LECTURE 16

CSV

MCS 260 Fall 2021

Emily Dumas

REMINDERS

- Read the project 1 solution and ask questions!
- Project 2 due 6pm central time Fri Oct 8
- Project 2 autograder opens Mon Oct 4
- Change from worksheet 7 onwards: Problem 1 is for whole-lab discussion.

CSV

CSV is a format for storing tabular data in a text file.

Basic format: Each line contains some values, separated by commas. There is no universally accepted way to indicate types (e.g. string vs int).

```
fullname,midterm,final,hwk_avg
Maureen Singh,82.0,91.5,94.0
Yousuf Shaw,78.0,89.0,96.5
```

Often, the first line contains column headers.

CSV VS JSON

- JSON - for arbitrary data structures, especially hierarchies. Verbose if many dictionaries have the same keys. Typed.
  - *Mostly for talking to other programs.*
- CSV - for tabular data, i.e. text representation of a spreadsheet. Untyped.
  - *Mostly for interacting with spreadsheets and databases.*

READING CSV

`csv.reader(f)` returns an iterable that gives the rows one by one, as lists of values.

Use the return value in a `for` loop to process the file row by row.

Hit it with `list()` if you need the whole list a once (rare).

Important: When opening to read/write CSV you need to give `open()` an extra argument `newline=""`.

READING CSV (CONT'D)

If the CSV file has has headers, a better option is `csv.DictReader(f)` which yields rows as dictionaries, using column headers as keys.

`csv.DictReader` does not return the header row.

WRITING CSV

`csv.writer(f)` takes a file object and returns a writer object, which has a useful method:

- `.writerow(L)` — Write the items in iterable `L` to a row in the file.

Note: When opening to read/write CSV you need to give `open()` an extra argument `newline=""`.

WRITING CSV (CONT'D)

`csv.DictWriter(f,fieldnames=L)` specifies an iterable `L` of field names, and returns a writer object that expects rows as dictionaries. Useful methods:

- `.writeheader()` — Write the field names to a header row.
- `.writerow(d)` — Write the values from dictionary `d` to a line of the output file (but only the ones corresponding to keys that are field names).

OTHER FEATURES

The CSV reader and writer functions can use a separator other than a comma, e.g. specify `delimiter="\t"` to read or write *tab separated values (TSV)*.

Some CSV files put values in quotes so that the separator character can appear in the value, e.g.

```
fullname,occupation
Octavia Spencer,"actor,author"
"Bond, James Bond","spy"
```

The csv module supports this convention.

REFERENCES

- The `csv` module documentation is quite good, especially the Examples.
- A few public data sources offering CSV:
  - City of Chicago data portal
  - State of Illinois data portal
  - data.gov (US federal government data portal)
  - National Oceanic and Atmospheric Administration (NOAA) climate data portal hosts a lot of climate and weather datasets, some of which are available in CSV, e.g.
    - The Integrated Surface Dataset contains hourly weather observations from 35,000 worldwide stations, with archives going back to 1901.
    - e.g. This dataset shows it was -13.1°C = 8.4°F at Chicago O'Hare International Airport at Noon (central time) on December 25, 1950.

REVISION HISTORY

- 2021-09-29 Initial publication