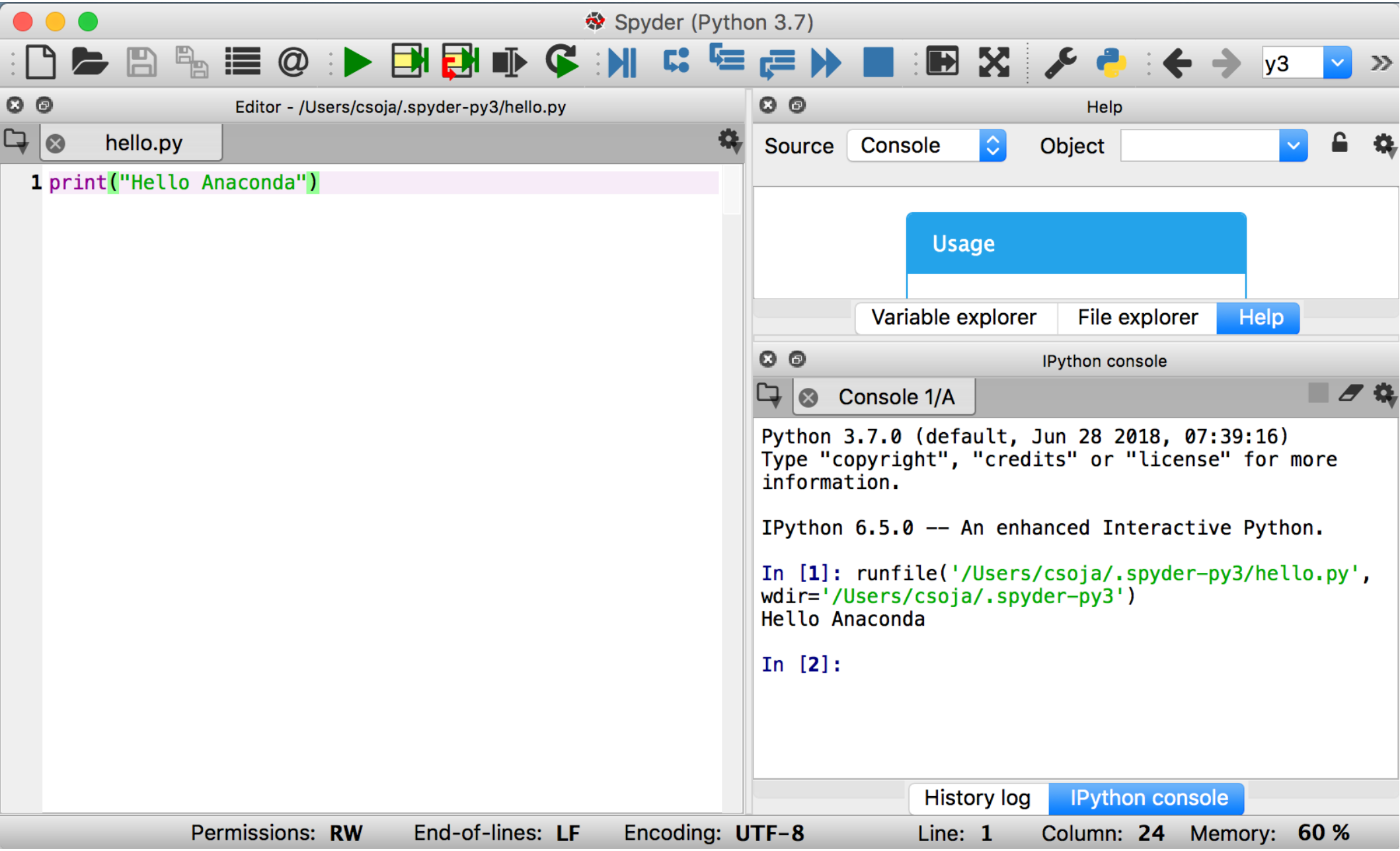
- Install anaconda from www.anaconda.com

- Build a virtual environment in a terminal with

$ conda create -n pyenv
$ conda activate pyenv

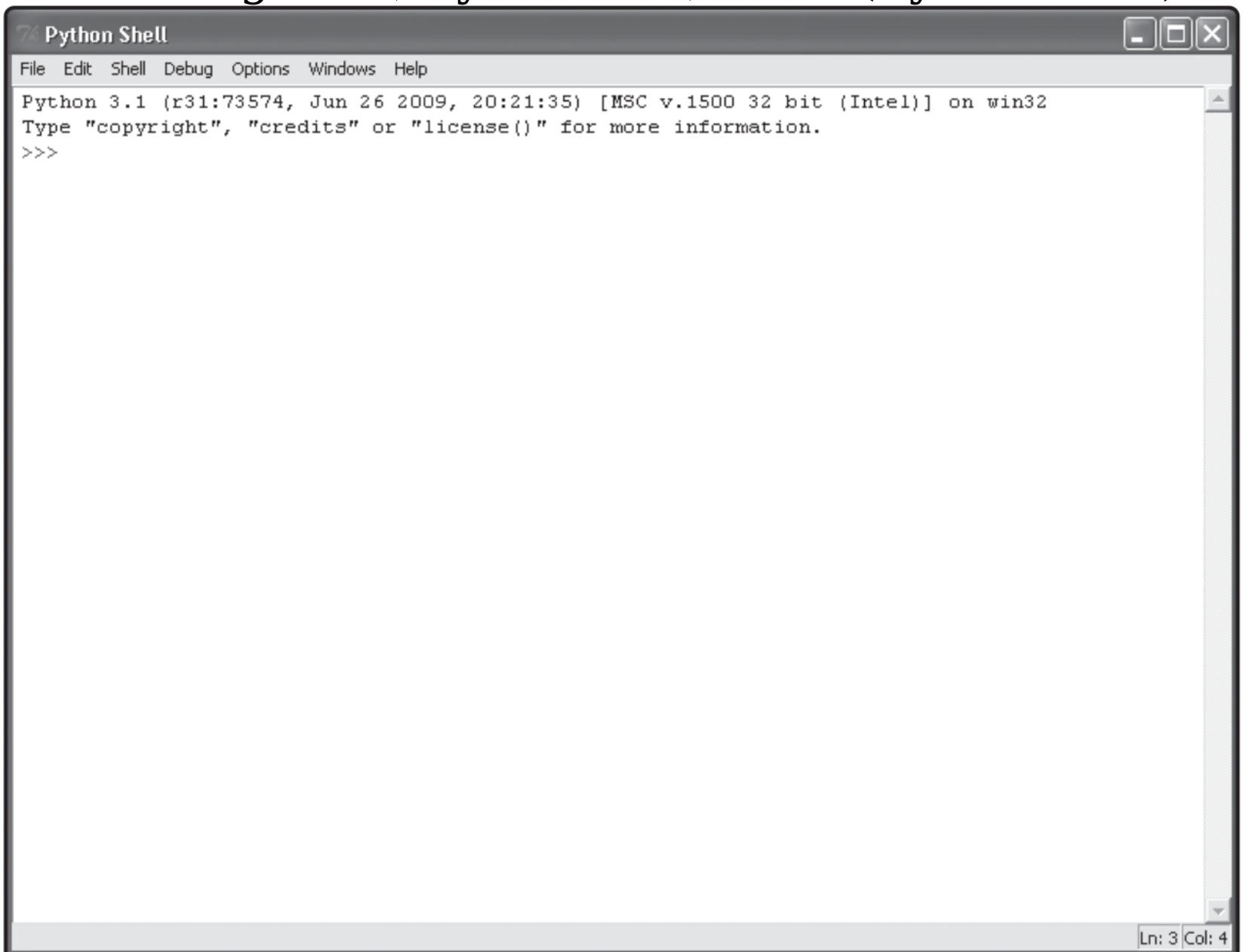
- Install spyder:

$ conda install spyder


Spyder (Python 3.7)
y3
Editor - /Users/csoja/.spyder-py3/hello.py
hello.py
1 print("Hello Anaconda")
Help
Source
Console
Object
Usage
Variable explorer
File explorer
Help
IPython console
Console 1/A
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more
information.
IPython 6.5.0 -- An enhanced Interactive Python.
In [1]: runfile('/Users/csoja/.spyder-py3/hello.py',
wdir='/Users/csoja/.spyder-py3')
Hello Anaconda
In [2]:
History log
IPython console
Permissions: RW
End-of-lines: LF
Encoding: UTF-8
Line: 1
Column: 24
Memory: 60 %

Writing Your First Program

- To begin your interactive session, from the Start menu, choose All Programs, Python 3.1, IDLE (Python GUI).



- This window is called the Python Shell.
- At the command prompt (>>>), type

print("Game Over")

and then press the Enter key.

- The interpreter responds by the following on the screen:

Game Over

- The **print()** function can display text, surrounded by quotes, that you put inside the pair of parentheses. If you put nothing inside the parentheses, it will print a blank line.
- Python is case-sensitive and by convention, function names are in lowercase.

Learning the Jargon

- A *function* is like a mini-program that goes off and performs some specific task.

- The task of the print() function is to display a value.

- You *call* a function by using the function name, followed by a set of parentheses.

- Sometimes you can *pass* a function values to work with. You put these values, called *arguments*, between the parentheses.

- In the 1st program, you passed the print() function the argument "Game Over".

- In this case, you can say that the value "Game Over" you passed the print() function is a *string*, ie, a series of characters.

- The line you entered in the interpreter is also considered a *statement*, ie, a complete instruction.

- *Code* means programming statements.

Generating an Error

- Computers take everything literally.

- At the interactive prompt, if you type primt("Game Over") , the interpreter will respond with something like:

```
Traceback (most recent call last):
File "<pyshell#0>", line 1, in <module>
primt("Game Over")
NameError: name 'primt' is not defined
```

Programming in Script Mode

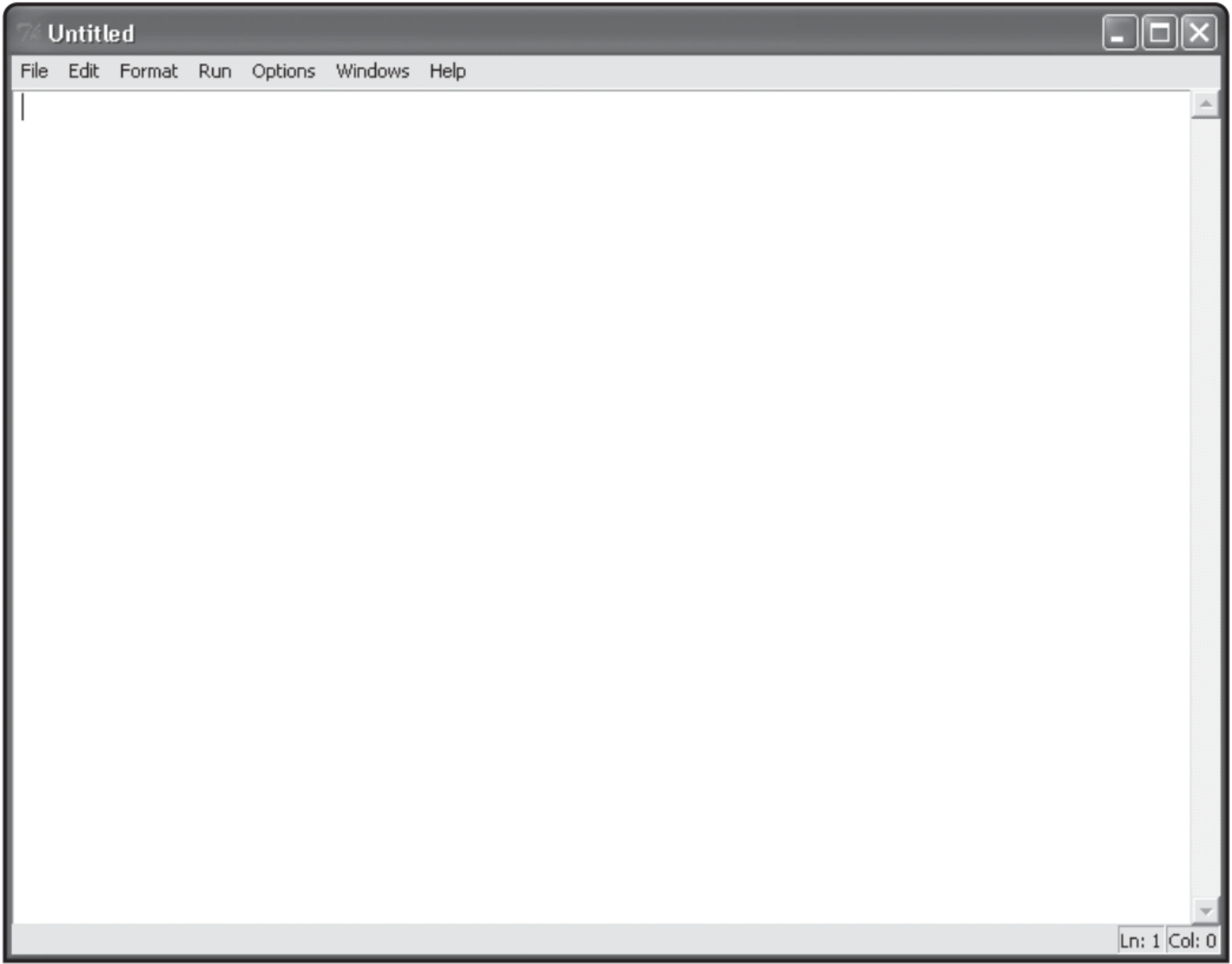
- Using the interactive mode gives you immediate feedback. But it's not designed to create programs you can save and run later.

- Python's IDLE also offers a script mode, in which you can write, edit, load, and save your programs.

- You can open a script mode window from the interactive window you've been using. Select the File menu, then New Window. A new window will appear.

- In this new script window, type print("Game Over") and press Enter. Nothing happens! That's because you're in script mode.

- What you're doing is writing a list of statements for the computer to execute later. Once you save your program, you can run it.


Untitled
File Edit Format Run Options Windows Help
Ln: 1 Col: 0

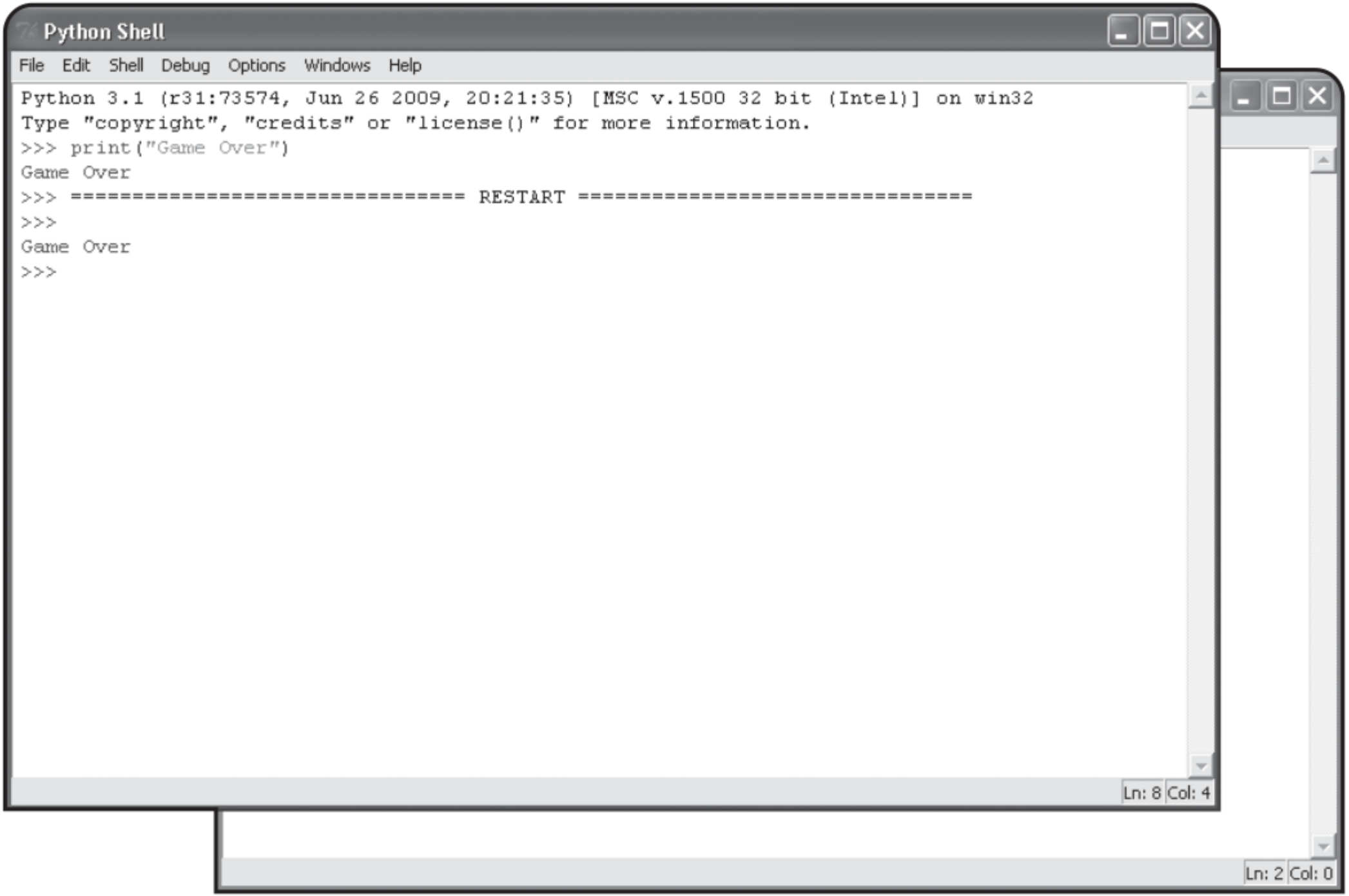
Saving and Running Your Program

- To save your program, select File, Save As, for example, game_over.py .

- Make sure to save your programs with the **.py** extension, to be recognized as a Python program.

- To run my Game Over program, select Run, Run Module. Then, the interactive window displays the results of the program.

- Interactive mode is great for trying out a small idea quickly. Script mode is perfect for writing programs you can run later. Using both modes together is a great way to code.


Python Shell
File Edit Shell Debug Options Windows Help
Python 3.1 (r31:73574, Jun 26 2009, 20:21:35) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Game Over")
Game Over
>>> ========== RESTART ==========
>>>
Game Over
>>>
Ln: 8 Col: 4
Ln: 2 Col: 0

game_over.py

```
# Game Over
# Demonstrates the print function

print("Game Over")

input("\n\nPress the enter key to exit.")
```

Using Comments

- The following are the first two lines of the program:

```
# Game Over
# Demonstrates the print function
```

- These *comments* aren't statements for the computer to execute. In fact, the computer totally ignores them. They are for the humans.

- Comments explain programming code in human language. Comments are invaluable to other programmers and help them to understand your code.

- Comments are also helpful to you. They remind you of how you accomplished something that may not be clear at first glance.

- Create a comment with the number sign symbol #. Anything after # on the rest of the line is a comment.

- It’s good to start a program with a few comments. It’s helpful to list the title of the program and its purpose.

- If you want to modify an old program, a few well-placed comments may make your life much easier.

- Comments are even more helpful to another programmer who needs to modify a program you wrote. This kind of situation comes up a lot in the world of professional programming.

Using Blank Lines

- The computer generally ignores blank lines; they are just for the humans reading the code.

- Blank lines can make programs easier to read.

Waiting for the User

- The last line of the program

```
input("\n\nPress the enter key to exit.")
```

displays the prompt, Press the enter key to exit , and waits for the user to press the Enter key.

- Once the user presses the key, the program ends. This is a nice trick to keep a console window open until the user is done with an application.