

Computer Systems and -architecture

Regular expressions and sed

1 Ba INF 2017-2018

Stephen Pauwels
stephen.pauwels@uantwerpen.be

Time Schedule

Exercises are made individually. Put all your files in a tgz archive, as explained on the course's website, and submit your solution to the exercises on Blackboard.

- Deadline: **October 12, 23u55**

Exercises

Fill in all regular expressions in the file `oefeningen.html`, and state clearly when you use the extended form.

1. Find a regular expression that matches a number at the beginning of a line.
2. Find a regular expression that matches filenames with a "tar.gz" extension.
3. Find a regular expression that matches all words of 4 characters long.
4. Find a regular expression that matches any number between 1 and 999.
5. Find a regular expression that matches dates of the form:
31/08/1933
2-03-2002
09 4 1966
15.12.1999
6. Find a regular expression that matches an IPv4 number (0.0.0.0 to 255.255.255.255).
7. Find a regular expression that matches hexadecimal representations of the form:
0x2a
0XF
0X1111
0x0

8. Find a regular expression that matches floating point numbers. Some examples of floating points are:

12.245
-234
+.0009
3.11e33
43.1E11
2e-14

9. Find a regular expression that matches strings surrounded by square brackets. Beware of greedy evaluation! For example, the HTML string "Hello, `this` is emphasized." should match twice, for `` and for ``.
10. Find a Sed command that extracts HTML tags (without attributes or nested tags) from a text. A text must be converted as follows:

```
<h1>This is a valid HTML tag</h1>.  
<i>These</b> <1>invalid</i> <a>tags</a>> should be ignored.
```

Becomes:

```
This is a valid HTML tag.  
<i>These</b> <1>invalid</i> <a>tags</a>> should be ignored.
```

11. Find a sed expression to trim unnecessary whitespace [space or tab] from the start and the end of a string.