Reading CSV Files

To pull data from a CSV file, you must use the reader function to generate a reader object.

The reader function is designed to take each line of the file and make a list of all columns. Then, you just choose the column you want the variable data for.

It sounds a lot more complicated than it is. To prove it, let’s take a look at an example.

*import CSV*

*With open(‘some.csv’, ‘r’) as f:*

*reader = csv.reader(f)*

*for row in reader:*

*print row*

Notice how the first command is used to import the CSV module?

Let’s look at another example.

*import csv*

*import sys*

*f = open(“somefile.csv” ‘r’)*

*reader = csv.reader(f)*

*for row in reader*

*print row*

*f.close()*

### Writing to CSV Files

When you have a set of data that you would like to store inside a CSV file, it’s time to do the opposite and use the write function. Believe it or not, this is just as easy to accomplish as reading them.

The **writer()** function will create an object suitable for writing. To iterate the data over the rows, you will need to use the **writerow()**function.

Here’s an example.

The following Python program converts a file called “test.csv” to a CSV file that uses tabs as a value separator with all values quoted. The delimiter character and the quote character, as well as how/when to quote, are specified when the writer is created. These same options are available when creating reader objects.

import csv

ifile  = open('test.csv', "rb")

reader = csv.reader(ifile)

ofile  = open('ttest.csv', "wb")

writer = csv.writer(ofile)

for row in reader:

    writer.writerow(row)

ifile.close()