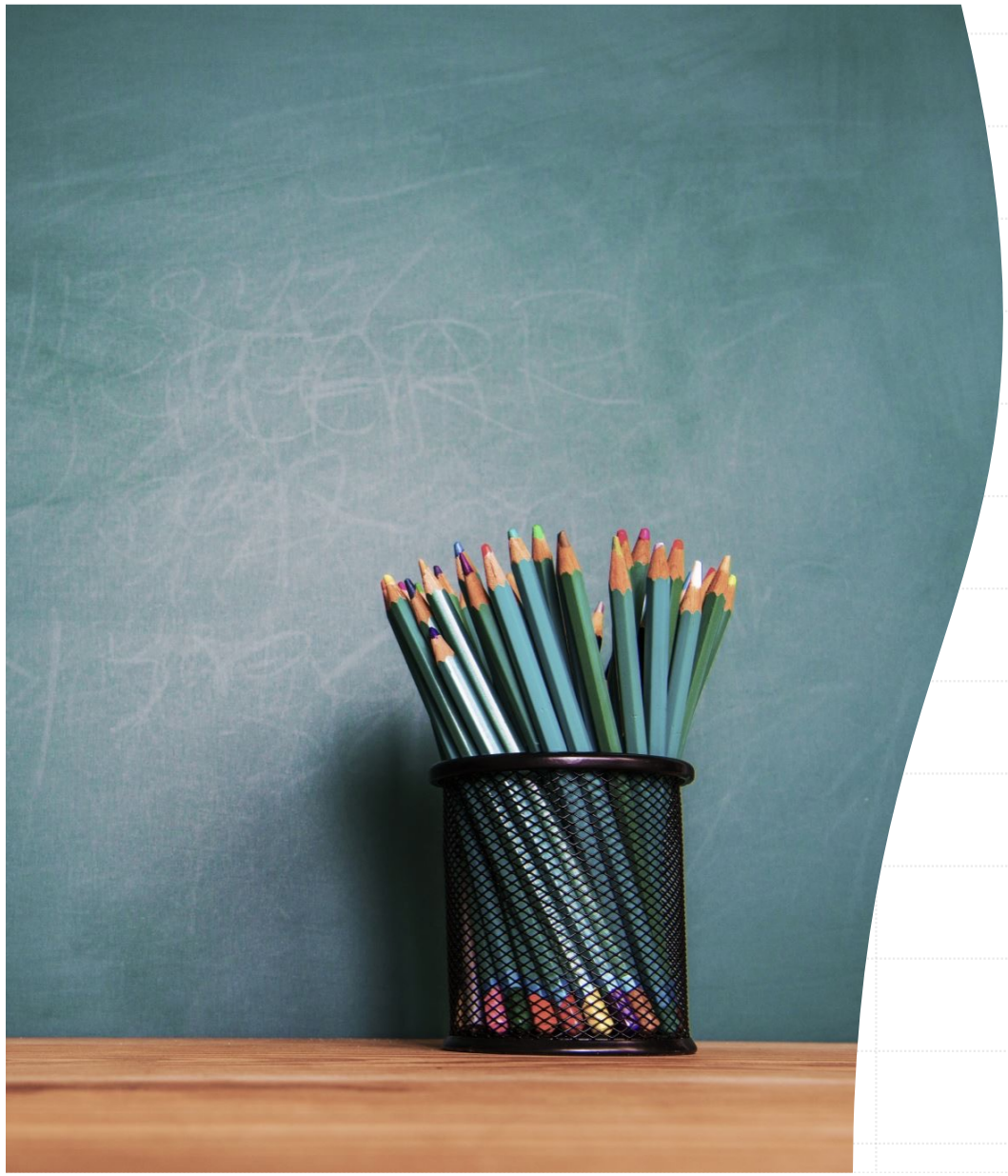


Gherkin & CUCumber (SE-Lab)

Prepared by

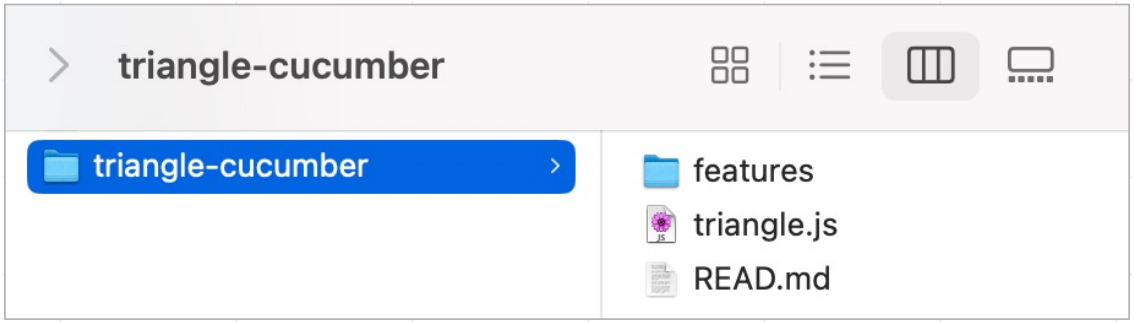
Pittipol Kantavat and

Taratip Suwannasart

Step-by-step Instructions

1. Download “triangle-cucumber.zip” -> https://www.mycourseville.com/sites/all/modules/courseville/files/ckfinder/userfiles/10161706464790065/files/triangle-cucumber_68abeb89bf89d.zip

2. Unzip “triangle-cucumber.zip”



3. cd triangle-cucumber and follow the **Setup Instructions** in the “READ.md”

triangle.js

JS triangle.js ×

JS triangle.js > ...

```
1  function triangleType(a, b, c) {
2    if (a <= 0 || b <= 0 || c <= 0) return 'Invalid';
3    if (a + b <= c || a + c <= b || b + c <= a) return 'Invalid';
4    if (a === b && b === c) return 'Equilateral';
5    if (a === b || b === c || a === c) return 'Isosceles';
6    return 'Scalene';
7  }
8  module.exports = triangleType;
9
```

Equilateral = ด้านเท่า
Isosceles = หน้าจั่ว
Scalene = ด้านไม่เท่า

Step Definition file: triangle.steps.js

JS triangle.steps.js ×

features > step_definitions > JS triangle.steps.js > ...

```
1  const { Given, When, Then } = require('@cucumber/cucumber');
2  const { expect } = require('expect');
3  const triangleType = require('../../triangle');
4
5  let a, b, c, result;
6
7  Given('the triangle has sides {int}, {int}, and {int}', (sideA, sideB, sideC) => {
8    a = sideA;
9    b = sideB;
10   c = sideC;
11 });
12
13 When('I check the triangle type', () => {
14   result = triangleType(a, b, c);
15 });
16
17 Then('the result should be {string}', (expectedType) => {
18   expect(result).toBe(expectedType);
19 });
20
```

Feature file: triangle.feature

triangle.feature

features > triangle.feature

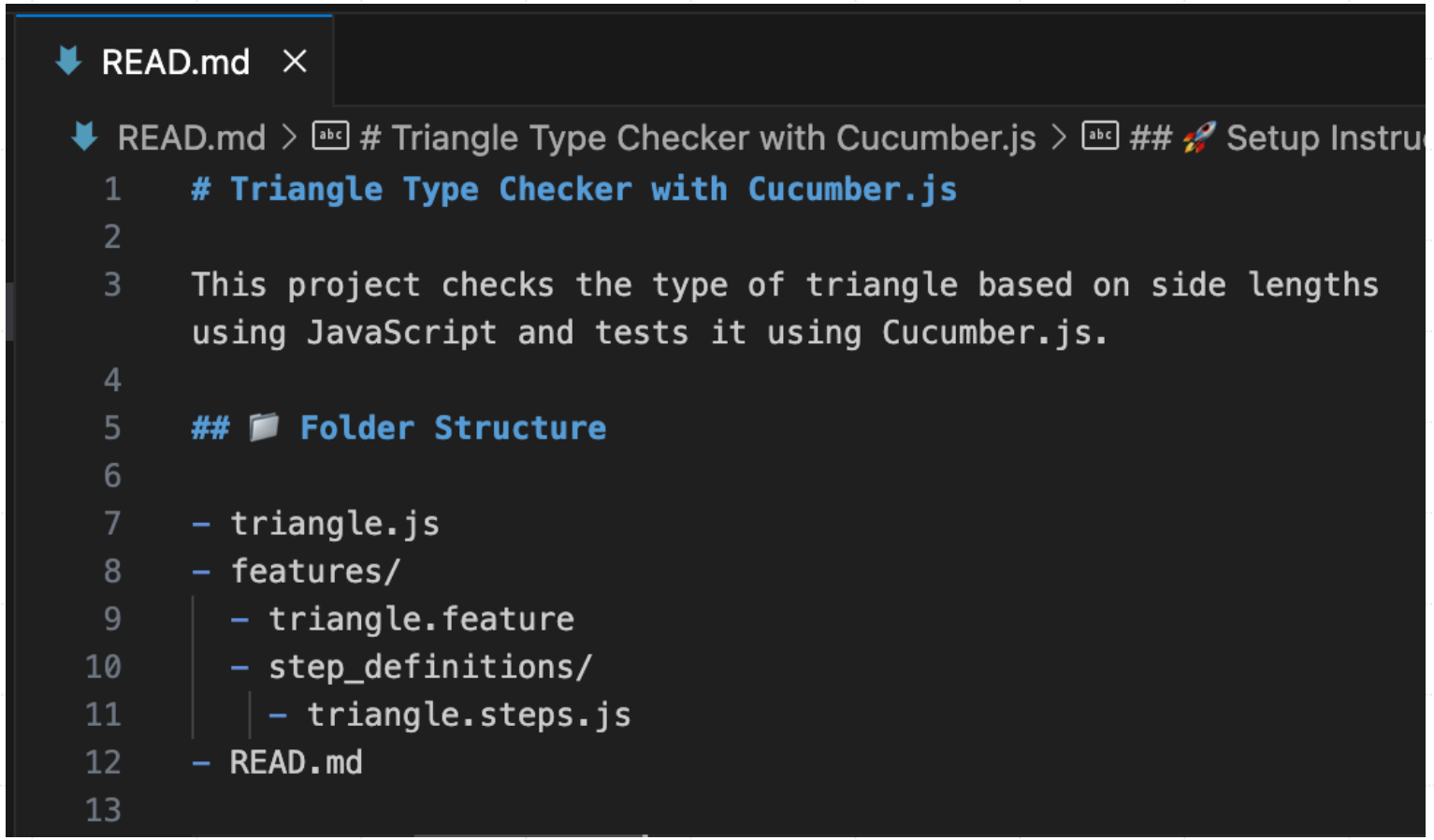
```
Feature: Triangle Type Checker

  Scenario Outline: Determine triangle type
    Given the triangle has sides <a>, <b>, and <c>
    When I check the triangle type
    Then the result should be "<type>"

    Examples:
      | a  | b | c | type       |
      |  3 | 3 | 3 | Equilateral |
      |  5 | 5 | 3 | Isosceles   |
      |  4 | 5 | 6 | Scalene     |
      |  1 | 2 | 3 | Invalid     |
      |  0 | 5 | 5 | Invalid     |
      | -1 | 2 | 2 | Invalid     |

```

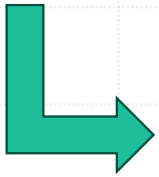
READ.md (1)


READ.md
READ.md > # Triangle Type Checker with Cucumber.js > ## 🚀 Setup Instru
1 # Triangle Type Checker with Cucumber.js
2
3 This project checks the type of triangle based on side lengths using JavaScript and tests it using Cucumber.js.
4
5 ## 📁 Folder Structure
6
7 - triangle.js
8 - features/
9 - triangle.feature
10 - step_definitions/
11 - triangle.steps.js
12 - READ.md
13

READ.md (2)

```
## 🚀 Setup Instructions

1. Initialize project and install dependencies:
   ```bash
   npm init -y
   npm install @cucumber/cucumber expect
   ```
2. Show the results in HTML or JSON
   npm install --save-dev @cucumber/pretty-formatter
   npx cucumber-js --format @cucumber/pretty-formatter
```



```
pittipol@Pittipols-MacBook-Air triangle-cucumber % npx cucumber-js --format @cucumber/pretty-formatter
Feature: Triangle Type Checker # features/triangle.feature:1

  Scenario Outline: Determine triangle type # features/triangle.feature:3
    Given the triangle has sides 3, 3, and 3
    When I check the triangle type
    Then the result should be "Equilateral"

  Scenario Outline: Determine triangle type # features/triangle.feature:3
    Given the triangle has sides 5, 5, and 3
    When I check the triangle type
    Then the result should be "Isosceles"

  Scenario Outline: Determine triangle type # features/triangle.feature:3
    Given the triangle has sides 4, 5, and 6
    When I check the triangle type
    Then the result should be "Scalene"

  Scenario Outline: Determine triangle type # features/triangle.feature:3
    Given the triangle has sides 1, 2, and 3
    When I check the triangle type
    Then the result should be "Invalid"

  Scenario Outline: Determine triangle type # features/triangle.feature:3
    Given the triangle has sides 0, 5, and 5
    When I check the triangle type
    Then the result should be "Invalid"

  Scenario Outline: Determine triangle type # features/triangle.feature:3
    Given the triangle has sides -1, 2, and 2
    When I check the triangle type
    Then the result should be "Invalid"
```

READ.md (3)

```
3. Use Cucumber report
   npm install --save-dev cucumber-html-reporter

4. Create 'generate-report.js'

   const reporter = require('cucumber-html-reporter');

   const options = {
   theme: 'bootstrap',
   jsonFile: 'report.json',
   output: 'report.html',
   reportSuiteAsScenarios: true,
   launchReport: true,
   };

   reporter.generate(options);

5. Run these commands
   npx cucumber-js --format json:report.json
   node generate-report.js
```