**Testing (in Ruby..)**

1. Error Drive Development
   1. Try the thing you want, get an error, fix the error, repeat
2. Build automated tests to check your code... instead of human inputs...
   1. Browser, System, Integration Testing - does the feature work as intended?
   2. Unit Testing - the smallest piece of code
   3. Error Testing - validate errors are triggering and denying vulnerable data
3. Tests are often based on a ‘spec’ or ‘specifications’
   1. A list of requirements of a feature that need to be turned into a test
   2. In a company, it may be the developer or the QA section that write the tests
4. Sensitivity vs Specificity
   1. Sensitivity -
   2. Specificity -
5. Test Driven Development - write the tests first, then develop the code
   1. Red - Write the failing test
   2. Green - Write the simplest code to get the test to pass
   3. Refactor - Clean up the code for the best implementation
6. Install rspec gem in folder with the original code...
   1. Make a dir called spec
   2. Make a file named originalFile\_spec.rb
      1. Example: If the original file is ‘testFile’, the rspec should be ‘./spec/testFile\_spec.rb’
   3. Run rspec in original folder to verify no tests exist
   4. Write tests / code / run tests
      1. Ex of production code:



* + 1. Ex of test code:



* 1. Run Final Tests (Terminal: rspec -fd)

1. Common JavaScript Testing Libraries:
   1. Jest
   2. Jasmine
   3. ~~Mocha~~
   4. Tape
2. Common Ruby Testing Libraries:
   1. Rspec
   2. MiniTest
3. Common Python Testing Libraries:
   1. PyTest - no need for anything else!