

CSV_in_py

February 15, 2018

1 CSVs in Python

Python has great [documentation](#) for the csv module.

Sometimes you might have issues with the csv module and files that contain a lot of Unicode, especially if you're using Python 2. If you run into that issue, check out [unicodectsv](#).

For both CSV and JSON files, and others, the amazing [tablib](#) library is a great thing to check out.

```
In [1]: import csv
```

2 Creating a CSV reader

The newline argument with the '=' differentiates between actual newlines and /n within a quoted section.

```
In [3]: with open('definitions.csv',newline='') as csvfile:
        defreader = csv.reader(csvfile, delimiter=',')
        rows = list(defreader)
        for row in rows:
            print(', '.join(row))
```

word, definition

Antipathy, Deep dislike, sometimes without reason.

3 Creating a CSV DictReader

```
In [11]: with open('definitions.csv',newline='') as csvfile:
        defreader = csv.DictReader(csvfile, delimiter=',')
        rows = list(defreader)
        for row in rows:
            print(row['word'], ': ', row['definition'] )
```

Antipathy : Deep dislike, sometimes without reason.

4 Writing to a CSV (using fieldnames)

- `.writeheader()` writes fieldnames to the top of file

```
In [13]: with open('items.csv', 'a') as csvfile:
          fieldnames = ['item', 'location']
          itemwriter = csv.DictWriter(csvfile, fieldnames=fieldnames)

          itemwriter.writeheader()
          itemwriter.writerow({
              'item': 'iPhone',
              'location': 'Table'
          })
          itemwriter.writerow({
              'item': 'Sheets',
              'location': 'Bed'
          })
```