

# 582332: Programming in Python, Fall 2009

Exercise sheet 6  
12-16.10.2009

1. Give regular expressions that can match exactly the following sets of strings:
  - Recognise the strings that represent Python's complex numbers with decimal integer real and imaginary parts. For example, the strings "2j+3" and "5 -1j" should be recognised. No need to evaluate any expressions. The real and imaginary parts are always present and are separated by the sign of the second part.
  - Python's lists of integers. For example the string "[ 123,32, 45 , -1]" should be recognised. Spaces or tabs can be used to separate the items.
2. Use the Unix command 'ls -l' and store its results in a file. Then, using regular expressions, extract from this file the filenames and corresponding file sizes.

For example, here is a part of a listing run on my current directory:

```
-rw-r--r-- 1 jttoivon tkol 85883 2009-09-18 19:20 administration.handout.pdf
-rw-r--r-- 1 jttoivon tkol 91654 2009-09-08 20:01 administration.pdf
-rw-r--r-- 1 jttoivon tkol 167529 2009-09-29 20:21 alice.txt
```

The output lines consists the following fields separated by whitespace: access rights, link count, owner's id, owner's group id, file size, date, time, and the filename. The result should be an iterator over tuples, where the tuples have the filename element and the size element. The above listing would result an iterator, that gives the elements of the following list:

```
[('administration.handout.pdf', 85883), ('administration.pdf', 91654), ('alice.txt', 167529)]
```

3. HTML code has six levels of headings: **h1**, **h2**, ..., **h6**, where **h1** is the topmost heading and **h6** is the lowest level heading. In HTML files the headings are written between the begin `<h $level$ >` and end `</h $level$ >` tags. For example, `<h1>This is a title</h1>` creates a level 1 heading. It makes no difference whether a lowercase or uppercase h is used.

Create a program that gets a filename of an HTML file as a parameter, and, using regular expressions, returns a list of titles, for each level 1, ... 6. You can, for example, try the program with the course homepage: <http://www.cs.helsinki.fi/u/jttoivon/python-09/index.html>. Store this page to a file and run it through your program.

Your program doesn't have to work in all special cases that might occur when dealing with HTML files, it's enough that it works, for example, with the course homepage.

4. Modify the previous program so that instead of printing the headings, create a new HTML page that changes all the headings to be written in red.

In HTML you can ask a piece of text to be written in red by surrounding it with the beginning tag `<font color="red">` and end tag `</font>`.

The program should work like this, for example with the file `index.html`: `python color.py index.html`. This will create a file called `index.new.html`. Try viewing this file from Firefox using the open file entry from the file menu.

5. Let's assume we have a dictionary `d`, whose keys and values are strings. And especially, you can assume that the key strings don't contain any punctuation or other special characters.

Create a function `dict_replace(str, d)` that returns a string where each occurrence of a key `k` of `d` in `str` is replaced by the value `d[k]`. For example, with `d={"one" : "yksi", "three" : "kolme"}` and `str="one two three"` the call to `dict_replace` would result the string "yksi two kolme".

If occurrences of keys  $k_1$  and  $k_2$  in text are overlapping, then replace only the occurrence that starts earlier in the string. If these occurrences start at the same point, replace only the key that comes first in the iteration order of the dictionary. Notice that this order is not stable or portable, and one therefore really should avoid overlapping occurrences.

Do this by using `re.sub` and call it only once! *Hint*. The replacement parameter for `re.sub` doesn't need to be a string.

*Extra question:* How would you do this if the keys of the dictionary were allowed to contain also punctuation and other special characters?

If time remains, voluntary guidance for project work is offered by the assistants until the exercise class is over.

After you have done the exam (20th of October), please use the following form to give feedback:

<https://ilmo.cs.helsinki.fi/kurssit/servlet/Valinta?kieli=en>

This information is used to develop teaching at the department.