SCHOOL OF INFORMATION
UNIVERSITY OF MICHIGAN

Discussion 7: CSV Files, Nested Structures, and Debugging

SI 206: Data-Oriented Programming

Instructor: Dr. Barbara (Barb) Ericson
GSI: Kexuan (Michael) Huang
IA: Cristina & Jade

School of Information
University of Michigan

Fall 2023

Deadlines

- Project 1 due this Friday (10/13)
- Fall Break next Monday and Tuesday!
  - No lecture next Monday or Tuesday
  - No discussion next week
  - Nothing due next week

Table of Contents

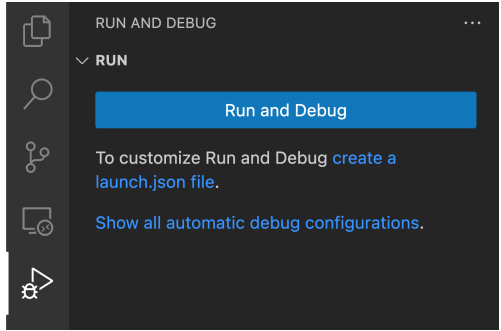
Table of Contents

Using a Debugger

- We don't want to print lots of lines. Use Debugger!
- Breakpoint: Stop at a particular line of code and see the values of variables
- Execute code line by line and see how variables change

VSCode Debugger

- Press the bug icon on the left side bar
- At the top, click Run and Debug
- Select "Python" if VSCode prompts for file type



VSCode Debugger Toolbar

- Continue: continue executing code until next error or breakpoint
- Step Over: stop after executing the function
- Step Into: stop before executing the first line in the function
- Step Out: exit the current function and stop before executing the line after the function is called
- Restart: restart the debugging of the program
- Stop: stop the debugger

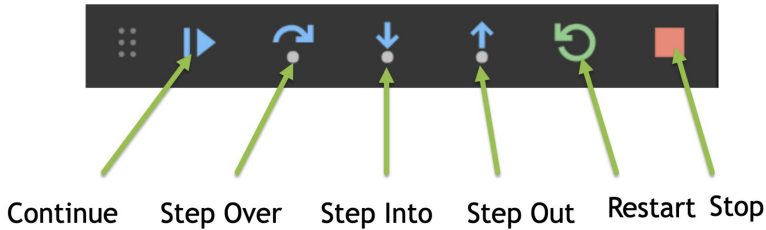


Table of Contents

CSV Reader

- Avoid problems: what if in some CSV files we have commas inside value?
- Quicker and easier to use
- Eventually you'll more likely to use Pandas for tabular data (But for now, do it this way!)

Example

```
1 import csv
2
3 with open('demo.csv') as csv_file:
4     csv_reader = csv.reader(csv_file) # read the file with csv reader
5     header = next(csv_reader)         # get the first line as a list
6     for cols in csv_reader:           # iterate through the rows
7         # here cols is a list of data in one row
```

CSV Writer

Example

```
1  import csv
2
3  with open('output.csv', 'w') as csv_file:
4      # write the file with csv writer
5      csv_writer = csv.writer(csv_file, delimiter=',', quotechar='"',
       ↪  quoting=csv.QUOTE_MINIMAL)
6
7      # iterate through the rows (a list of lists)
8      for cols in rows:
9          # here cols is a list of data in one row
10         csv_writer.writerow(cols)
```

Table of Contents

SCHOOL OF INFORMATION
UNIVERSITY OF MICHIGAN

Nested Dictionaries

How to represent a table in Python?

Example

```
Month,2020,2021,2022
JAN,304,316,306
FEB,395,359,369
MAR,404,387,421
APR,348,403,461
MAY,363,421,472
JUN,435,479,535
JUL,491,548,605
AUG,506,559,628
SEP,404,463,508
...
```

```
{
  '2020': { 'APR': '348',
            'AUG': '506',
            ... },
  '2021': { 'APR': '403',
            'AUG': '559',
            ... },
  '2022': { 'APR': '461',
            'AUG': '628',
            ... }
}
```

Table of Contents

Daily Visitors

daily_visitors.csv

```
1 Month,2020,2021,2022
2 JAN,304,316,306
3 FEB,395,359,369
4 MAR,404,387,421
5 ...
```

- Data on number of visitors to a public park each month from 2020 to 2022
- Have one header row on the first line
- CSV reader defaults all values to strings (only change when you need to)

Discussion 7 Exercise

Go to Canvas → Assignment → Discussion 7 and clone the GitHub Repo.

Your task

- Implement `load_csv()`: returns a nested dictionary
- Implement `get_annual_max()`: returns list of tuples with year, maximum value, and month
- Implement `get_month_avg()`: returns dictionary with year as key and month average as value

```
{'2020': {'JAN': '304', 'FEB': '395', 'MAR': '404', 'APR': '348', 'MAY': '363', 'JUN': '435', 'JUL': '491', 'AUG': '506', 'SEP': '404
', 'OCT': '487', 'NOV': '299', 'DEC': '337'}, '2021': {'JAN': '316', 'FEB': '359', 'MAR': '387', 'APR': '403', 'MAY': '421', 'JUN': '
479', 'JUL': '548', 'AUG': '559', 'SEP': '463', 'OCT': '407', 'NOV': '362', 'DEC': '314'}, '2022': {'JAN': '306', 'FEB': '369', 'MAR'
: '421', 'APR': '461', 'MAY': '472', 'JUN': '535', 'JUL': '605', 'AUG': '628', 'SEP': '508', 'OCT': '461', 'NOV': '390', 'DEC': '365'
}}

[('2020', 'AUG', 506), ('2021', 'AUG', 559), ('2022', 'AUG', 628)]

{'2020': 398.0, '2021': 418.0, '2022': 460.0}
```