Chapter 1
Getting Up and Running

WELCOME TO *Learning Python with Raspberry Pi*. In this book, you'll learn how to unlock the full power of the tiny computer, from 3D graphics to games programming to controlling electronics to tweeting. You'll see what's going on under the hood and learn how to create programs that take advantage of every feature of this minuscule computer.

Setting Up Your Raspberry Pi

To follow this book, you'll need a few bits of equipment:

- Raspberry Pi
- USB keyboard
- USB mouse
- SD card
- Monitor
- Power supply

There are also a few optional bits of kit that may help:

- Powered USB hub (highly recommended)
- Camera module
- USB webcam
- USB WiFi dongle

It is possible to do everything in this book with a model A Raspberry Pi. The real advantage of a model B as far as programming is concerned is the network port. This port will make it easier to connect to the Internet, which you'll need to do to install some software.

Any USB keyboard and mouse should work fine. Most SD cards should work, although there are a few that will cause problems. If you're unsure, buy one from a Raspberry Pi online shop (there are links to a few on `http://raspberrypi.org`).

The Raspberry Pi has a HDMI (high-definition multimedia interface) video output, but most monitors have VGA or DVI input. If at all possible, use a monitor that has DVI or HDMI input. A HDMI-to-DVI converter should cost only a few pounds/dollars and shouldn't detract from the image quality. HDMI-to-VGA converters are available, but they're more expensive and can cause problems, so use them only if you have no other option.

Most micro USB power supplies from reputable manufacturers should work; however, some cheap ones from no-name companies have caused problems, so if possible, don't skimp too much on this. You could use a USB cable from a normal computer to power your Pi.

Powered USB hubs are recommended for the power-related problems described later in this chapter. Not all USB hubs are powered, so make sure that whatever one you get plugs into the mains electricity to get extra power.

We talk more about camera options in Chapter 9 on multimedia. The only thing to say here is that if you do choose to get a USB webcam, make sure it's compatible with the Raspberry Pi. There's a partial list of working web cams at `http://elinux.org/RPi_USB_Webcams`.

You'll need to connect your Pi to the Internet to install the software you need in this book. You can do this either by plugging your Pi into your router with a network cable or by using a USB wireless dongle, which will add WiFi connectivity.

Solving Problems

The most common problems with the Raspberry Pi are power-related issues. Not all micro USB power sources can provide enough power, and it becomes more of a problem as you connect peripherals to your Pi, or when you overclock it (see Chapter 5 for more details). Power-related problems will usually manifest themselves as the computer crashing, so if you find that your Pi becomes unstable, this is the best place to start. A good way to get around such issues is to connect your Pi to one power source and connect all the peripherals (keyboard, mouse, and so on) via a powered USB hub.

The second most common cause of problems with Pis is the SD card. These issues can be caused by power supply problems, or they can be problems with the cards themselves. It's important to take preventative measures here to ensure that your data is safe, and that means backups! You can use a service such as Google Drive (although this runs slowly on the Pi), or you can simply keep extra copies of any work on a USB memory stick. SD card issues will usually manifest themselves by the Pi displaying error messages when you try to start it. Most of the time you can solve the problem by reinstalling Raspbian, but if this doesn't work, you'll need to get a new SD card.

If neither of these help, then you'll need to dig a little deeper. The most useful places to look are the kernel buffer and the system log file. The kernel buffer is usually best if you're having problems with hardware, such as a USB device not working. If you open LXTerminal and type:

```
dmesg
```

It will output all the messages from the Linux Kernel. The last ones are the most recent and should show any problems.

The system log file (often called syslog) can be displayed with:

```
cat /var/log/syslog
```

Again, the most recent messages will be at the end. The information in both of these can be somewhat cryptic. If you still can't work out the problem after reading these, the best place to go is the Raspberry Pi forums at `www.raspberrypi.org/phpBB3/`. There's a community of helpful people who should be able to point you in the right direction.

A Quick Tour of Raspbian

This is a book about programming, not about generally using Raspbian, so we won't dwell on it too much, but you'll find it useful to know a bit about what's going on.

There are a few operating systems available for the Raspberry Pi, but the instructions in this book are all based on Raspbian, which is the default operating system, and the best choice for a new user. If you have some experience with Linux, you could use Arch or Fedora if you like, but you'll have to change the `apt-get` commands to ones suitable for your package manager.

The easiest way to install Raspbian on your Pi is using NOOBS, which is available from `www.raspberrypi.org/downloads`. You'll also find a quick start guide at that website that will tell you everything you need to know to get up and running.

There are two different ways of interacting with Raspbian—from the terminal and using the graphical system (LXDE).

Using LXDE (Lightweight X11 Desktop Environment)

The Lightweight X11 Desktop Environment is the standard windowing system for Raspbian. Its basic setup is the same as most versions of Windows pre-Windows 8. There's a button in the bottom-left side of the screen that opens an applications menu, and currently running applications are displayed in the bar along the bottom (see Figure 1-1).

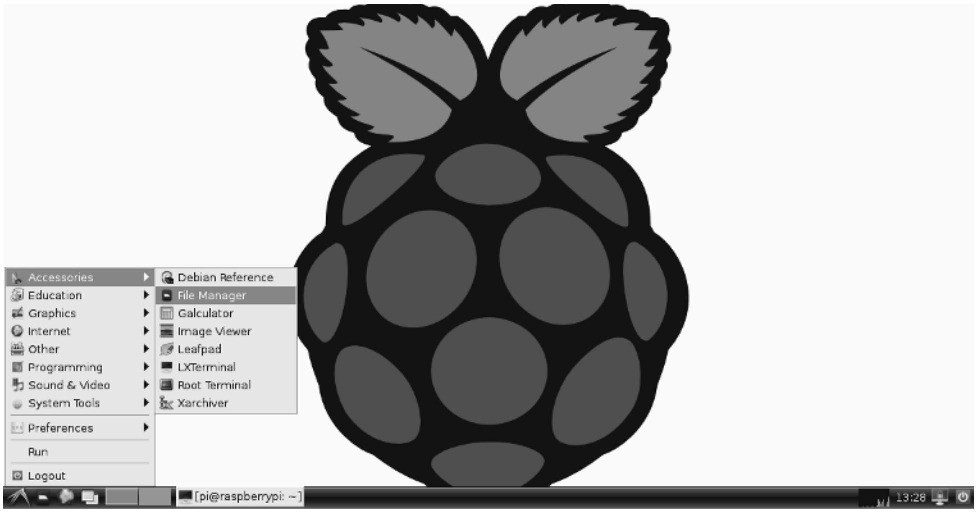

FIGURE 1-1: The LXDE desktop with the menu open.

If you get a black screen with white text asking you to log in when you boot up your Pi, it means that you haven't set it up to start LXDE automatically. Don't worry; just log in with the username `pi` and the password `raspberry`, and then type the following:

```
startx
```

You can set it up to boot into LXDE automatically using `raspi-config` (see the next section).

Using the Terminal

LXDE is great for many tasks, but sometimes you'll need to use the command line. This is an incredibly powerful interface that's accessed through the terminal. In LXDE, that means opening the LXTerminal application.

When you open LXTerminal, you should see the following line:

```
pi@raspberrypi~$
```

This signifies that you are using the username `pi` on a computer called `raspberrypi`, and you are in a directory called ~.

In Linux, all directories start from / or root. This is the base of the directory tree and every directory is located in some subdirectory of this. You can move between directories using the `cd` (change directory) command. Start by moving to this root directory with:

```
cd /
```

You should now seen that the command prompt has changed to

```
pi@raspberrypi/$
```

You can list the contents of this directory with the command `ls`. One of the subdirectories is called `home`. This is where every user on the system has his `home` directory. Move into it and view its contents with:

```
cd home
ls
```

There should only be one directory called `pi`. The command prompt should now have changed to show that you're in the directory `/home`. Move into the only subdirectory with:

```
cd pi
```

Now the command prompt will have reverted to:

```
pi@raspberrypi~$
```

This is because the character ~ is a shorthand for the current user's home directory. When you type ~ in the terminal, the computer converts it to `/home/pi`.

There is much more to learn about the command line. So much so that it would take another book this size to cover it with any semblance of completeness. However, you don't need to know everything to start using it, and whenever we tell you to use LXTerminal, we tell you exactly what to type.

TIP

If you are interested in learning more about the Raspberry Pi, or Linux in general, the command line is an excellent place to start, and there's loads of information about it both online and in print. The Linux command-line book, which you can browse for free online, is an excellent place to start. See `http://linuxcommand.org/tlcl.php`.

We'll leave you with two pieces of advice. Firstly, don't be afraid of the terminal. It can be a bit daunting at first, but the only way to learn how to use it is to use it. Secondly, almost all commands have built-in help that you can access using the flag `--help`. For example, if you want to learn more about how to use the command `ls`, you can enter:

```
ls --help
```

This will output:

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by
default). Sort entries alphabetically if none of -cftuvSUX nor
--sort is specified.
```

It then goes on to list all the various flags you can use with the command.

Changing Configurations with Raspi-Config

Raspbian comes with a tool to help you set up the hardware on your Raspberry Pi; it's called `raspi-config`. To start it, open LXTerminal and type:

```
sudo raspi-config
```

Here, you'll find options to start LXDE automatically when you boot up, overclock your Pi, and other things. Overclocking your Pi will make a few things in this book run a little better, most notably, installing new software.

Installing Software

You can install new software on your Raspberry Pi using the `apt-get` command in the terminal. Before installing anything, it's a good idea to update all your software to the latest version. You can do this with:

```
sudo apt-get update
sudo apt-get upgrade
```

Then you can use `apt-get` to install whatever you want. For example, if you want to use `iceweasel` (a re-branded version of Firefox), you need to open LXTerminal and type:

```
sudo apt-get install iceweasel
```

If you prefer to do this using a graphical program, you can get the program synaptic with:

```
sudo apt-get install synaptic
```

When you want to install something, you can start it with:

```
sudo synaptic
```

From there you'll be able to search for whatever you want.

NOTE

Whenever you install software, you need to use the word `sudo` before the command. It tells the computer that you want to make a system-wide change and gives the program sufficient permissions to do this.

Python 3

In this book, you'll learn how to use the Python 3 programming language. In Raspbian, there are a couple of ways to use this language.

The Python Interpreter

There are two ways of using Python, from the shell and saved programs. The shell executes each instruction as you type it, which means it's a really good way of trying out things and doing experiments. Saved programs are bits of Python code that are saved in a text file and run all at once. It's easy to tell which environment you're in because in the shell, all the lines will start with three chevrons:

```
>>>
```

Most of the time in this book, we'll deal with saved programs, but there are some occasions (particularly early on) when we tell you to use the shell. To make it clear which bits of code are for which, we've started every bit of code for the shell with three chevrons.

Running Python Programs

There are two different ways you can write programs for Python. You can create text files that contain the code, and then run these files with Python, or you can use an Integrated Development Environment (IDE) such as IDLE 3. Either way will result in the code being run in the same way and it's just a matter of personal preference.

If you want to write the programs as text files, you need to use a text editor such as Leafpad. A word processor such as LibreOffice's Writer is unsuitable because the various formatting it uses will confuse Python. As an example, open Leafpad and create a new file that just has the line:

```
print("Hello World!")
```

Once you've created your file, just save it with the extension .py; for example `testfile.py`. You can then run it by opening LXTerminal and navigating to where the file is saved. Then you run `python <filename>`. You can use the `cd` command to move to different directories. For example, if you save the file in a folder called `programming` in your `home` directory, you could run it by typing the following into LXTerminal:

```
cd programming
python3 testfile.py
```

If everything has worked correctly, you should see the following line appear on the screen:

```
Hello World!
```

The second way is a little simpler. Using an IDE, the text editor and Python interpreter are in the same program. For example, open IDLE 3 (make sure to use the one with the 3), and go to File➪New Window. In the new window, enter this code:

```
print("Hello IDLE")
```

Then go to Run➪Run Module. It will prompt you to save the module, so select a filename. Once you've done this, it will switch back to the Python interpreter and display the following:

```
Hello IDLE
```

It doesn't really matter which one you use, so just go with the way you feel most comfortable with.

Summary

After reading this chapter, you should understand the following a bit better:

- You'll need a few extra bits of hardware to get the most out of your Raspberry Pi.
- Insufficient power is the most common cause of problems.
- If you're having problems, `dmesg` and `syslog` are the best places to find out what's going on.
- Raspbian uses the LXDE desktop environment.
- The terminal provides the most powerful way of interacting with the underlying operating system.
- The `raspi-config` tool lets you configure your Raspberry Pi.
- Use `apt-get` to install new software.
- You can run Python either through the interpreter or by running saved programs.