Review

We have just covered Test Driven Development with Mocha.

**At a high-level the process is:**

* Write The Test — Start with a test describing the functionality we’d like to see.
* Fail The Test — Write code in response to the test that fails.
* Pass The Test — The tests fail and communicate feedback to developers through error messages. It’s our responsibility to read those messages, then respond by writing the minimum amount of code to address those messages.
* Refactor Your Code — See below.

**The development process is guided by the red-green-refactor cycle.**

**Red**

Write a test that covers the functionality you would like to see implemented. You don’t have to know what your code looks like at this point, you just have to know what it will do.

Run the test. You should see it fail. Most test runners will output red for failure and green for success. While we say “failure” do not take this negatively. It’s a good sign! By seeing the test fail first, we know that once we make it pass, our code works.

**Green**

Read the error message from the failing test, and write as little code as possible to fix the current error message. By only writing enough code to see our test pass, we tend to write less code as a whole. Continue this process until the test passes.

This may involve writing intermediary features covering lower level functionality which require their own Red, Green, Refactor cycle. The **edge-case** was an example of this.

Do not focus on code quality at this point. Be shameless! We simply want to get our new test passing.

**Refactor**

Clean up your code, reducing any duplication you may have introduced. This includes your code as well as your tests.

Treat your test suite with as much respect as you would your live code, as it can quickly become difficult to maintain if not handled with care. You should feel confident enough in the tests you’ve written that you can make your changes without breaking anything.