# CSV Files

## 1. Introduction

### What is a CSV File?

CSV files are plain text files which use specific format to store tabular data. CSV stands for "Comma Separated Values".

- Each line of the file is a data record.
- Each record consists of one or more fields, separated by commas.
- Normally first line of the file gives table header.

```
year,sex,type_of_course,no_of_graduates
1993,Males,Humanities & Social Sciences,481
1993,Males,Mass Communication,na
1993,Males,Accountancy,295
1993,Males,Business & Administration,282
```

### Why Uses CSV?

- CSV is a common format for data exchange because it is simple and compact.
- Most relational databases provides tools to import and export CSV files.
- CSV files can be easily opened in Excel.

### Python CSV Module

While we could use the built-in `open()` function to work with CSV files in Python, there is a dedicated `csv` module that makes working with CSV files much easier. It contains following built-in functions:

- csv.reader
- csv.writer
- writerow()

**Exercise:**

Find out all attributes, including properties and functions, in `csv` module.

In [4]:
```
1 import csv
2
3 csv.*?
```

## 2. Read CSV Files

## Using `csv.reader`

After opening a CSV file, create a `csv.reader` object which returns a iterable object to process CSV data.

- Each record is represented as a list.
- All fields are `string` type.

**Exercise:**

- Use `csv.reader` to read and print out all rows in `'olympics-medals-sample.csv'`

In [25]:
```
import csv
with open('./data/olympics-medals-sample.csv') as f:
    reader = csv.reader(f)
    for row in reader:
        print(row)
```

```
['NOC', 'Country', 'Total', 'Medal']
['USA', 'United States', '2088', 'Gold']
['URS', 'Soviet Union', '838', 'Gold']
['GBR', 'United Kingdom', '498', 'Gold']
['FRA', 'France', '378', 'Gold']
['GER', 'Germany', '407', 'Gold']
['AUS', 'Australia', '293', 'Gold']
```

**Question:**

Instead of printing out, how do you save all rows in `'olympics-medals-sample.csv'` into a list `data` ?

In [26]:
```
import csv
with open('./data/olympics-medals-sample.csv') as f:
    reader = csv.reader(f)
    data = [row for row in reader]

print(data)
```

```
[['NOC', 'Country', 'Total', 'Medal'], ['USA', 'United States', '2088', 'Go
ld'], ['URS', 'Soviet Union', '838', 'Gold'], ['GBR', 'United Kingdom', '49
8', 'Gold'], ['FRA', 'France', '378', 'Gold'], ['GER', 'Germany', '407', 'G
old'], ['AUS', 'Australia', '293', 'Gold']]
```

**Iterable Objector**

For any iterator object, you can use `next()` function to retrieve its next item.

**Try Code:**

```
obj = iter([1,3,5,7])
print(next(obj))
print(next(obj))
```

In [31]:

```
1 obj = iter([1,3,5,7])
2 print(next(obj))
3 print(next(obj))
```

```
1
3
```

**Skip Header Row**

Besides using list slicing, you can also use `next()` function to get first row, which is commonly its header.

**Exercise:**

- From file `'olympics-medals-sample.csv'`, retrieve header and data in separate lists.

In [32]:

```
1 import csv
2 with open('./data/olympics-medals-sample.csv') as f:
3     reader = csv.reader(f)
4     header = next(reader)
5     data = [row for row in reader]
6
7 print(header)
8 print(data)
```

```
['NOC', 'Country', 'Total', 'Medal']
[['USA', 'United States', '2088', 'Gold'], ['URS', 'Soviet Union', '838',
'Gold'], ['GBR', 'United Kingdom', '498', 'Gold'], ['FRA', 'France', '378',
'Gold'], ['GER', 'Germany', '407', 'Gold'], ['AUS', 'Australia', '293', 'Go
ld']]
```

## Optional Keyword Arguments

The `csv.reader()` function only has one required argument, which is the file object, but it has a couple of other optional arguments:

- **delimiter:** This argument specifies which delimiter the writer will use. It defaults to `','`, but you can set it to any other character.
- **quotechar:** This specifies which character will be used for quoting. It defaults to `'"'`
- **escapechar:** This specifies the character that will be used to escape the delimiter if quoting is not being used. It defaults to nothing.

**Exercise:**

Check out documentation of `csv.reader` function.

In [38]:
```
import csv
csv.reader?
```

### Delimiter

The character used to separate values is called a **delimiter**. Apart from comma ( `,` ), other delimiters include the tab ( `\t` ), colon ( `:` ) and semi-colon ( `;` ) characters.

For tab separated values, it is common to save it with extension `*.tsv` .

**Exercise:**

- Use `csv.reader` to read file `'olympics-medals-sample.tsv'` ; save both header and data in list.

In [39]:
```
import csv
with open('./data/olympics-medals-sample.tsv') as f:
    reader = csv.reader(f, delimiter='\t')
    header = next(reader)
    data = [row for row in reader]

print(header)
print(data)
```

```
['NOC', 'Country', 'Total', 'Medal']
[['USA', 'United States', '2088', 'Gold'], ['URS', 'Soviet Union', '838',
'Gold'], ['GBR', 'United Kingdom', '498', 'Gold'], ['FRA', 'France', '378',
'Gold'], ['GER', 'Germany', '407', 'Gold'], ['AUS', 'Australia', '293', 'Go
ld']]
```

# 3. Write CSV Files

### Using `csv.writer`

A `csv.writer` can be used to write a CSV file. The `csv.writer()` function returns a `writer` object that converts the user's data into a delimited string and write to file using its `writerow()` function.

The `newline` argument is set to '' when opening a file which the `csv.writer` will write each row in a line.

**Exercise:**

- Use `csv.writer` to save following data into a csv file `'sample.csv'` .

```
["Symbol", "Name", "Price (Intraday)"]
["TMVWY", "TeamViewer AG", 21.05]
["AXSM", "Axsome Therapeutics, Inc.", 88.87]
["SAGE", "Sage Therapeutics, Inc.",    53.36]
```

In [5]:
```
import csv

data = [["Symbol", "Name", "Price (Intraday)"],
["TMVWY", "TeamViewer AG", 21.05],
["AXSM", "Axsome Therapeutics, Inc.", 88.87],
["SAGE", "Sage Therapeutics, Inc.",    53.36]]

with open('sample.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    for row in data:
        writer.writerow(row)
```

In [6]:
```
!notepad sample.csv
```

### Write Multiple Rows

The `writerows()` function of writer allow you to write 2-dimensional list to a CSV file.

**Exercise:**

Save following data to a csv file `stock_sample.csv` using `csv.writer` .

```
[['stock', 'price', 'cost', 'profit'], ['21', '121.34', '45.34', '76']]
```

In [70]:
```
import csv
csv_rowlist = [['stock', 'price', 'cost', 'profit'], ['21', '121.34', '4
with open('stock_sample.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(csv_rowlist)
```

In [71]:
```
!notepad stock_sample.csv
```

# 4. Open Data

## Data.gov.sg

The Singapore government's one-stop portal to publicly-available datasets from 70 public agencies. It is an open repository of data captured by the public sector. It helps people understand the data using visualizations and articles, and provides realtime APIs for developers to use.

https://data.gov.sg/ (https://data.gov.sg/)