

Lesson

Monday

## Intermediate JavaScript (/intermediate-javascript)

### / Test-Driven Development and Environments with JavaScript

#### (/intermediate-javascript/test-driven-development-and-environments-with-javascript)

#### / Red Green Refactor Workflow

Text

Before we start writing automated tests, let's cover the **Red, Green, Refactor** workflow. We've actually been applying some of the principles of this workflow already. However, we need automated testing in place to really apply it.

### The Red-Green-Refactor WorkFlow

Let's take a look at how it works. We've already been applying most of these steps. However, the steps in **bold** are new.

1. Identify the simplest possible functionality our program must exhibit.
2. Write a coded test.
3. **Before coding, make sure the test fails.**
4. Implement the functionality with the least amount of code possible.

5. Run the automated test to confirm it passes. (If it doesn't pass, revisit step 4.)
6. **Make sure all previous tests still pass.**
7. Check if code can be refactored. If so, refactor and repeat step 6.
8. Commit your code! You should always commit after each passing test.
9. Repeat this entire process with the next simplest piece of functionality.

There are two new steps in the list above. In step 3, we need to confirm that the test fails before we write the code to get it passing. We haven't covered this yet. It's a little more involved than it sounds — there are "good" fails and "bad" fails. We'll cover both soon.

We covered step 6 briefly in Introduction to Programming. However, running all previous tests manually is time-consuming so it wasn't part of our workflow. With automated testing, we'll be able to run all of our tests with a single terminal command. If *any* of our tests don't work (whether it's our most recent test or an older one), we need to isolate the issue and fix it before moving on.

It should be clear why this is called the Red, Green, Refactor workflow:

- **Red** means we need to write a test that fails at first. This needs to be a "good" fail before we move on. We'll learn more about "good" fails soon.
- **"Green"** means that all of our tests need to be passing before we move on.
- **"Refactor"** means that we should always look for opportunities to refactor after each passing test.

In the next lesson, we'll learn how to set up Jest. Then we'll be ready to apply the Red, Green, Refactor workflow and start writing tests.

[Previous \(/intermediate-javascript/test-driven-development-and-environments-with-javascript/tdd-review\)](#)

Next (/intermediate-javascript/test-driven-development-and-environments-with-javascript/setting-up-jest)

Lesson 25 of 49

Last updated more than 3 months ago.

disable dark mode



© 2023 Epicodus (http://www.epicodus.com/), Inc.