Chapter 9

Parsing text files (Homework for week 3)

The KNMI (Royal Dutch Meteorological Institute) offers a dataset for download that covers weather observations from the Dutch weather stations. In this assignment you will build a program to load the data that the KNMI offers and convert it to appropriate Python data types. The data is available from the KNMI website at: `http://projects.knmi.nl/klimatologie/daggegevens/selectie.cgi` In building this program you will encounter the basic Python datatypes, some flow control and iteration. The goals for this week are to:

- read a text data file (provided by KNMI),
- use the basic Python types and conversions between them,
- properly name variables in your program, and
- to structure a simple program.

This exercise will be graded as follows, you can earn at most 9 points by completing the exercise and handing in on time. Your grade g, awarded on a scale from 1 to 10, is calculated as follows:$g = 1 + s$, where s are the points you earned. For this exercise the points are awarded for the following things:

(3 points) Style, in this case appropriate variable naming, code layout and comments.

(4 points) For completing Section 9.1 correctly.

(2 points) For completing Section 9.2 correctly.

Note you cannot use pre-built CSV readers of any kind, this exercise is about building a simple one yourself.

9.1 Assignment part 1

- Go to the KNMI webpage, and download all the data you can. Make sure that the data you download starts at 1901 01 01 and runs through to today, for all weather stations in the Netherlands. Save the text file to an appropriate directory. The file you download will in the 100 MB - 200 MB size range.
- When building a program it is convenient to start with a small amount of data so that you program runs quickly when you are still looking for errors. Use one of the commandline tools you learned about and grab the first 500 lines of the data file and write them to a new file. You will now have an appropriately small dataset that contains both a header and a data part.

- Start a new Python file, possibly using the template from Section 9.3, name the file `read_data.py`. Make sure to add you name and student number to this file.
- Look up how you can access a file line-by-line (see Chapter 7). Find out how you can split a string on a certain character. Using this write a simple program that opens the small data file, skips the header, reads every line, splits it on an appropriate character and turns the strings into numbers.
- Start a document as well (A4-sized PDF). How do you deal with missing data and why? What are appropriate values if you choose to use a number to represent missing data? (Add to the report, refer to this question.)
- Change the program you wrote so far. The function `read_knmi_data_file` should return a list of lists of numbers. Where each inner list corresponds to line in the data file with appropriate values for missing data.
- The header consists of 3 parts. Separate these out into several files that we will use in building a programs to read the header.
- Build a program that can read the part of the header where the stations and their locations are defined. The format of this data should be a list containing one list for each station. That list for each station should contain 5 entries (station number, longitude, latitude, altitude, its full name) in appropriate data types. Explain in you report why you chose the data types that you did. Call this program `station_locations.py`.
- Build a program that can read the part of the header that explains the meaning of the column names. The output of this program should be a dictionary mapping the column name to its description. Call this program `column_meaning.py`.
- Build a program to read the line of column headers, its output should be a list of column names.

9.2 Simple analysis part 2

If your program that reads the data entries and skips the header works on the small data file. Create a new Python program called `analysis.py`, copy to it the aforementioned program and adapt it to answer the following questions.

- Find the wettest day in your data set, find the hottest day. Where were these records broken? Note these can be found by looping through the data set and keeping track of the highest encountered value (of e.g. temperature) and the date that it occured at.
- Compare the summers in "De Kooy" with those in "Valkenburg". Calculate monthly averages for min, max temperature and the amount of precipitation for 2016. Add a table to your report with these monthly average for 2016.

9.3 Hints

Below you find a template that you can use as a start for this exercise. If you choose to use it, copy it to a Python source file and run it. It should give no errors before you move to the rest of the exercise. Note that copying and pasting the program below will probably not work.

```
#!/usr/bin/env python
"""
Parse KNMI data and do simple analysis.
"""
# this program scaffold shows you how you can structure a program

def read_knmi_data_file(filename):
    """Read KNMI data"""
    print 'Loading the file', filename
    data = []
    # you write the data reading part here
    return data


def main(filename):
    """Run program for exercise of week 2"""
    data = read_file(filename)
    # you can work with the data here


if __name__ == '__main__':
    main('knmi-data.txt')
```