WHAT IS TESTING?
WHAT IS TESTING?

Tests are a set of assertions against the behavior of code that a developer has written.
WHY TESTING?

• “It runs on my machine!”
• Agile!
• Machines > Humans
• Why code comments?
• Why documentation?
• Code is read more often than it is written
WHY TESTING?

• Helps you verify that your code does what you think it does
• Helps you refactor/improve code
• Helps your contributors understand the intent of your code
• Helps unintentional failure in the future
• Dynamic Languages: Assert “implicit types”
TDD

• Red - start a test you know currently fails
• Green - modify your code to pass
• Refactor - change so that it is good code
• Coverage - percentage of code that is touched by tests
UNIT TESTING

- The smallest independent unit of code that does something
- Assert various edge cases
FIZZ BUZZ
FIZZ BUZZ

• Function with 1 input
• print input in the default case
• print ‘fizz’ when input is a multiple of 3
• print ‘buzz’ when input is a multiple of 5
• print ‘fizzbuzz’ when input is a multiple of 3 and 5
DEPENDENCIES

-Mocks
-Double
-Stub
INTEGRATION TESTS

- Setup individual components, and see that they work together
ACCEPTANCE TESTS

• Simulate user behavior as much as possible
• Other tests assume good test setup, here little assumptions
• Last defense layer
• Fragile
TEST LAYERS

- Unit Tests
- Integration Tests
- Acceptance Tests
- Manual Testing

Cheap
Easy
Reliable

Closer to reality
Expensive
Fragile
TO THE REAL WORLD!
CONTINUOUS INTEGRATION

• Running tests as much as possible
• Server that executes your tests against all your commits/branches
• git bisect to find offending commit
THANK YOU!