Chapter 9

Choosing a Code Editor

Skills you will learn: This tutorial discusses the different options available for editing code.

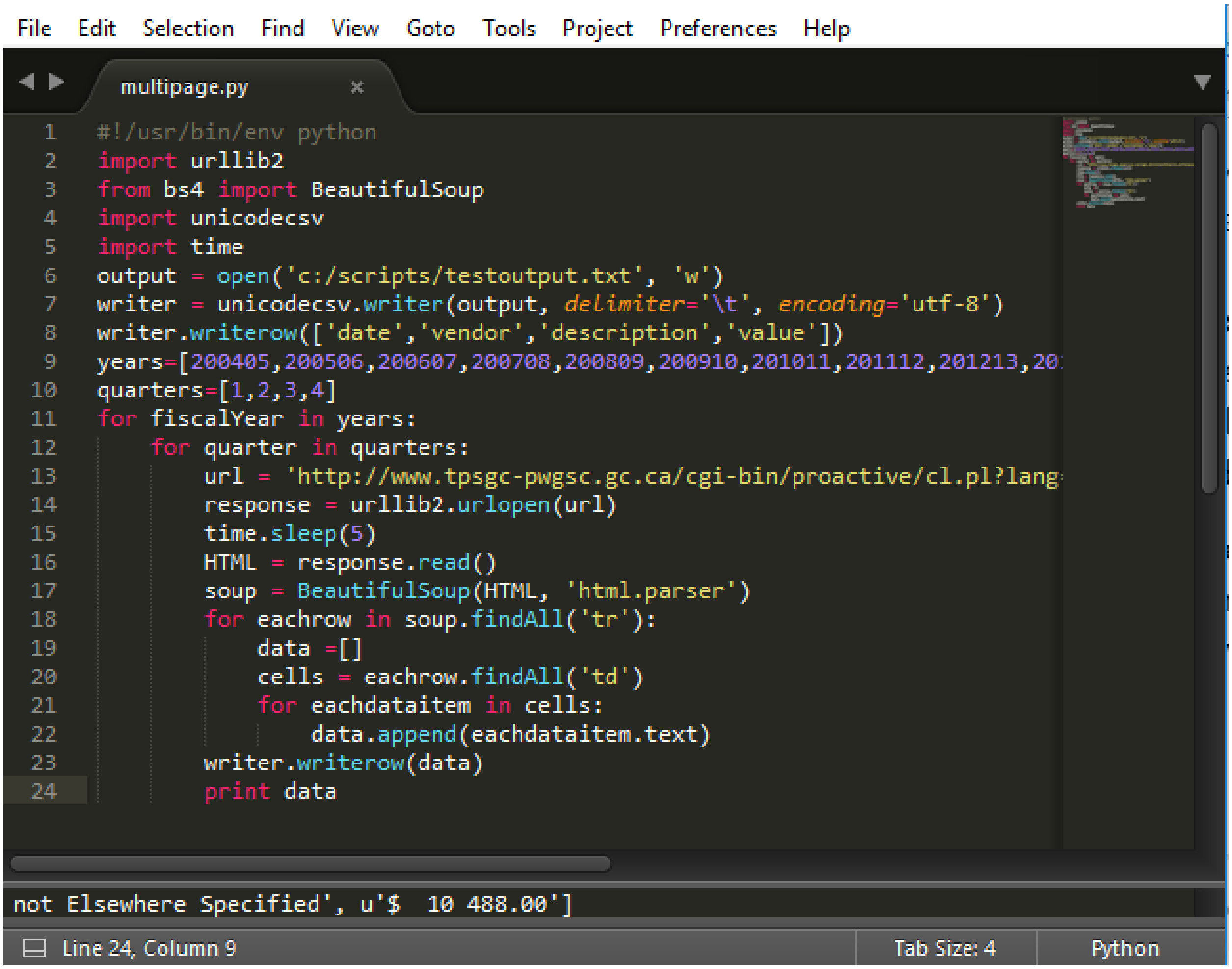
Different types of code editors

Computer code is normally written in plain text, and the range of possibilities for editing that text goes from bare bones to feature-packed. What you choose is a matter of personal preference, though as the complexity of code you write grows, you may wish to take advantage of the features available in the more advanced editors.

The simplest choice is a **plain text editor**, such as TextEdit on a Mac and Notepad on a Windows PC. The advantage of this type of editor is it comes with the computer and there are no complicated features to figure out. Just open the editor and start writing. It works just as well for HTML, CSS, JavaScript or Python. Save the file with the appropriate extension, then run it. It's as simple as that. For many who are just learning to write code, a plain text editor can be an excellent way to begin as you learn to write the code without any software assistance. If it works, great. If not, figure out where it's failing, make the changes, and try again.

If you are looking for a little help, a dedicated **code editor** may be a good choice. These add features such as highlighting code in various colours, so different elements, such as variable names, strings, numbers and code blocks stand out from one another. They may automatically indent Python code blocks, highlight errors in syntax, and autocomplete repetitive code. Some may even offer some limited ability to execute code without leaving the editing environment. In our experience, these kinds of features can definitely help make coding a more enjoyable experience by making structures and errors more obvious.

This image shows the popular Sublime Text code editor with one of the scripts used in our tutorials:



```python
#!/usr/bin/env python
import urllib2
from bs4 import BeautifulSoup
import unicodecsv
import time
output = open('c:/scripts/testoutput.txt', 'w')
writer = unicodecsv.writer(output, delimiter='\t', encoding='utf-8')
writer.writerow(['date','vendor','description','value'])
years=[200405,200506,200607,200708,200809,200910,201011,201112,201213,20
quarters=[1,2,3,4]
for fiscalYear in years:
    for quarter in quarters:
        url = 'http://www.tpsgc-pwgsc.gc.ca/cgi-bin/proactive/cl.pl?lang
        response = urllib2.urlopen(url)
        time.sleep(5)
        HTML = response.read()
        soup = BeautifulSoup(HTML, 'html.parser')
        for eachrow in soup.findAll('tr'):
            data =[]
            cells = eachrow.findAll('td')
            for eachdataitem in cells:
                data.append(eachdataitem.text)
        writer.writerow(data)
        print data
```

```
not Elsewhere Specified', u'$  10 488.00']
```

We aren't making a recommendation on which code editor to use, because everyone will have his or her favourite. If you are looking for recommendations, searching online using terms such as "best code editors" will bring up plenty of reviews. Features do change, so reading the most recent reviews is probably a good idea.

Free options include TextWrangler for Mac and Notepad++ for Windows computers. Other popular code editors include Sublime Text, Komodo Edit, and Atom.

With sophisticated debugging tools, integration with Github, and the ability to run code from within the editor, **integrated development environments** such as PyCharm and Komodo IDE offer the most advanced features and control such as debugging. They may be overkill for those writing simple scripts, but for professional developers they can greatly increase productivity. These programs can be quite expensive, but for those coding for a living, it can be more than worthwhile.