C: Reading CSV files

Comma-Separated Variable (CSV) format

Very common format for transferring data

1. Text files (can be read and written with Visual Studio Code)
2. Values are separated by commas (!)
3. Strings with embedded commas are enclosed in quotes
4. Often have variable names in first row

Example: Population by US county

File demodata.csv with four fields:

Field	Description	Type
name	Full name of county	str
pop	Population	int
fipsState	FIPS code for the state	⚠ **str** ⚠
fipsCounty	FIPS code for the county within the state	⚠ **str** ⚠

FIPS: Federal Information Processing Standard

- ⚠ Numeric but should **always** be used in **string** form
- ⚠ **Leading zeros** are significant and need to be retained
- ⚠ Spreadsheet programs often destroy FIPS codes

Same is true for US Zip codes: may have leading zeros

File contents:

Bold shows columns:

name,**pop**,fipsState,**fipsCounty**
"Adams County, Colorado",**497115**,08,**001**
"Adams County, Idaho",**4019**,16,**003**
"Adams County, Illinois",**66427**,17,**001**

Splitting into columns:

name	pop	fipsState	fipsCounty
"Adams County, Colorado"	497115	08	001
"Adams County, Idaho"	4019	16	003
"Adams County, Illinois"	66427	17	001

Reading using csv.DictReader():

Automatically builds a dictionary for each line

Used as follows:

1	`import csv`	Import csv module
2	`fh = open('demodata.csv')`	Open the file
3	`reader = csv.DictReader(fh)`	Set up the reader

4	`records = []`
	`for rec in reader:`
	`  rec['pop'] = int( rec['pop'] )`
	`  records.append(rec)`
		Loop through the **reader** building a list of dictionaries converting pop in the process

Result:

```
[
  {
    "name":"Adams County, Colorado",
    "pop": 497115,                          pop is an int
    "fipsState": "08",                      fipsState is a string
    "fipsCounty": "001"                     fipsCounty is a string
  },
  {
    "name":"Adams County, Idaho",
    "pop": 4019,
    "fipsState": "16",
    "fipsCounty": "003"
  },
  …
]
```