10.10. Reading in data from a CSV File

We are able to read in CSV files the same way we have with other text files. Because of the standardized structure of the data, there is a common pattern for processing it. To practice this, we will be using data about olympic events.

Typically, CSV files will have a header as the first line, which contains column names. Then, each following row in the file will contain data that corresponds to the appropriate columns.

All file methods that we have mentioned - `read`, `readline`, and `readlines`, and simply iterating over the file object itself - will work on CSV files. In our examples, we will iterate over the lines. Because the values on each line are separated with commas, we can use the `.split()` method to parse each line into a collection of separate value.

Save & Run Original - 1 of 1

```
fileconnection = open("olympics.txt", 'r')
lines = fileconnection.readlines()
header = lines[0]
field_names = header.strip().split(',')
print(field_names)
for row in lines[1:]:
    vals = row.strip().split(',')
    if vals[5] != "NA":
        print("{}: {}; {}".format(
                vals[0],
                vals[4],
                vals[5]))

```

Activity: 1 -- ActiveCode (ac9_13_1)

Error

```
SyntaxError: bad input on line 1
```

Description

This message indicates that Python can't figure out the syntax of a particular statement. Some examples are assigning to a literal, or a function call

To Fix

Check your assignment statments and make sure that the left hand side of the assignment is a variable, not a literal or a function.

In the above code, we open the file, olympics.txt, which contains data on some olympians. The contents are similar to our previous olympics file, but include an extra column with information about medals they won.

We split the first row to get the field names. We split other rows to get values. Note that we specify to split on commas by passing that as a parameter. Also note that we first pass the row through the .strip() method to get rid of the trailing n.

Once we have parsed the lines into their separate values, we can use those values in the program. For example, in the code above, we select only those rows where the olympian won a medal, and we print out only three of the fields, in a different format.

Note that the trick of splitting the text for each row based on the presence of commas only works because commas are not used in any of the field values. Suppose that some of our events were more specific, and used commas. For example, “Swimming, 100M Freestyle”. How will a program processing a .csv file know when a comma is separating columns, and when it is just part of the text string giving a value within a column?

The CSV format is actually a little more general than we have described and has a couple of solutions for that problem. One alternative format uses a different column separator, such as | or a tab (t). Sometimes, when a tab is used, the format is called tsv, for tab-separated values). If you get a file using a different separator, you can just call the `.split('|')` or `.split('\\t')`.

The other advanced CSV format uses commas to separate but encloses all values in double quotes.

For example, the data file might look like:

```
"Name","Sex","Age","Team","Event","Medal"
"A Dijiang","M","24","China","Basketball","NA"
"Edgar Lindenau Aabye","M","34","Denmark/Sweden","Tug-Of-War","Gold"
"Christine Jacoba Aaftink","F","21","Netherlands","Speed Skating, 1500M","NA"
```

If you are reading a .csv file that has enclosed all values in double quotes, it’s actually a pretty tricky programming problem to split the text for one row into a list of values. You won’t want to try to do it directly. Instead, you should use python’s built-in csv module. However, there’s a bit of a learning curve for that, and we find that students gain a better understanding of reading CSV format by first learning to read the simple, unquoted format and split lines on commas.

10.9. CSV Format">
SV Format">
10.11. Writing data to a CSV File">

You have attempted 2 of 1 activities on this page

| Back to top