Introduction\_to\_Python\_for\_Kids\_c03

**CHAPTER 3 Your First Python Program**

In this chapter, let us look at creating and running programs in Python, and create our very first Python program.

**Creating and running programs in Python**

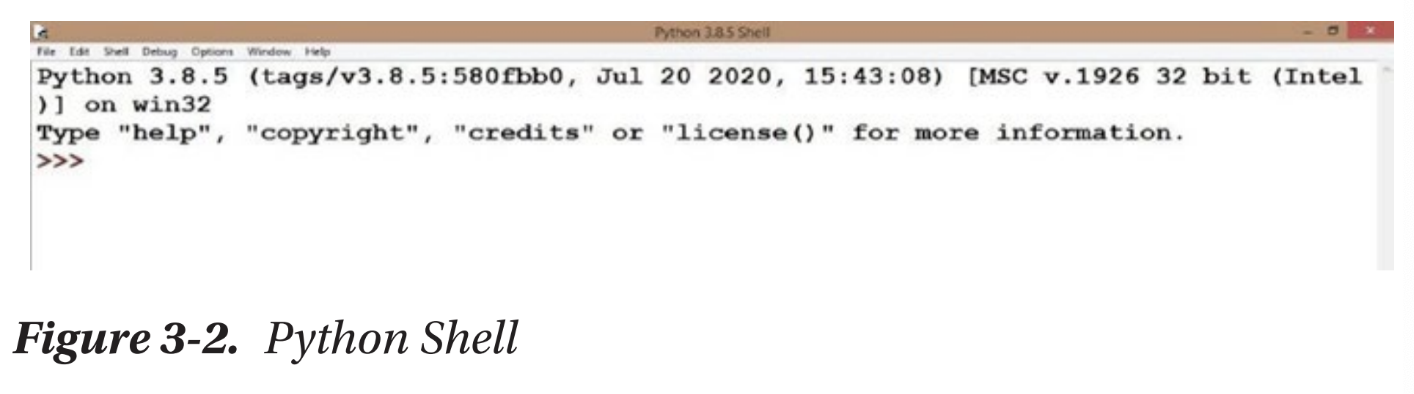
Okay, now that we have installed Python, let us start creating our very first programs. We cannot just write Python programs in MS word or notepad. That is not how it works. We need a specific application that can understand the Python code you write. This application will process your code and give you the desired result.

One of the default Python applications is called IDLE. It is Integrated Development and Learning Environment, and it was developed by the Python Software Foundation. It automatically gets installed when you install Python (Figure 3-1):

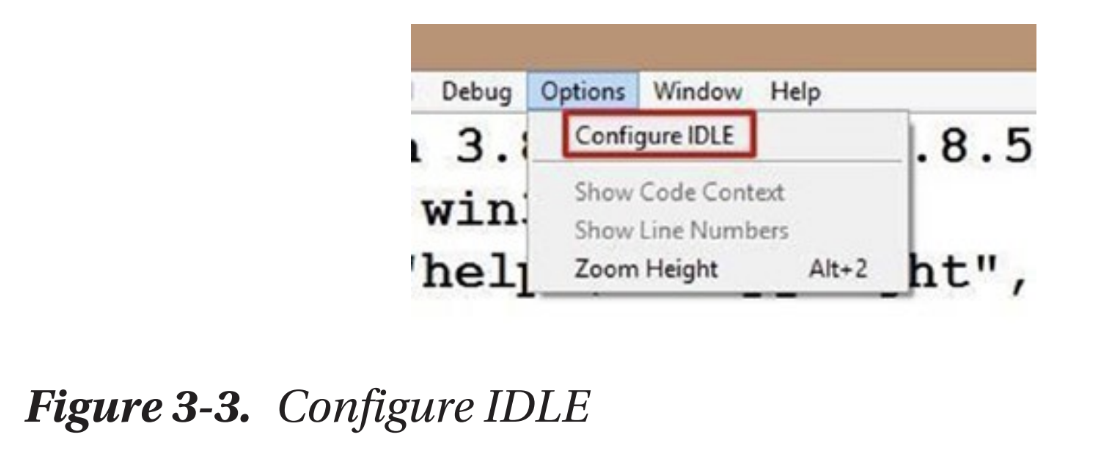
1. Go to your applications (in Windows or Mac) and type IDLE (Figure 3-1).



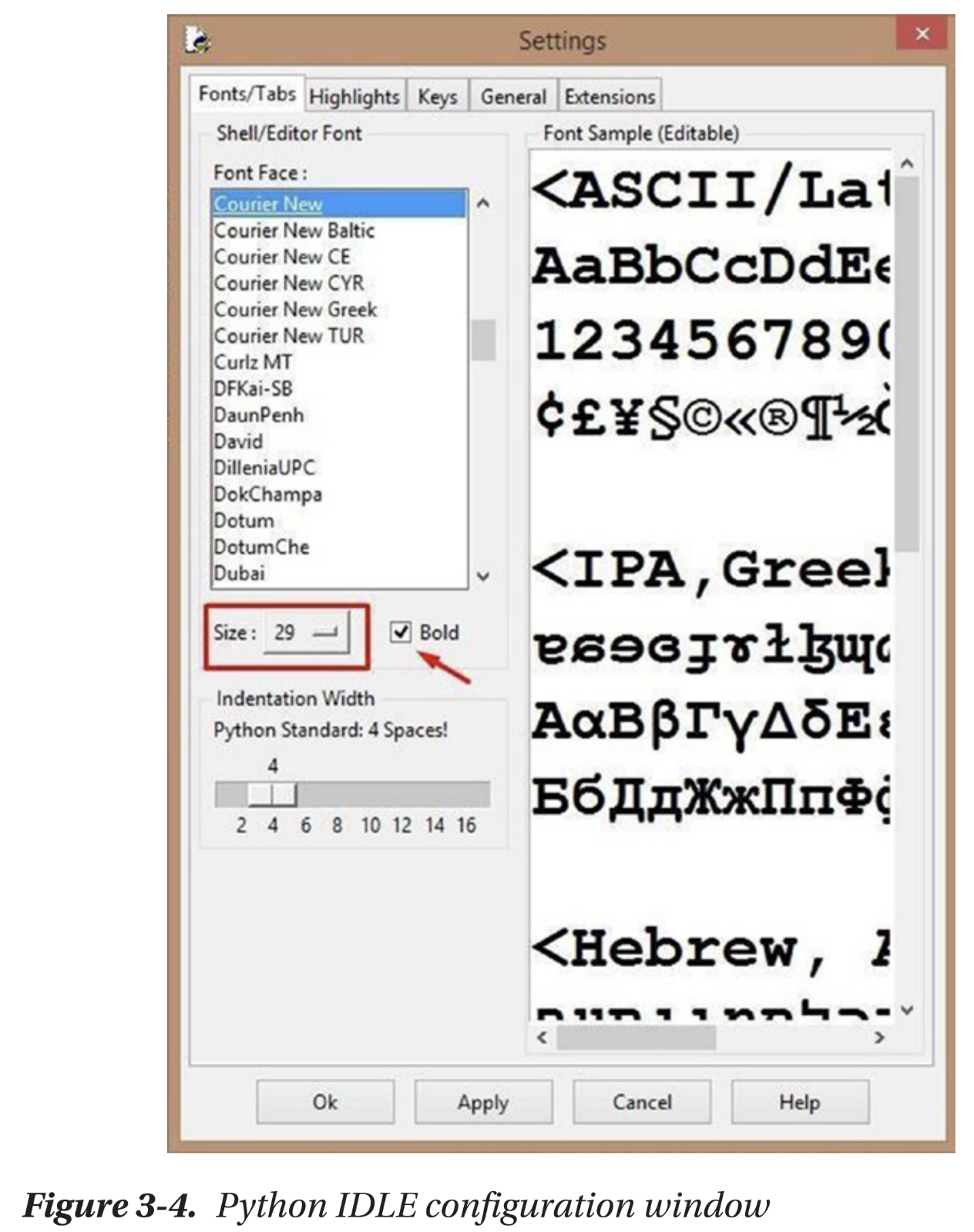
2. When you open the application, the Python Shell will open. This is where we will type our Python programs and get our outputs (results) (Figure 3-2).



3. You can change the way the text looks in this application. You can increase the font, make the text bold, and change the font style. In order to do that, go to Options and click Configure IDLE (Figure 3-3).



4. When you click it, the following window will pop up (Figure 3-4).

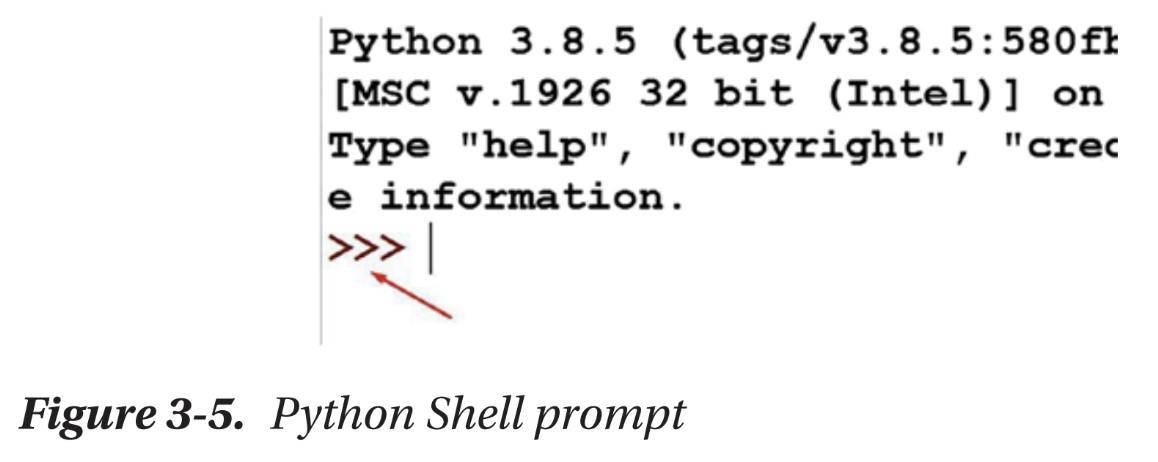


5. Let us change the font size to 29 (look at the highlighted square) and select the checkbox across “Bold” if it is not already checked.

That is all we are going to change now, but as you can see, you have a lot more options. Play around with them to format your Python Shell’s text in any way you want.

**Python interactive mode (Python Shell)**

There are two ways to run Python programs using IDLE. The default method is by directly typing your code into the Python Shell (Figure 3-5).



Do you see the >>> the arrow is pointing to? That is called the Python Shell prompt. It is asking you to type your Python code after the prompt so it can run it and give you the result you are expecting.

Every time you type Python code in the Shell, press Enter; it will run that line of code and execute it. It is quite handy because you get immediate results.

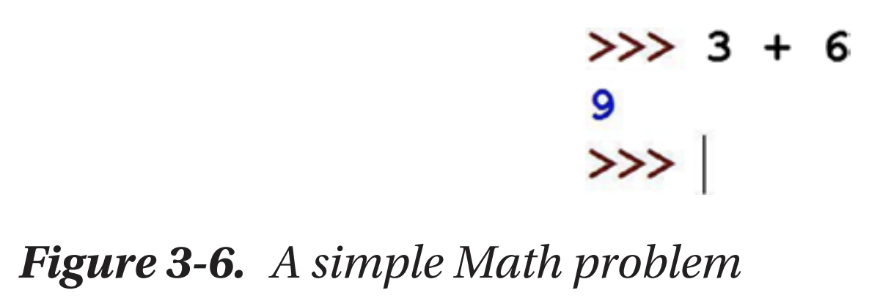
**Your Shell can do Math**

That is right. You can do Math in your Python Shell. Let us try with some basic operations, shall we? I want to prove to you that Python is not an alien language you are learning for the first time. You can do extremely complicated mathematical calculations and get results for those in the Shell as well. Want to see how that works?

Alright, let us start simple. Type the following in the prompt:

3 + 6

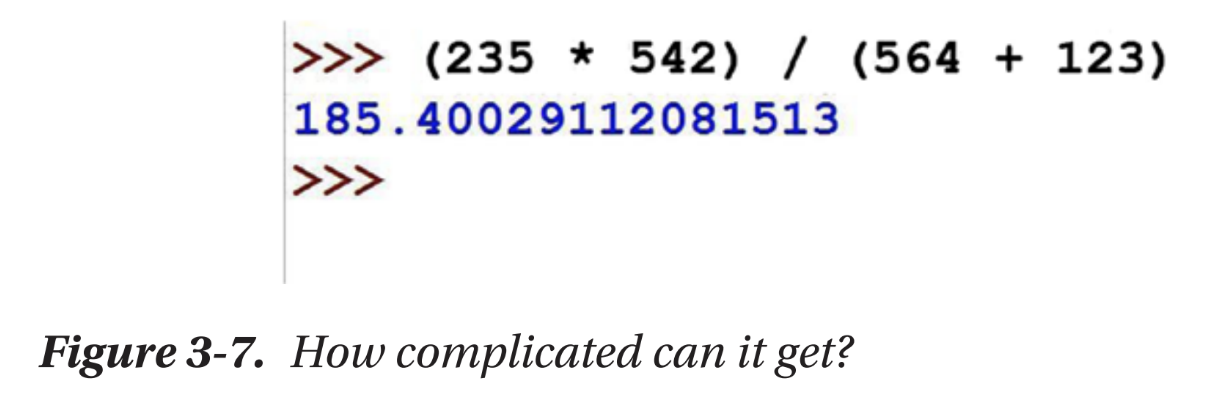
Press Enter and you should see the following result (Figure 3-6).



Did your Python application just do Math? How cool is that? Let us try something more complicated.

(235 \* 542) / (564 + 123)

Run the preceding mathematical expression and you should see the following result (Figure 3-7).



You could cross-verify the result with your calculator. It is correct. You can make the equation as complicated as you want, and your Shell will spit out the result in less than a second. Try a couple more and see!

But is that all you can do? Math problems? Not even close! You can even print things on your screen, and that is what we are going to look at next. But then again, it does not stop there either. You can do a host of things with Python. I do not want to overwhelm you too much from the get-go though, so we are going to take it slow, alright?

**Print with Python**

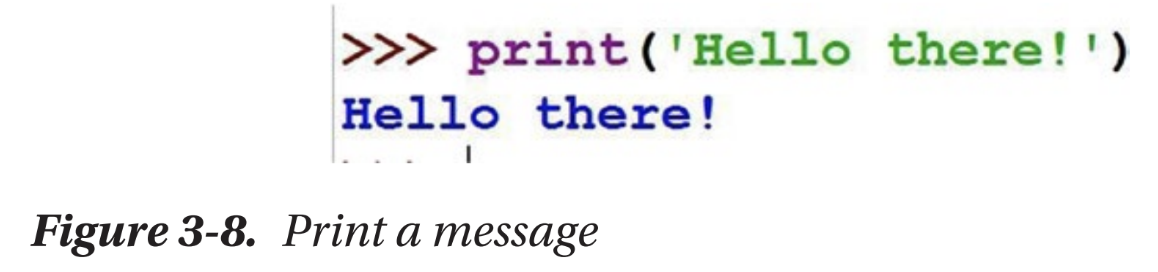
Python is a very easy-to-learn language. Proof? If you want to print something to the screen, just use the “print” command. A pre-defined code/command in Python or any programming language is called a syntax.

So, the syntax to print a message to the screen is as follows:

print('Hello there!')

You need to start and close parenthesis right after “print” and type your message within quotes. It could either be a single (‘message’) or a double quoted (“message”).

When you run the preceding line of code, this is what you will get (Figure 3-8).



But be careful here. The “p” in “print” should be a lowercase p. If you use an uppercase p, you will get an error message like in Figure 3-9.

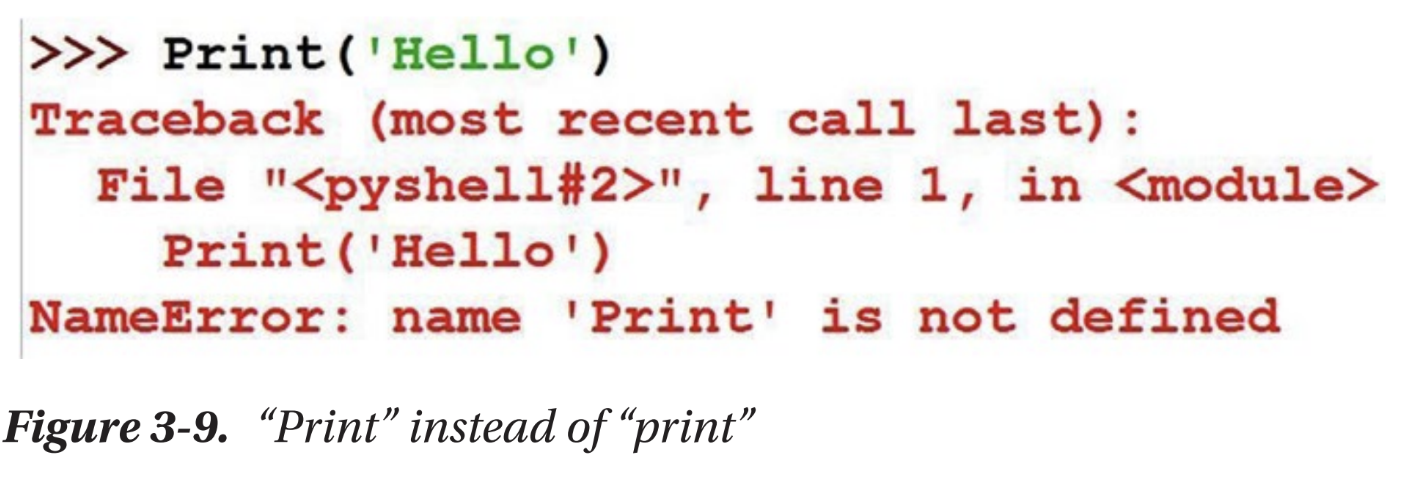


Figure 3-8. Print a message Figure 3-9. “Print” instead of “print

The error message says ‘Print’ is not defined. That is because as far as Python is concerned, “print” is different from “Print” and the command to print something to the screen uses a lowercase p. In other words, Python is case sensitive. So be sure to use the “commands” or “syntax” as it is.

**IDLE script mode**

Remember how I said that there are two ways to write programs with your IDLE? We have looked at the first method so far. It looks easy at the first glance, but did you notice a problem with it?

While using the Shell, you get outputs for every single line of code, and that will work as long as you write very simple lines of code. But once you start writing actual programs, you would want an application that processes multiple lines of code together and gives you the final result. You need the script mode to make that happen.

Let us look at how that works. Let us print the same ‘Hello there!’ statement, but now in script mode.

Go to File ➤ New File (Figure 3-10).

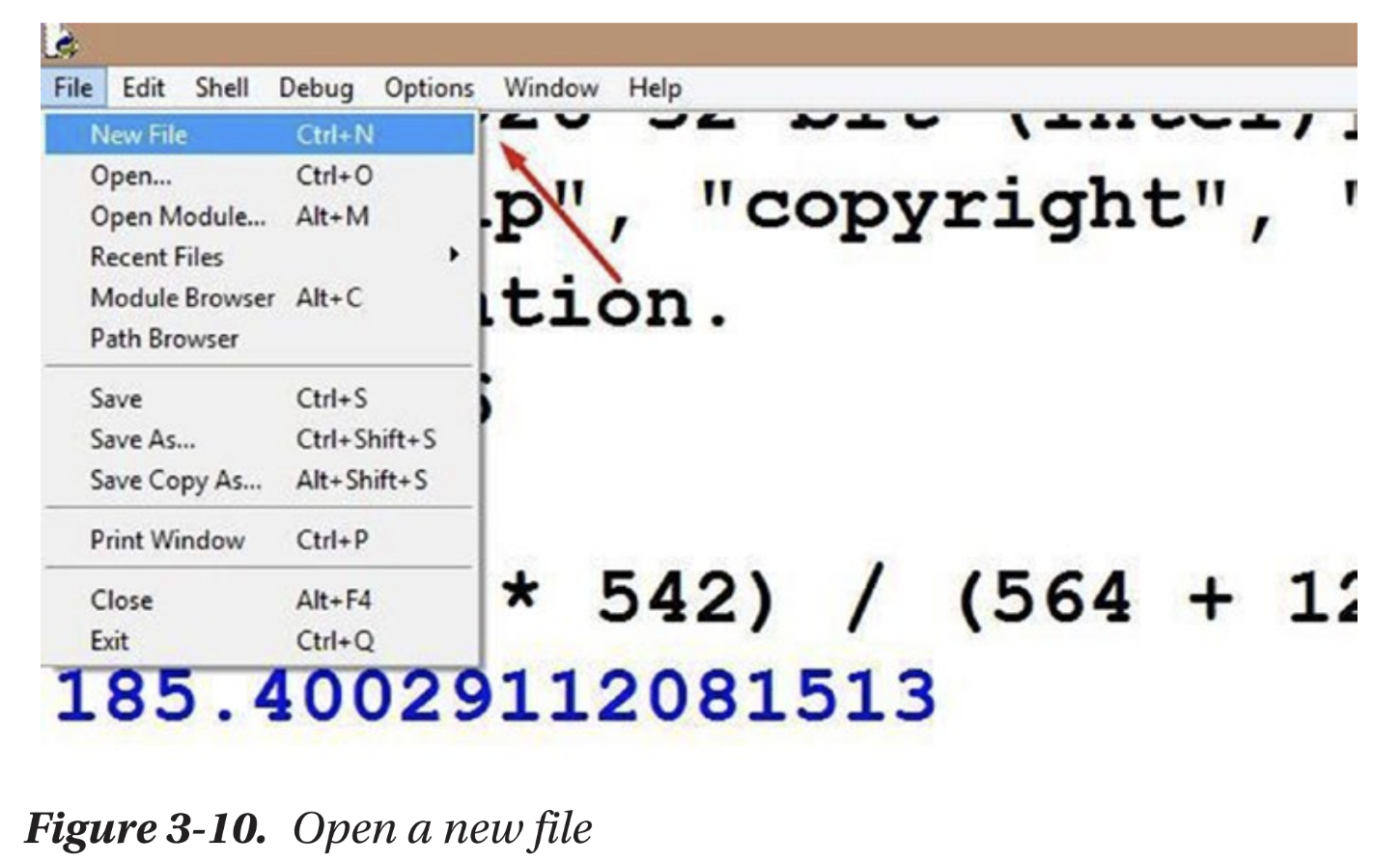
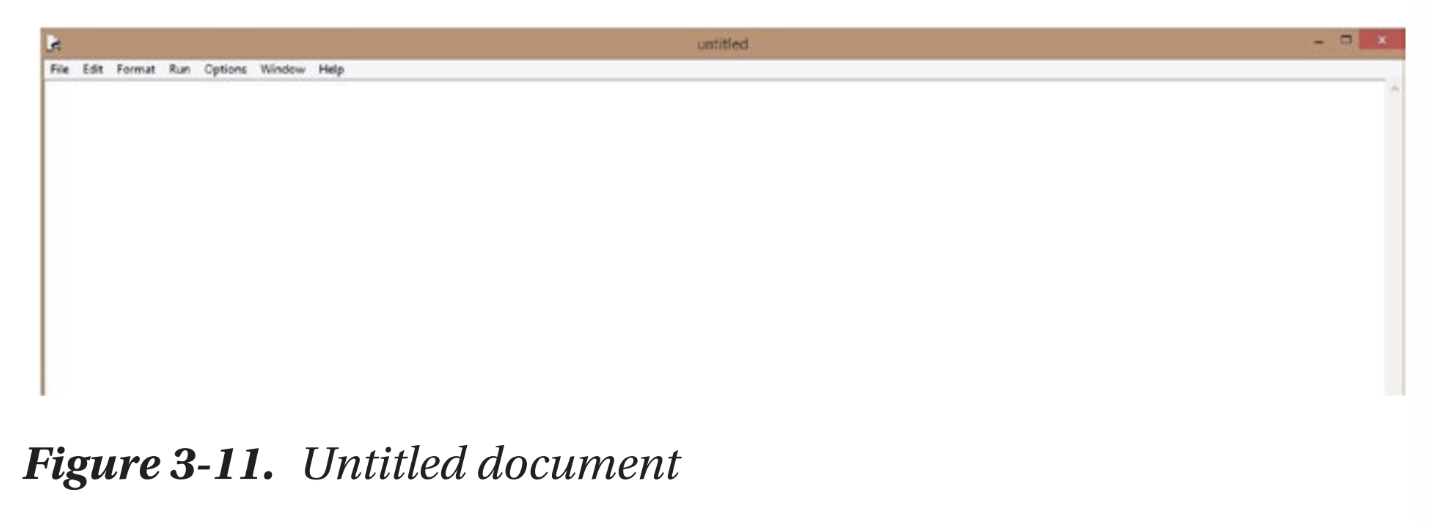
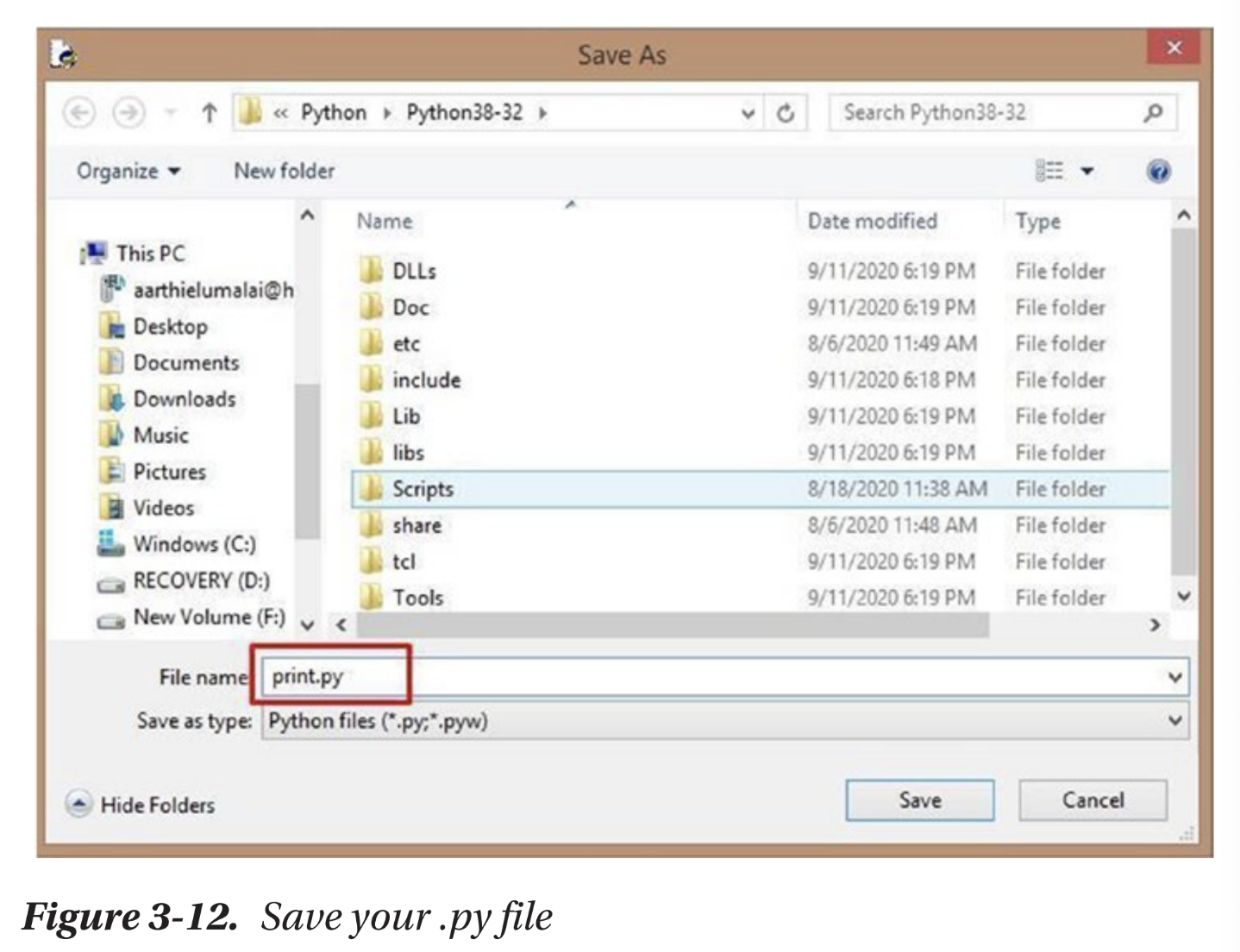


Figure 3-10. Open a new file

An untitled document will open like the following one (Figure 3-11).



Go to File ➤ Save As (Figure 3-12) and save the document with the .py extension. .py denotes that a particular file has Python code in it and needs to be executed as such.



We named our file print.py. Click Save, and you should see the name of your file change from untitled to print.py. Now, you can name your file anything you want, but just make sure that the Save as type is “Python files” or you give an extension .py or both, alright? Now, let us type our line of code again (Figure 3-13):

print('Hello there!')



There you go! You have written your very first Python program. Whoohoo! �

Let us run it, shall we?

Go to Run ➤ Run Module (Figure 3-14).

It will ask you to save the code again. Click OK. Your IDLE should open the Shell window again, and in the very last prompt, you should see the result (“Hello there!”) printed, like in Figure 3-15.

Now you have run your first Python program and gotten your very first result! Whoohoo again! � Python activity: Print your name (and some more) We have come to our very first Python activity now. You are going to print your name. Actually, why don’t we make it a small introduction? You are going to introduce yourself and print that introduction on the screen. Do not worry. It is not hard to do. I will teach you how. Let us create an imaginary character and name her Susan Smith. Let us assume that she is 9 years old and she loves puppies. Now let us introduce her to the world! Open a new file and save it as introduction.py. You know how to do that, don’t you? Now, follow along with me. Type the following into the file: print('Hello there!') print('My name is Susan Smith.') print('I am 9 years old.') print('I love puppies! :)') We need to print out multiple lines on our screen, so we have created multiple print statements. Let us save the code we just created and run it. Go to Run ➤ Run Module.

Our introduction is on the screen (Figure 3-16)! Now, I want you to do the same with your introduction. What is your name? How old are you? What do you like? Print everything on the screen. Go ahead. It is quite easy to do. Congrats, you are now a budding Python developer. � Summary In this chapter, we looked at IDLE and its interactive programming environment. We tried a few Math problems in the Python Shell and then created our very first print statement. Then we learned about the script mode and writing and running more than one line of code at a time. We finished the chapter with an activity where we introduced ourselves and printed the introduction to the screen over multiple lines of code. In the next chapter, we will look at using numbers with your Python code, manipulating them, and a lot more.