# Regular Expressions — Activities

1. Print out all lines in "alice.txt" that contain a double hyphen.

2. Assume that you dumped a directory listing into a temp file that contains one file name per line. Use a regular expression to print out all lines that have a combination of something, a dot (remember to escape) and 'py' in them.

3. Find all lines in 'alice.txt' that contain the following pattern: The first letter is a 't'. The second letter is any small letter. The third letter is either a 't' or an 'l'. The fourth letter is either a 't' or an 'l',

4. Use the following pattern to find the proscribed matches in the text file “alice.txt”:

```
match = re.search(pattern, text)

s = match.start()
e = match.end()

print('Found "{}"\nin "{}"\nfrom {} to {} ("{}")'.format(
    match.re.pattern, match.string, s, e, text[s:e]))
```

   (a) All lines that contain a double hyphen `'--'`
   (b) All lines that contain a word of length 10 or more. (Hint: \w means a letter and braces afterwards specify the length.)
   (c) All lines that contain a word of exact length 10. (Hint: We are looking for a pattern of non-letter (\W) followed by 10 letters (\w) followed by a non-letter.)
   (d) All words starting with “dou” and finishing with “lly”. (Hint: We use the same trick, but now we specify the non-letter, the beginning letters followed by an arbitrary number of letters, followed by the last three letters followed by a non-letter.

5. Find all the tags in “article.html". A tag is given by opening and closing < >-symbols. Here, we need to insure that we match as short a string as possible. If we use the regular expression `r'<.>'`, then we will match all of “ `<h1>A Level One Title</h1>`”. This is the greedy aspect of regular expression matching. However, if we use the non-greedy modifier ‘?’ (a single question mark), then we match the shortest string possible.