LECTURE 30

THE CSV MODULE

MCS 260 Fall 2020

Emily Dumas

REMINDERS

- Project 3 is due Friday
- Quiz 10 due Today
- Worksheet 11 available
- Nov 3: No discussions
- Nov 5: Discussion converted to TA office hours

CSV

CSV is a format for storing tabular data in a text file. It is often used to export data from spreadsheets.

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.

READING CSV

The function `csv.reader(f)` takes a file object `f` and returns an iterable that yields a tuple for each row.

`csv.reader` will also accept an iterable of strings instead of a file object.

Common annoyance: `csv.reader` forces you to remember the number of the column you want.

Note: Should always pass `newline=""` to `open()` when opening to read/write CSV.

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. Must be in the proper order!

Note: Should always pass `newline=""` to `open()` when opening to read/write CSV.

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 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 8°F at Chicago O'Hare International Airport at Noon (central time) on December 25, 1950.

REVISION HISTORY

- 2020-11-03 Add note about `newline=""` when opening to read/write CSV.
- 2020-11-02 Add data portal links
- 2020-11-01 Initial publication