

Intro to TDD

(Test Driven Development)

What is TDD?

- Test Driven Development
- A cyclical process:
 - write a test
 - watch test fail (why?)
 - write smallest amount of code to make test pass
 - refactor

From this morning, you should have a basic understanding of...

- classes
- objects
- methods and arguments
- string interpolation

By the end of today, you'll be able to...

- install a Ruby gem
- create and run a minitest test suite
- write assertions using minitest
- read error messages
- fix error messages
- read test failures
- fix failing assertions
- explain and demonstrate the TDD workflow

Why write tests for your code?

- Fast feedback about whether your code works
- Encourages “clean” code
- Makes code easier to refactor
- Your job

Today's Process

- Test something all together (1pm)
 - ask lots of questions!
- Test something in pairs (2pm)
 - from tutorial
 - will mimic what we do all together (only one new part)
- If you finish, test something on own (3pm)
 - either repeat tutorial on own or create own test suite for own object

Today's Pairs

Glen Egbert
Jessica Goulding

Cara Dennis
Kavita Sachdeva

Jim Sutton
Aaron Paul Wortham

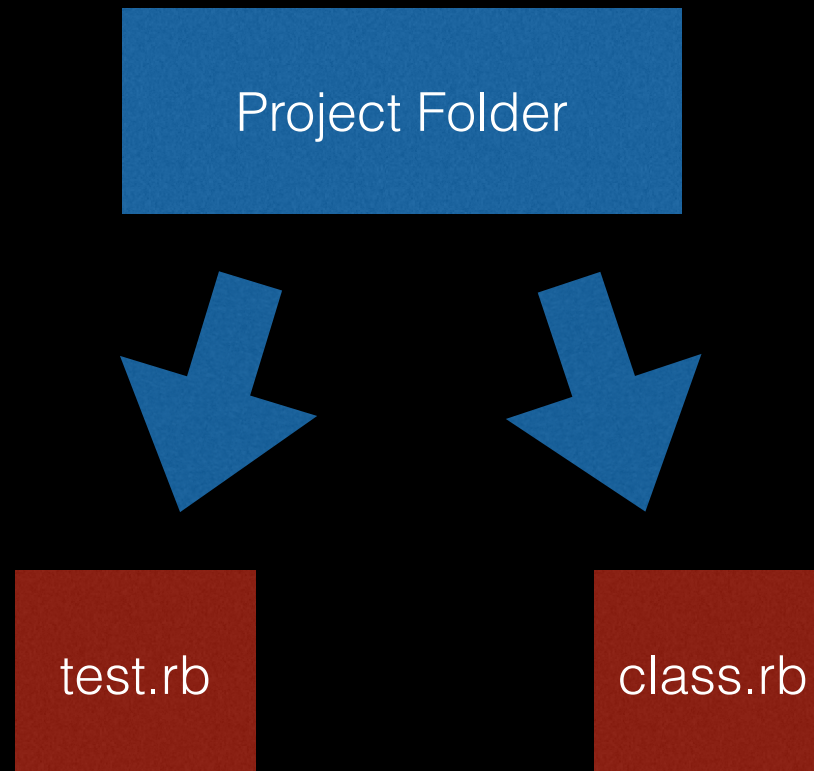
Luke Aiken
Sara Simon

Rolando Navarrete
Ian Anderson

What we're going to test

- We'll be implementing a Student class using TDD. Here's what we want to test:
 - Students should have names
 - Students should have laptops. The laptop is usually an Apple, but it can be any brand
 - Students can bring various flavors of cookies to their instructors

Basic File Structure for Testing



Let's go!

From Terminal, type:

```
gem install minitest
```

(don't let me forget to show pairing list!)