cucumber school

# Lesson 3 Answers

*Matching Steps*

## 1 - Reflection questions

### 1.1 - Purpose

We need regular expressions to link automation code to plain text Gherkin steps. Regular expressions gives us flexibility to match variations of steps.

### 1.2 - Benefits of regular expressions

Regular expressions can be used to define synonyms, or deal with plural and singular. This makes it easier to design a flexible domain language that reads well.

This flexibility also makes it possible to reuse step definitions across different Gherkin steps with slight variations. This means fewer step definitions to maintain.

### 1.3 - Capture groups

A regular expression *capture group* is a pair of parenthesis. It lets us capture interesting part of a piece of text. We can use capture groups to pass arguments to a step definition.

### 1.4 - Wildcards

Wildcards can be used to match different variations of text. For example, the `.` will match any character. A `\d` will match any digit.

### 1.5 -Test runs

If we make a lot of changes between each time we run our tests, it can be difficult and time consuming to identify the cause of a failing test.

This is why we prefer to run our tests as often as possible. This means our tests have to be *fast*. We'll learn more about how to keep tests fast in a future lesson.

## 2 - Exercises

### 2.1

There are several possible answers. Here are some of them:

- *Lucy shouts back to (Sean|Sarah)*
- *Lucy shouts back to (\w+)*
- .*

### 2.2

- *Sarah has \d+ cucumbers in her bag*

### 2.3

They all match - except for the 3rd:

- *Sarah buys  -3 cucumbers*

A slight modification to the regular expression would make it match them all:

- Sarah buys (no|-?\d+) cucumbers?

We highly recommend http://rubular.com/ and http://scriptular.com/ for playing with regular expressions!