Writing Unit Tests Lab (CodeGrade)

[(https://github.com/learn-co-curriculum/js-tdd-writing-unit-tests-lab)](https://github.com/learn-co-curriculum/js-tdd-writing-unit-tests-lab) [(https://github.com/learn-co-curriculum/js-tdd-writing-unit-tests-lab/issues/new)](https://github.com/learn-co-curriculum/js-tdd-writing-unit-tests-lab/issues/new)

Learning Goals

- Follow a test-driven development process for writing code
- Write a unit test using Jest

Introduction

Time for some practice! In this lab, you'll write your own unit test and get it to pass by following a test-driven development workflow.

As before, we'll use Jest as our test framework. We'll be writing the tests in the `src/__tests__/utils.test.js` file and the application code in the `src/utils.js` file.

Fork and clone this lesson, then run `npm install` to install the dependencies.

Instructions

Our word game needs another feature. We'll give users double the amount of points if their word is a **palindrome** (https://en.wikipedia.org/wiki/Palindrome) : a word that reads the same forwards and backwards, like "mom" or "racecar".

You should implement a function `isPalindrome` by following test-driven development that accepts a string as an argument and returns `true` if the string is a palindrome, and false if it isn't.

Try testing some known use-cases first: `isPalindrome("racecar")` should return `true` and `isPalindrome("car")` should return `false` .

Follow the same workflow we did in the previous lesson:

1. Understand the feature we're building
2. Translate the feature into a test specification
3. Write and implement code that passes the test
4. Clean up and refactor
5. Repeat!

Then, consider adding tests for some of these edge cases:

- Should return true for words that are a combination of uppercase and lowercase letters
- Should return false for an empty string
- **Bonus** Should throw an error if input has any non-alphabetic characters (hint: look into the `.toThrow` Jest matcher (https://jestjs.io/docs/expect#tothrowerror)). You can check if a word has only alphabetic characters with this regular expression code: `/^[A-Za-z]+$/.test(word)`
- **Bonus** Should throw an error if input that isn't a string (hint: look into the `.toThrow` Jest matcher (https://jestjs.io/docs/expect#tothrowerror))

Since you'll be *writing* the tests, this lesson doesn't have any tests for you to run to check your implementation. You can see both the completed tests and the working solution in the **solution branch** (https://github.com/learn-co-curriculum/react-hooks-tdd-writing-unit-tests-lab/tree/solution/src) of this repository.

Resources

- **Jest** (https://jestjs.io/)

This tool needs to be loaded in a new browser window

Load Writing Unit Tests Lab (CodeGrade) in a new window