

## Lab: Files

This lab will involve loading temperatures from a CSV file into instances of a class `temperature`.

You should now create a program to load data from a file (rather than requiring the user to enter values or hard coding those values in the program).

This lab is comprised of 6 steps:

1. Create the `loader.py` module
2. Create the data file
3. Prompt the user for the filename to load
4. Load the data file
5. Invoke the `load_data` function

### Step 1: Create the `loader.py` module

Create a new file to define the temperature class and a load function in.

To do this add a new module / file to your application called `loader.py`.

At the start of the file add the required imports, for example:

```
import csv
```

The class `temperature` to be used is given below:

```
class TemperatureReading:
    """ Class representing temperature info """

    def __init__(self, temp, date, location, scale=CELSIUS):
        self.temp = temp
        self.date = date
        self.location = location
        self.scale = scale

    def __str__(self):
        return f'TemperatureReading[{self.scale}]({self.temp} on {self.date} at {self.location})'

    def __repr__(self):
        return f'Temperature({self.temp}, {self.date}, {self.location}, {self.scale})'
```

Use the `data.csv` file in the labs directory for this, this file has the format:

- Temperature, scale, date and location.

An example of the contents of the file is given below:

```
13.5,Celsius,01/05/20,London
12.6,Celsius,02/05/20,London
```

```
15.3,Celsius,03/05/20,London
12.2,Celsius,04/05/20,London
16.6,Celsius,05/05/20,London
14.6,Celsius,06/05/20,London
15.6,Celsius,07/05/20,London
```

Save this data into a file in the same directory as your program.

### Step 3: Prompt the user for the filename to load

The program should ask the user for the name of the file to load. You can allow the user to press enter to use a default file name (such as 'data.csv'). For example:

```
filename = input('Please enter the data file to load
(data.csv): ')
if filename == '':
    filename = 'data.csv'
```

### Step 4: Load the data file

Write a function to load the data from a csv file. To do this write a function called `load_data()` in the module (file) called loader.

The `load_data()` function should take the name of the file to load.

It should then use the csv library to load data from a CSV file into a list container type.

To do this you should use the `with` as language construct when opening the file. You can then use the `csv.reader()` function to read the contents of the file into a reader object. You can then loop through each row in turn in the reader.

Each row should contain 4 items of information. If a row does not contain 4 items of information, then you should log this for the user to investigate at a later stage.

When you load a row, you should use it to create a new instance of the `TemperatureReading` class, for example:

```
temp = float(row[0])
scale = row[1]
date = row[2]
location = row[3]
reading = TemperatureReading(temp, date, location, scale)
```

The `load_data()` function should return a list containing a list of instances representing the rows in the CSV file.

### Step 5: Run the following code

```
filename = input('Please enter the data file to load (data.csv): ')
if filename == '':
    filename = 'data.csv'
```

```
temperatures = load_data(filename)
print(temperatures)
```