

Module 8: Testing; Linting

Edgardo Molina, PhD | Head Instructor

CUNY Tech Prep 2017-2018

- Testing
 - Jest
- Linting
 - ESLint

How do we know our code is done
and correct?

How do we validate our code?

- **Formal Reasoning:** A mathematical formal proof that demonstrates the code is correct.
- **Code Review:** Another programmer reviews the code and verifies that it is correct.
- **Testing:** Run the program against a collection of known inputs and expected outputs.

Formal Reasoning

Difficult and costly to do.

Rarely done outside of small *critical* pieces of software.

Think interpreters, OS schedulers, NASA code, etc.

Code Review

Commonly done in industry before code is merged into the product.

Code reviews are about correctness, but *also* about clarity and style.

Correct code can be rejected for being unclear, messy, inefficient, or any other reasoning established by the organization.

Testing

Testing helps us ensure the code is bug free today and in the future

Writing tests for our code:

- ensures the code meets the specifications
- ensures future changes/features don't break pieces that already work
- allows new developers to understand expected behavior of the code

Testing

Three primary types of tests

- **Unit Test:** tests a single unit of code. This can be a class or function. Unit tests should not depend on any other code outside of what is being tested.
- **Integration Test:** tests how multiple units work together. Ensures that their integration continues to produce expected output.
- **System Test:** tests the input and output of the entire system. Simulates actual user input, actions, and workflows into the test and verifies the user receives expected output.

Jest is an *increasingly* popular testing framework for JavaScript.

Developed at Facebook. Doesn't require configuration. Can be used for React and Express apps.

<https://facebook.github.io/jest/>

Linting

Linters are static code analyzers. They:

- verify that code is written in a *consistent* style
- notify you of problematic (*ambiguous/confusing*) coding patterns
- provides some basic type analysis for dynamic interpreted languages

ESLint is one of the commonly used linters for JavaScript

It is a pluggable linter, so that rules can be customized. It also provides 3 base references to start from. We will use the 'AirBnB' style.

JavaScript ESLint (<http://eslint.org/>)

Google JavaScript Style Guide

(<https://google.github.io/styleguide/javascriptguide.xml>)

AirBnB JavaScript Style Guide (<https://github.com/airbnb/javascript>)

Installing ESLint

Install ESLint into your project

```
npm install eslint --save-dev
```

Set it up the first time (Setup is interactive, you can (Select Popular Style -> AirBnB -> JSON)

```
./node_modules/.bin/eslint --init
```

You can then run the linter on the entire project or on specific files

```
./node_modules/.bin/eslint .
```

```
./node_modules/.bin/eslint controllers/authors.js
```

Text Editors have linter plugins

These run as you code and notify you of problems