Exercise

Loading a pickled file

There are a number of datatypes that cannot be saved easily to flat files, such as lists and dictionaries. If you want your files to be human readable, you may want to save them as text files in a clever manner. JSONs, which you will see in a later chapter, are appropriate for Python dictionaries.

However, if you merely want to be able to import them into Python, you can serialize them. All this means is converting the object into a sequence of bytes, or a bytestream.

In this exercise, you'll import the `pickle` package, open a previously pickled data structure from a file and load it.

Instructions 100 XP

- Import the `pickle` package.
- Complete the second argument of `open()` so that it is read only for a binary file. This argument will be a string of two letters, one signifying 'read only', the other 'binary'.
- Pass the correct argument to `pickle.load()`; it should use the variable that is bound to `open`.
- Print the data, `d`.
- Print the datatype of `d`; take your mind back to your previous use of the function `type()`.

script.py — Light Mode

```
# Import pickle package
import pickle

# Open pickle file and load data: d
with open('data.pkl', 'rb') as file:
    d = pickle.load(file)

# Print d
print(d)

# Print datatype of d
print(type(d))
```

Run Code | Submit Answer

IPython Shell | Slides | Notes

```
{'June': '69.4', 'Aug': '85', 'Airline': '8', 'Mar': '84.4'}
<class 'dict'>


<script.py> output:
    {'June': '69.4', 'Aug': '85', 'Airline': '8', 'Mar': '84.4'}
    <class 'dict'>

In [1]:
```