# EPAM Python Software Engineer Training

## Lesson 2: Regular expressions in Python

# Contents

# 1  Course

## 1.1  Regular Expressions

Python supports a perl-like syntax for regular expressions with some deviations. All routines to work with regular expressions are present in a *re* module.

# 2  Tasks

If not specified otherwise for all tasks within this course:

- an `alice.txt` file shall be used as an input (present in the same folder);

- all searches and substitutions shall be done using regular expressions only;

- all output shall be written to an `alice00.txt` file (suffixed with a task number) placed in a lesson folder;

## 2.1  Empty

Remove all empty lines in a file and print a number of removed lines.

## 2.2  Blank

Replace all blank lines (lines consisting of just a white space) and print a number of modified lines. Empty lines shall not count.

## 2.3  White Space

Remove all leading and trailing white-space from a file and print a number of modified lines.

## 2.4  Vocals

Print a number of vocal letters in first 100 lines of a file.

## 2.5  Numbers

Print a number of numbers in a file; each number shall count only once (e.g. `1234` shall count only once, not 4 times).

## 2.6  Doubles

Print a number of all occurences of double characters in a file (e.g. `ee`).

## 2.7  Advanced doubles

The same task as above but tripples shall not count (e.g. `eee` shall not count).

## 2.8  Sentence

Print a number of sentences in a file (a sentence shall and in either a dot `.` or a tripple-dot `....`

## 2.9  Words

Print a number of words in a file (words can be separated by either white space or any separator (e.g. `,` or `-`). Pure integers shall not count but identifiers consisting of a mix of characters and integers shall count).

## 2.10  Time

Replace each occurence of `Alice was` to `Alice is` and print a number of modified phrases; sentences breaking though lines shall be modified correctly as well.