Chapter 18

Ten Ways to Make a Living with Python

In This Chapter

- Using Python for QA
- Creating your own way in a smaller organization
- Employing Python for special product-scripting needs
- Working as an administrator
- Demonstrating programming techniques
- Delving into location data
- Mining data of various sorts
- Working with embedded systems
- Processing scientific data
- Analyzing data in real time

You can literally write any application you want using any language you desire given enough time, patience, and effort. However, some undertakings would be so convoluted and time consuming as to make the effort a study in frustration. In short, most (possibly all) things are possible, but not everything is worth the effort. Using the right tool for the job is always a plus in a world that views time as something in short supply and not to be squandered.

Python excels at certain kinds of tasks, which means that it also lends itself to certain types of programming. The kind of programming you can perform determines the job you get and the way in which you make your living. For example, Python probably isn't a very good choice for writing device drivers, as C/C++ are, so you probably won't find yourself working for a hardware company. Likewise, Python can work with databases, but not at the same depth that comes natively to other languages such as Structured Query Language (SQL), so you won't find yourself working on a huge corporate database project. However, you may find yourself using Python in academic settings because Python does make a great learning language. (See my blog post on the topic at `http://blog.johnmuellerbooks.com/2014/07/14/python-as-a-learning-tool/`.)

The following sections describe some of the occupations that do use Python regularly so that you know what sorts of things you might do with your new-found knowledge. Of course, a single source can't list every kind of job. Consider this an overview of some of the more common uses for Python.

Working in QA

A lot of organizations have separate Quality Assurance (QA) departments that check applications to ensure that they work as advertised. Many different test script languages are on the market, but Python makes an excellent language in this regard because it's so incredibly flexible. In addition, you can use this single language in multiple environments — both on the client and on the server. The broad reach of Python means that you can learn a single language and use it for testing anywhere you need to test something, and in any environment.

REMEMBER

In this scenario, the developer usually knows another language, such as C++, and uses Python to test the applications written in C++. However, the QA person doesn't need to know another language in all cases. In some situations, blind testing may be used to confirm that an application behaves in a practical manner or as a means for checking the functionality of an external service provider. You need to check with the organization you want to work with as to the qualifications required for a job from a language perspective.

Why you need to know multiple programming languages

Most organizations see knowledge of multiple programming languages as a big plus (some see it as a requirement). Of course, when you're an employer, it's nice to get the best deal you can when hiring a new employee. Knowing a broader range of languages means that you can work in more positions and offer greater value to an organization. Rewriting applications in another language is time consuming, error prone, and expensive, so most companies look for people who can support an application in the existing language, rather than rebuild it from scratch.

From your perspective, knowing more languages means that you'll get more interesting jobs and will be less likely to get bored doing the same old thing every day. In addition, knowing multiple languages tends to reduce frustration. Most large applications today rely on components written in a number of computer languages. In order to understand the application and how it functions better, you need to know every language used to construct it.

Knowing multiple languages also makes it possible to learn new languages faster. After a while, you start to see patterns in how computer languages are put together, so you spend less time with the basics and can move right on to advanced topics. The faster you can learn new technologies, the greater your opportunities to work in exciting areas of computer science. In short, knowing more languages opens a lot of doors.

Becoming the IT Staff for a Smaller Organization

A smaller organization may have only one or two IT staff, which means that you have to perform a broad range of tasks quickly and efficiently. With Python, you can write utilities and in-house applications quite swiftly. Even though Python might not answer the needs of a large organization because it's interpreted (and potentially open to theft or fiddling by unskilled employees), using it in a smaller organization makes sense because you have greater access control and need to make changes fast. In addition, the ability to use Python in a significant number of environments reduces the need to use anything but Python to meet your needs.

TIP

Some developers are unaware that Python is available in some non-obvious products. For example, even though you can't use Python scripting with Internet Information Server (IIS) right out of the box, you can add Python scripting support to this product using the steps found in the Microsoft Knowledge Base article at `http://support.microsoft.com/kb/276494`. If you aren't sure whether a particular application can use Python for scripting, make sure that you check it out online.

Performing Specialty Scripting for Applications

A number of products can use Python for scripting purposes. For example, Maya (`http://www.autodesk.com/products/autodesk-maya/overview`) relies on Python for scripting purposes. By knowing which high-end products support Python, you can find a job working with that application in any business that uses it. Here are some examples of products that rely on Python for scripting needs:

- 3ds Max
- Abaqus
- Blender
- Cinema 4D
- GIMP
- Google App Engine
- Houdini
- Inkscape
- Lightwave
- Modo
- MotionBuilder
- Nuke
- Paint Shop Pro
- Scribus
- Softimage

This is just the tip of the iceberg. You can also use Python with the GNU debugger to create more understandable output of complex structures, such as those found in C++ containers. Some video games also rely on Python as a scripting language. In short, you could build a career around creating application scripts using Python as the programming language.

Administering a Network

More than a few administrators use Python to perform tasks such as monitoring network health or creating utilities that automate tasks. Administrators are often short of time, so anything they can do to automate tasks is a plus. In fact, some network management software, such as Trigger (`http://trigger.readthedocs.org/en/latest/`), is actually written in Python. A lot of these tools are open source and free to download, so you can try them on your network. Also, some interesting articles discuss using Python for network administration, such as "Intro to Python & Automation for Network Engineers" at `http://packetpushers.net/show-176-intro-to-python-automation-for-network-engineers/`. The point is that knowing how to use Python on your network can ultimately decrease your workload and help you perform your tasks more easily. If you want to see some scripts that are written with network management in mind, check out 25 projects tagged "Network Management" at `http://freecode.com/tags/network-management`.

Teaching Programming Skills

Many teachers are looking for a faster, more consistent method of teaching computer technology. Raspberry Pi (`http://www.raspberrypi.org/`) is a single-board computer that makes obtaining the required equipment a lot less expensive for schools. The smallish device plugs into a television or computer monitor to provide full computing capabilities with an incredibly simple setup. Interestingly enough, Python plays a big role into making the Raspberry Pi into a teaching platform for programming skills (`http://www.piprogramming.org/main/?page_id=372`).

In reality, teachers often use Python to extend native Raspberry Pi capabilities so that it can perform all sorts of interesting tasks (`http://www.raspberrypi.org/tag/python/`). The project entitled, Boris, the Twitter Dino-Bot (`http://www.raspberrypi.org/boris-the-twitter-dino-bot/`), is especially interesting. The point is that if you have a teaching goal in mind, combining Raspberry Pi with Python is a fantastic idea.

Helping People Decide on Location

A Geographic Information System (GIS) provides a means of viewing geographic information with business needs in mind. For example, you could use GIS to determine the best place to put a new business or to determine the optimum routes for shipping goods. However, GIS is used for more than simply deciding on locations — it also provides a means for communicating location information better than maps, reports, and other graphics, and a method of presenting physical locations to others. Also interesting is the fact that many GIS products use Python as their language of choice. In fact, a wealth of Python-specific information related to GIS is currently available, such as

- The GIS and Python Software Laboratory (`http://gispython.org/`)
- Python and GIS Resources (`http://www.gislounge.com/python-and-gis-resources/`)
- GIS Programming and Automation (`https://www.e-education.psu.edu/geog485/node/17`)

Many GIS-specific products, such as ArcGIS (`http://www.esri.com/software/arcgis`), rely on Python to automate tasks. Entire communities develop around these software offerings, such as Python for ArcGIS (`http://resources.arcgis.com/en/communities/python/`). The point is that you can use your new programming skills in areas other than computing to earn an income.

Performing Data Mining

Everyone is collecting data about everyone and everything else. Trying to sift through the mountains of data collected is an impossible task without a lot of fine-tuned automation. The flexible nature of Python, combined with its terse language that makes changes extremely fast, makes it a favorite with people who perform data mining on a daily basis. In fact, you can find an online book on the topic, *A Programmer's Guide to Data Mining,* at `http://guidetodatamining.com/`. Python makes data mining tasks a lot easier. The purpose of data mining is to recognize trends, which means looking for patterns of various sorts. The use of artificial intelligence with Python makes such pattern recognition possible. A paper on the topic, "Data Mining: Discovering and Visualizing Patterns with Python" (`http://refcardz.dzone.com/refcardz/data-mining-discovering-and`), helps you understand how such analysis is possible. You can use Python to create just the right tool to locate a pattern that could net sales missed by your competitor.

Of course, data mining is used for more than generating sales. For example, people use data mining to perform tasks such as locating new planets around stars or other types of analysis that increase our knowledge of the universe. Python figures into this sort of data mining as well. You can likely find books and other resources dedicated to any kind of data mining that you want to perform, with many of them mentioning Python as the language of choice.

Interacting with Embedded Systems

An embedded system exists for nearly every purpose on the planet. For example, if you own a programmable thermostat for your house, you're interacting with an embedded system. Raspberry Pi (mentioned earlier in the chapter) is an example of a more complex embedded system. Many embedded systems rely on Python as their programming language. In fact, a special form of Python, Embedded Python (`https://wiki.python.org/moin/EmbeddedPython`), is sometimes used for these devices. You can even find a YouTube presentation on using Python to build an embedded system at `http://www.youtube.com/watch?v=WZoeqnsY9AY`.

Interestingly enough, you might already be interacting with a Python-driven embedded system. For example, Python is the language of choice for many car security systems (`http://www.pythoncarsecurity.com/`). The remote start feature that you might have relies on Python to get the job done. Your home automation and security system (`http://www.linuxjournal.com/article/8513`) might also rely on Python.

Python is so popular for embedded systems because it doesn't require compilation. An embedded-system vendor can create an update for any embedded system and simply upload the Python file. The interpreter automatically uses this file without having to upload any new executables or jump through any of the types of hoops that other languages can require.

Carrying Out Scientific Tasks

Python seems to devote more time to scientific and numerical processing tasks than many of the computer languages out there. The number of Python's scientific and numeric processing modules is staggering (`https://wiki.python.org/moin/NumericAndScientific`). Scientists love Python because it's small, easy to learn, and yet quite precise in its treatment of data. It's possible to produce results using just a few lines code. Yes, you could produce the same result using another language, but the other language might not include the prebuilt modules to perform the task, and it would most definitely require more lines of code even if it did.

The two sciences that have dedicated Python modules are space sciences and life sciences. For example, there is actually a module for performing tasks related to solar physics. You can also find a module for working in genomic biology. If you're in a scientific field, the chances are good that your Python knowledge will significantly impact your ability to produce results quickly while your colleagues are still trying to figure out how to analyze the data.

Performing Real-Time Analysis of Data

Making decisions requires timely, reliable, and accurate data. Often, this data must come from a wide variety of sources, which then require a certain amount of analysis before becoming useful. A number of the people who report using Python do so in a management capacity. They use Python to probe those disparate sources of information, perform the required analysis, and then present the big picture to the manager who has asked for the information. Given that this task occurs regularly, trying to do it manually every time would be time consuming. In fact, it would simply be a waste of time. By the time the manager performed the required work, the need to make a decision might already have passed. Python makes it possible to perform tasks quickly enough for a decision to have maximum impact.

Previous sections have pointed out Python's data-mining, number-crunching, and graphics capabilities. A manager can combine all these qualities while using a language that isn't nearly as complex to learn as C++. In addition, any changes are easy to make, and the manager doesn't need to worry about learning programming skills such as compiling the application. A few changes to a line of code in an interpreted module usually serve to complete the task.

As with other sorts of occupational leads in this chapter, thinking outside the box is important when getting a job. A lot of people need real-time analysis. Launching a rocket into space, controlling product flow, ensuring that packages get delivered on time, and all sorts of other occupations rely on timely, reliable, and accurate data. You might be able to create your own new job simply by employing Python to perform real-time data analysis.