

1.3 Using Python

The Python system that you have installed on your computer has several tools for developing and running Python programs. One of these is called "python" on Windows and "Python Launcher" on a Mac. This tool allows the system to run programs written in the Python language. This opens a window for the program's input and output, runs the program, and then closes this window. Think for a moment, however, about what this means. Suppose you have a program that opens up a dictionary and looks for words that contain four consecutive vowels, such as "queue". This program needs no input from the user. It opens the dictionary file, opens a window to print its output, prints that output and stops. At the end of the program it closes any files or windows it has opened. This program can run in a small fraction of a second. It does give some output, but that output goes into a window that closes before you have a chance to read it. The entire process is useless – the program has done its job, but its output has not been available for the human user.

In Chapter 8 of these notes we will look at programs that create a graphical user interface, with buttons, menus and similar "widgets". As we will see, these programs need to be run through the Python Launcher. The programs and the windows they create will persist until the user clicks a Quit button, so we will have time to examine any output from the program. For all of the work we will do prior to Chapter 8 we will use a different tool for both creating and running the program. This is called IDLE (after Eric Idle, one of the founding members of the Monty Python comedy group). IDLE can be found the way you find other programs on your computer – from the Start window's All Programs link on Windows or from the Applications folder of the Macintosh Finder. On either system look for the Python (or Python3.4) folder, and IDLE will be one of the programs inside it.

IDLE has two primary windows: an editor window and a command interpreter, or shell, window. (A command interpreter is called a "shell" because it is the interface between the internals of a program and the outside world.) The editor lets you create programs. The command interpreter will run the program that is in the editor window, and it also has a prompt for you to enter Python command that will be executed as you enter them. For example, if you enter

```
> 3+4*5
```

at the prompt the system will reply

```
23
```

Each of these windows has an Options menu with a Configure IDLE item that allows you to specify which window should be on top when you start IDLE. From the editor window there is a Shell item in the Run menu that brings up a shell. Alternatively, if you select the Run Module item in the Run menu the program in the edit window will be executed. If you are in a shell window and want to create a new program, from the File menu select New Window and a new editor window will be opened, into which you can type your program.

Note that you must save the program to a file (Save As from the File menu) before you can run it. Be sure you know which window you are saving – it is not uncommon for beginners to save the shell window where the program was executed. You want to save the IDLE editor window that has the code for the program.

Suppose we type the following program into the editor and save it as file Heads.py. Note that by convention Python programs are always given names that end in ".py". This suffix is the what the system uses to recognize the file as containing Python code.

```
from random import *

def Toss():
    x = randint(0, 1)
    if x == 0:
        return "Heads"
    else:
        return "Tails"

def CountHeads(N):
    count = 0
    for x in range(N):
        result = Toss()
        if result == "Heads":
            count = count + 1
    return count

def main():
    num = CountHeads(1000)
    print( "%d out of %d" %(num, 1000) )

main()
```

Program 1.3.1: Counting Heads in a series of tosses

This needs to be typed in exactly as it looks here; if there are typos the program won't run. We will see later how to find and eliminate errors; for now we'll just assume there aren't any. You will see that the editor colors some words differently than others; we will discuss the significance of this later. After we save this program we can run it by selecting Run Module from the Run menu of the edit window. This will open a shell in which you will see the output of the program, a line such as

483 out of 1000

This program simulates 1000 random tosses of a coin, so the output will change each time we run it. We can run the program again either by going back to the edit window (which should still be open) and again selecting Run Module from the Run menu, or in the shell window we could type

```
> main()
```

You might not fully understand why this works until you have read Chapter 5 on functions, but we will set up all of our programs so at the top level there is a function `main()`, so you can always re-run a program by typing that into the shell after the program has run once.

Most of the times you use Python you will not be typing a program into the system from scratch. It is much more common to start with an existing program and modify it into the program you want. Sometimes you will just want to examine a program to see how it works. To bring up a program in the IDLE editor you need to click on it in a special way. On Windows systems you click with the right mouse button on a ".py" file. This brings up a *context menu* with the options for this file. One of these is **Edit with IDLE** and that opens the IDLE editor on the program. Macintoshes generally don't have 3-button mice. On Mac you <CTRL>-click (hold down the Control or <CTRL>-key while clicking the mouse button) on the file. This time there are three choices: **IDLE**, **Build Applet** and **Python Launcher**. One of these will be labeled **Default**, which means that it is the choice that will apply if you just click normally on the file. You want to choose **IDLE** here.

To change the default action on a Mac, select any Python (.py) file and go to the **File** menu of the Finder. This has an option **GetInfo**, which brings up a window with various items related to this file. One of them is **Open With**, which has the **IDLE**, **Build Applet** and **Python Launcher** options. You can either change the default for this file, or if you select **Change All** you can change the default for all .py files.