



# RUBY

WEEK 7

# MID-TERM REVIEW

- Git issues: take some time to fix now.
- Questions??

# REVIEW

- Git
- Objects
- RSpec
- Complex Data

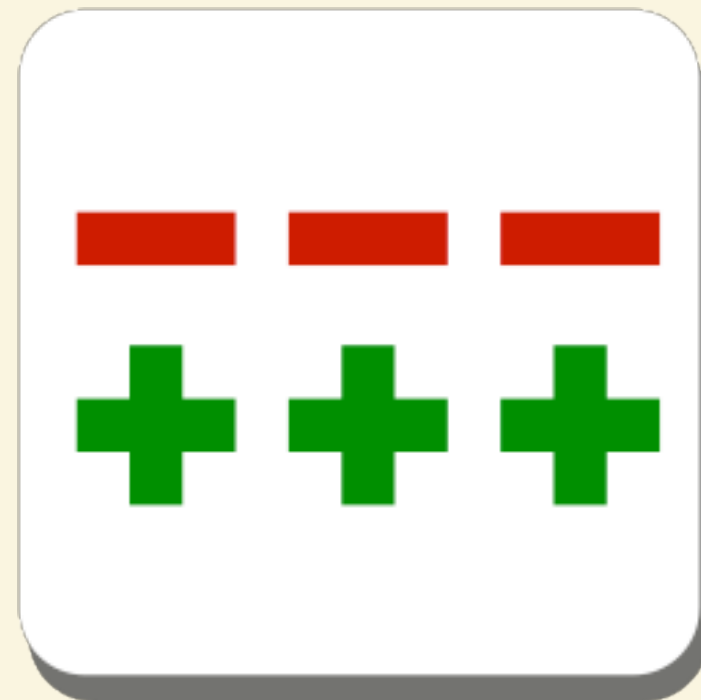


A red pen is positioned at the top left of a rating scale. The scale is titled "RATING" and has five rows. The first row is checked with a red checkmark and labeled "Exceptional". The second row is labeled "Exceeds requirements". The third row is labeled "Meets requirements". The fourth row is labeled "Needs improvement". The fifth row is labeled "Poor".

RATING	
<input checked="" type="checkbox"/>	Exceptional
<input type="checkbox"/>	Exceeds requirements
<input type="checkbox"/>	Meets requirements
<input type="checkbox"/>	Needs improvement
<input type="checkbox"/>	Poor

# GIT

- `git init`
- `git add`
- `git commit`
- `git log`
- `.gitignore`



# GITHUB / GIT REMOTES

- Fork
- Pull Request
- `git remote`
- `git clone`
- `git branch (grb)`



# OBJECTS

- new / initialize
- method signatures
- relationships



# RSpec

- describe
- it
- matchers
- Relationship between our code and RSpec



# TESTING AGENDA

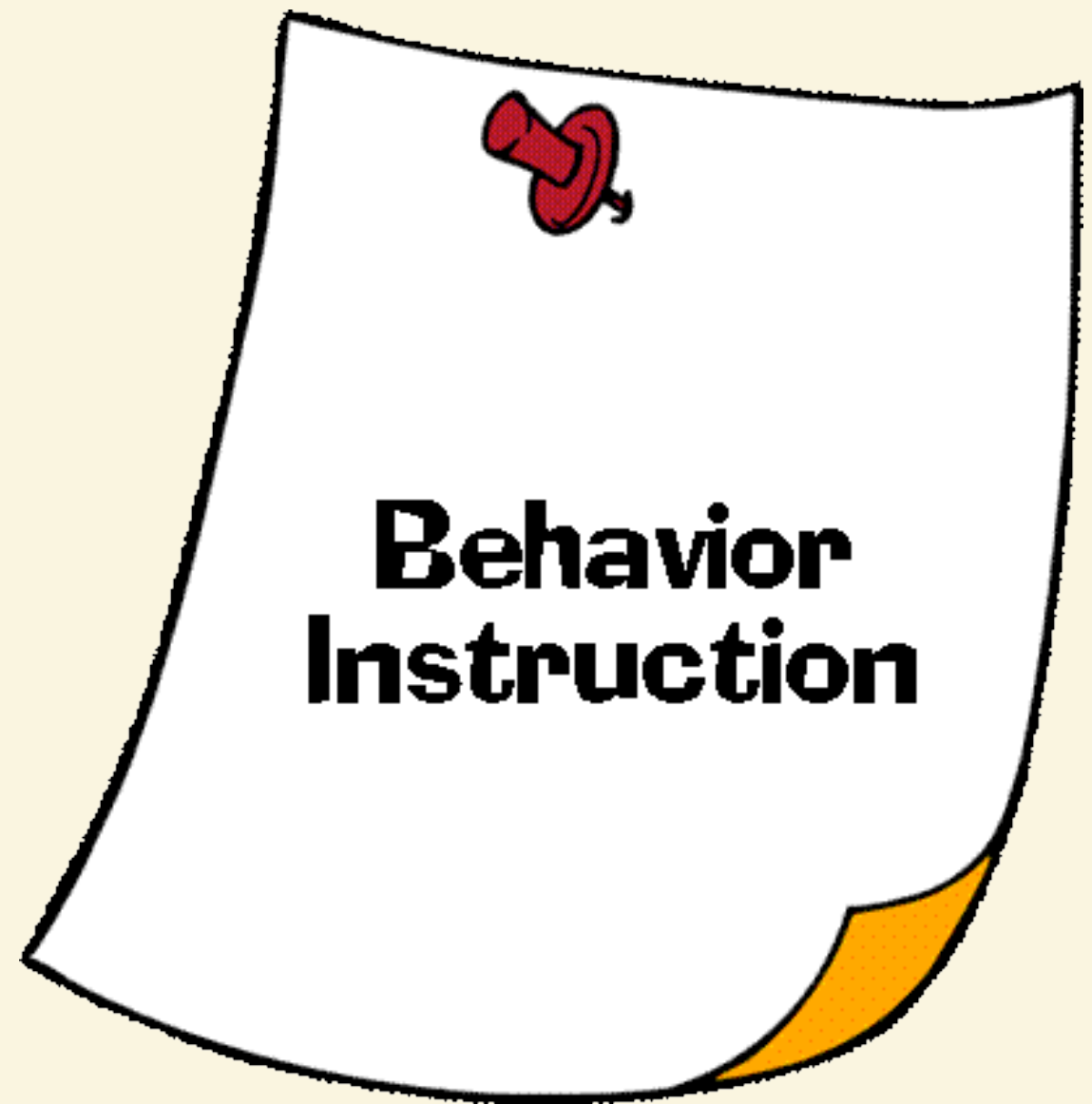
- Behavior Driven Development with Cucumber
- Fun with MiniTest
- TestUnit and Cucumber with Jeweler and Travis
- Why Testing? With Nell!





# BEHAVIOR DRIVEN DEVELOPMENT

- User Stories
- Business Value
- Communication
- Shared Tools



# BEHAVIOR SPECS

- Title
- Narrative
- Acceptance Criteria / Scenarios / Outcomes



# *Cucumber*

*behaviour driven development  
with elegance and joy*

- `gem install cucumber`
- `cucumber --help`



# FEATURES

- Human Read/Writable description of behavior
- Business Users write the features, Programmers develop the features to spec
- <https://github.com/cucumber/cucumber/wiki/Feature-Introduction>

# GIVEN-WHEN-THEN

- Describing your feature
- Preconditions
- Something happens
- Expected outcome

# STEP DEFINITIONS

- REGEXP!!
- Translation between the human narrative and the code
- This type of sentence, matches to this type of code

# DEMO!


- I want a Celsius to Fahrenheit converter
- I tell my story
- I copy the generated regex
- I write my tests
- TDD!

# EXAMPLE!

- New Feature!
- I'm in London and I can't share bar stories without being able to convert Fahrenheit to Celsius
- Write the Steps
- Make Them Pass!



# MINITEST

- Built into Ruby 1.9!
- SeattleRB! (meets Tuesday night  )
- Light, Fast, FUN!

# EXAMPLE

- Make a ruby file
- Add some tests
- Make them pass
- All Ruby!

# TEST UNIT

- Jeweler!
- Simple testing framework
- Used lots of places!

# CUCUMBER WITH JEWELER AND TRAVIS

- EXAMPLE!
- Add cucumber to Gemfile
- Run bundler
- Add feature, steps, and support (env.rb)
- Add to travis.yml file

# HOMework

- Read Chapter 23 and 24 DuckTyping and MetaProgramming
- Make the cucumber features pass