INDEX PAGE
Pages
Lab 1
FAQ
Lab 2
Lab 3
Lab 7
Lab 6
Testing tips
Lab 4
Lab 10
Lab 5
Module overview
Support
Installing and finding Python
Snippets
Lecture slides
Lab 8
Lab 9
First steps
Testing demo

Home | **Lab 4**

Laboratory 4: File Input/Output

Prerequisites: as in lab3, +file i/o, type conversion

Contents

PEP8

To monitor the coding style we turn on the PEP8 style guide monitoring in Spyder by:

1. Going to preferences in Spyder menu
2. Clicking on `Editor` in the selection list on the left
3. Clicking on `Code Introspection/Analysis` (top right corner)
4. Ticking the box `Real-time code style analysis`
5. Clicking `Apply` and `OK` (bottom right corner)

Exercises

Create a file `lab4.py` which provides:

1. A function `line_averages(filename)` that takes a string `filename` which contains the name of a file to be processed. The function should open and read that file. The file is expected to contain numbers that are separated by commas (the format is known as a comma-separated-value file, short csv). The function should compute the average value for every line, and return the average values in a list. For example, if we call `line_averages("data.csv")` and file data.csv reads:

   ```
   1,2
   1,1,1,1
   -1,0,1
   42,17
   ```

   then the function should return the list `[1.5, 1.0, 0.0, 29.5]`.

2. The U.S. National Oceanic and Atmospheric Administration (NOAA) provides observations of current weather conditions around the globe. Using the special 4-letter station code for Southampton -- which is is EGHI -- we can find quantitative information on current weather conditions by pointing a web browser to: http://tgftp.nws.noaa.gov/data/observations/metar/decoded/EGHI.TXT

   We provide the following function (to be included in `lab4.py`) which downloads this webpage and returns it as a string:

   ```
   def noaa_string():
       url = "http://tgftp.nws.noaa.gov/data/observations/metar/decoded/EGHI.TXT"
       noaa_data_string = urllib.request.urlopen(url).read()
       return noaa_data_string.decode("utf-8")
   ```

   The library `urllib.request` allows to access a webpage like a file through its `urlopen()` function, and you should include `import urllib.request` at the beginning of your file `lab4.py`.

   Call the function `noaa_string` from the Python prompt and inspect the return value.

   Your task is to write a function `noaa_temperature(s)` which should take a string `s` as returned from `noaa_string()` as the input argument, extract the temperature in degree Celsius from the string, and return this temperature as an integer number:

   ```
   In [ ]: noaa_temperature(noaa_string())
   Out[ ]: 10
   ```

NOAA may at times change the number and order of lines in the data, but you can assume that the format of the line containing the temperature data does not change.

3. A function `seq_sqrt(xs)` which takes a list of non-negative numbers `xs` with elements `[x0, x1, x2, ..., xn]`, and returns the list `[sqrt(x0), sqrt(x1), sqrt(x2), ..., sqrt(xn)]`. In other words, the function takes a list of numbers, and returns a list of the same length that contains the square root for each number in the list.

4. A function `mean(xs)` that takes a sequence `xs` of numbers, and returns the (arithmetic) mean (i.e. the average value).

   Example:

   ```
   In [ ]: mean([0, 1, 2])
   Out[ ]: 1.0
   ```

5. A function `wc(filename)` that returns the number of words in file `filename`. The name `wc` stands for Word Count. To split a string `s` into words, use `s.split()` for this exercise (i.e. the behaviour of the `split()` method is here used to define what a word is).

   Example 1: For a file `data.txt` with content:

   ```
   One Two
   ```

   a function call `wc('data.txt')` should return `2`.

   Example 2: For a file `data.txt` with content:

   ```
   One Two
   Three Four Five
   ```

   a function call `wc('data.txt')` should return `5`.

   You can test your function on the "Alice in Wonderland" book (available at http://www.gutenberg.org/files/28885/28885-8.txt), and should expect that this has several ten-thousand words.

Then submit lab4.py by emailing it to feeg1001@soton.ac.uk with the subject lab 4 for automatic testing of this laboratory session.

Return to Top

Page last modified Thu Nov 1 16:40:01 2018. |