Introduction to Computer Programming

Week 7.2: Modules for reading & writing files

The `writer` class from the `csv` module handles delimiters automatically.

`writerow` :

- a method belonging to the `writer` class
- writes list to csv file as row

Example: Write the high score table to a csv file scores_.csv

In [95]:

```python
names = ['Elena', 'Sajid', 'Tom', 'Farhad', 'Manesha']
scores = [550, 480, 380, 305, 150]

with open('sample_data/scores_.csv', 'w', newline='') as f: # f

    w = csv.writer(f)            # writer object is created

    for n, s in zip(names, scores):
        w.writerow([n, s])
```

Try it yourself

Example: Use the `csv` module to write the header and first row of the high score table shown to a .txt file.

Hint

.txt file so delimiter is `' '` not `','` (default).

Specify delimiter when creating the writer object using the named argument `delimiter=' '`.

Place	Name	Score
1	Elena	550
2	Sajid	480
3	Tom	380
4	Farhad	305
5	Manesha	150

In [96]:

```python
import csv

with open('sample_data/scores_.txt', 'w') as f:

    w = csv.writer(f, delimiter=' ')

    w.writerow(['place', 'name', 'score'])

    w.writerow([1, 'Elena', 500])


```

Example: Write the high score table data to a csv file, scores_.csv

`writerows` can be used to write a data structure (containing multiple lists) to the file

In [97]:

```python
header = ['place', 'name', 'score']

# data is list of lists
data = [[1, 'Elena', 550],
        [2, 'Sajid', 480],
        [3, 'Tom', 380],
        [4, 'Farhad', 305],
        [5, 'Manesha', 150]
       ]
```

In [98]:

```
import csv

with open('sample_data/scores_.csv', 'w') as f:

    w = csv.writer(f)

    w.writerow(header)

    w.writerows(data)
```

Reading CSV files

Reading csv files and converting them to a useful format (list of strings) can be a lengthy process!

In [100]:

```
with open('sample_data/scores_.csv') as f:
    file = list(f)                              # list of strings (

    print(file)

    lines = [line.strip() for line in file] # remove \n from ea

    print(lines)

    lists = [l.split(',') for l in lines]   # convert each line

    print(lists)
```

```
['place,name,score\n', '1,Elena,550\n', '2,Sajid,480\n', '3,Tom,38
0\n', '4,Farhad,305\n', '5,Manesha,150\n', '6,Carl,100\n', '7,Theo
,105\n', '8,Mark,75\n', '9,Grace,50\n']
['place,name,score', '1,Elena,550', '2,Sajid,480', '3,Tom,380', '4
,Farhad,305', '5,Manesha,150', '6,Carl,100', '7,Theo,105', '8,Mark
,75', '9,Grace,50']
[['place', 'name', 'score'], ['1', 'Elena', '550'], ['2', 'Sajid',
'480'], ['3', 'Tom', '380'], ['4', 'Farhad', '305'], ['5', 'Manesh
a', '150'], ['6', 'Carl', '100'], ['7', 'Theo', '105'], ['8', 'Mar
k', '75'], ['9', 'Grace', '50']]
```

The `writer` class from the `csv` module removes delimiters and `\n` characters automatically.

Example: Read the contents of the file scores_.csv

In [101]:

```
with open('sample_data/scores_.csv') as f:

    r = csv.reader(f)

    for line in r:      # iterable
        print(line)

    # stream position at end of file

    f.close()
```

```
['place', 'name', 'score']
['1', 'Elena', '550']
['2', 'Sajid', '480']
['3', 'Tom', '380']
['4', 'Farhad', '305']
['5', 'Manesha', '150']
['6', 'Carl', '100']
['7', 'Theo', '105']
['8', 'Mark', '75']
['9', 'Grace', '50']
```

Example: Read the file scores_.csv and print the third line.

In [102]:

```
import csv

with open('sample_data/scores_.csv') as f:

    r = csv.reader(f)

    r = list(r)

    print(r[2])
```

```
['2', 'Sajid', '480']
```

Reading and writing csv files

Example: Add a new entry to the file scores_.csv and then print the contents.

What if we want to print the contents before adding a new line?

In [103]:

```python
with open('sample_data/scores_.csv', 'a+') as f:

    r = csv.reader(f)
    w = csv.writer(f)


    w.writerow([10, 'Lois', 70])

    f.seek(0)

    for line in r:
        print(line)
```

```
['place', 'name', 'score']
['1', 'Elena', '550']
['2', 'Sajid', '480']
['3', 'Tom', '380']
['4', 'Farhad', '305']
['5', 'Manesha', '150']
['6', 'Carl', '100']
['7', 'Theo', '105']
['8', 'Mark', '75']
['9', 'Grace', '50']
['10', 'Lois', '70']
```