Brief Introduction to Regular Expressions

A regular expression (or regex) is a sequence of characters that defines a string-matching pattern.

To work with regular expressions in Python, you can use the built-in package `re`.
Note that the regex syntax varies from language to language.

In this lab, a regex was used in order to match tokens in the text:

```
wordRE = re.compile('[A-Za-z]+')
```

The regex above matches all sequences of capital (`'[A-Z]'`) or lowercase letters (`'[a-z]'`) that appear at least once or more (`'[A-Za-z]+'`). This `+` symbol is called Kleene plus.

If we want to match exactly one capital or lowercase letter, the regex should be `'[A-Za-z]'`. Alternatively, if we want a sequence of e.g. lowercase letters to appear zero or more times, we will use a different symbol called Kleene star: `'[a-z]*'`.

In the variable `wordRE` we have stored the pattern that we want to match. Later in the code, we access each of the matching sequences:

```
for word in wordRE.findall(line.lower()):
```

The code above detects all the sequences that match the pattern stored in `wordRE` by looking at the `line` string. Because the `line` string contains both capital and lowercase letters, we prefer to convert it into lowercase (using the method `.lower()`), since we want the tokens to be treated regardless of their case.
This also means that the code should work even if we used this regular expression instead:`'[a-z]+'`, as the string we are using (`line`) only contains lowercase letters.

More details about regex:

- Lab Class Week 9
- Jurafsky & Martin, Speech and Language Processing, Ch. 2.1