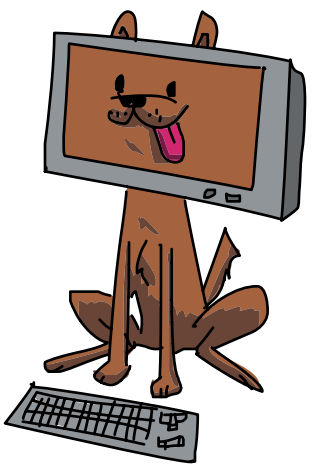
GETTING STARTED WITH PYTHON

To begin coding, we have to speak the computer's language. Computers need step-by-step instructions, and they can only understand certain languages. Just like a person from Russia might not be able to understand English, computers only understand languages made for them.

Computer code is written in programming languages like Python, C++, Ruby, or JavaScript. These languages allow us to "talk" to our computer and give it commands. Think about when you teach a dog to do tricks—when you give the "sit" command, he sits; when you say "speak," he barks. The dog understands those simple commands, but not much else you say.



3. TEST PYTHON WITH A PROGRAM

In your Start menu or *Applications* folder, find the IDLE program and run it. You'll see a text-based command window like the one shown in Figure 1-4. This is called the Python shell. A *shell* is a window or screen that lets the user enter commands or lines of code.

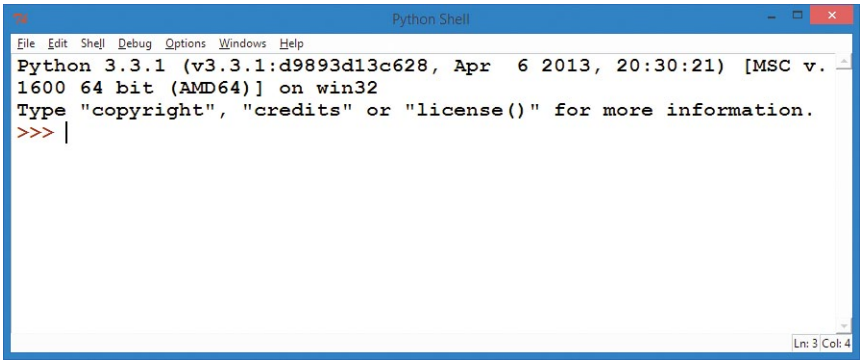


Figure 1-4: The IDLE Python shell—our command center for learning Python

The >>> is called a *prompt*, and it means that the computer is ready to accept your first command. The computer is asking you to tell it what to do. Type

```
print("Hello, world!")
```

and press ENTER or RETURN on your keyboard. You should see the Python shell respond by printing the text in quotes that you entered inside the parentheses: `Hello, world!`. That's it—you've written your first program!

WRITING PROGRAMS IN PYTHON

You'll usually want to write programs that are longer than a single line, so Python comes with an *editor* for writing longer programs. In IDLE, go to the **File** menu and select **File ▸New Window** or **File ▸New File**. A blank screen will pop up, with *Untitled* at the top.

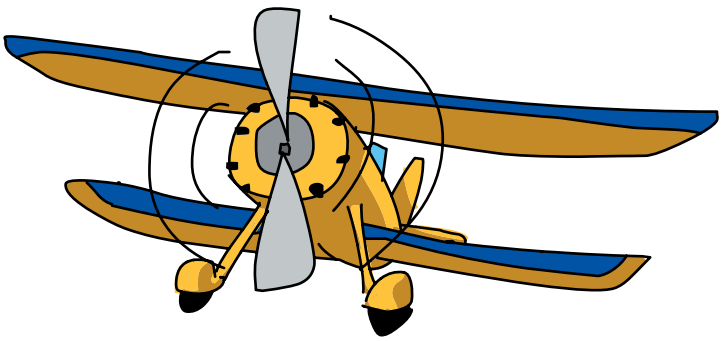
Let's write a slightly longer program in Python. In the new, blank window, type the following three lines of code:

```
# YourName.py
name = input("What is your name?\n")
print("Hi, ", name)
```

The first line is called a *comment*. Comments, which begin with a hash mark (#), are programming notes or reminders that the computer ignores. In this example, the comment is just a note to remind us of the program's name. The second line asks the user to input their name and remembers it as `name`. The third line prints `"Hi, "` followed by the user's name. Notice that there's a comma (,) separating the quoted text `"Hi, "` from the name.

RUNNING PROGRAMS IN PYTHON

Go to the **Run** option on the menu above your program and select **Run ▸Run Module**. This will *run*, or carry out, the instructions in your program. It will first ask you to save the program. Let's call our file *YourName.py*. This tells your computer to save the program as a file called *YourName.py*, and the *.py* part means this is a Python program.



When you save the file and run it, you'll see your Python shell window start the program by showing the question `What is your name?`. Type your name on the next line and press ENTER. The program will print `Hi,` followed by the name you typed. Since this is all that you asked your program to do, the program will end, and you'll see the >>> prompt again, as shown in Figure 1-5.

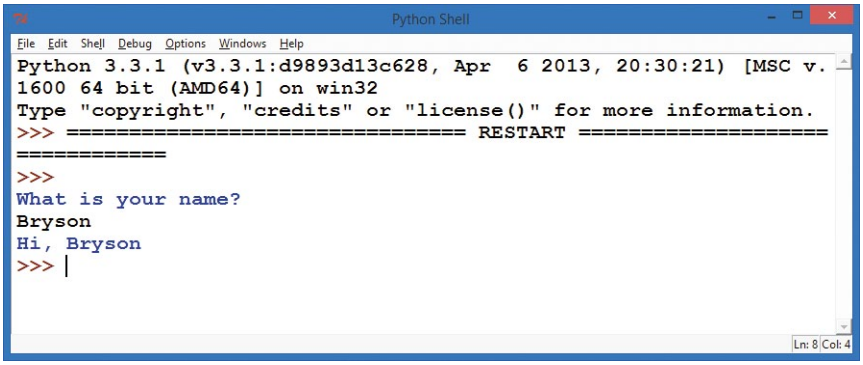


Figure 1-5: The computer knows my name!

For younger learners, like my three-year-old son, it's fun to explain that the program is asking them to type their name. Max knows the letters in his name, so he types *m-a-x* on the keyboard, and he loves it when I tell him the program said `Hi, max` back to him. Ask your young learner if she'd like the program to say something different. Max said "Hello," so I edited the earlier program on the third line to say `Hello,` instead of `Hi,`.

Then I changed the third line to read:

```
print("Hello, ", name, name, name, name, name)
```

Max loved it when the program replied to him with `Hello, max max max max max`. Try experimenting with the second and third lines of the program to have the computer ask different questions and print different answers.

WHAT YOU LEARNED

Learning to code is like learning to solve puzzles, riddles, or brainteasers. You start with a problem, apply what you know, and learn new things along the way. By the time you finish, you've

exercised your mind, and you've answered a question. Hopefully, you've also had fun.

In this chapter, we solved our first major problem: we installed the Python programming language on our computers so that we could start coding. It was as easy as downloading a file, installing it, and running it.

In the chapters that follow, you'll learn how to solve problems using code. You'll start with simple visual puzzles, like drawing shapes on the computer screen (or a tablet or phone), and then find out how to create simple games like Guess a Number, Rock-Paper-Scissors, and Pong.

From the foundation you'll build in these first programs, you can go on to code games, mobile apps, web apps, and more.

At this point, you should . . .

- Have a fully functional Python programming environment and text editor.
- Be able to enter programming commands directly into the Python shell.
- Be able to write, save, run, and modify short programs in IDLE.