



# Ruby

---

Week 5



# Homework Review

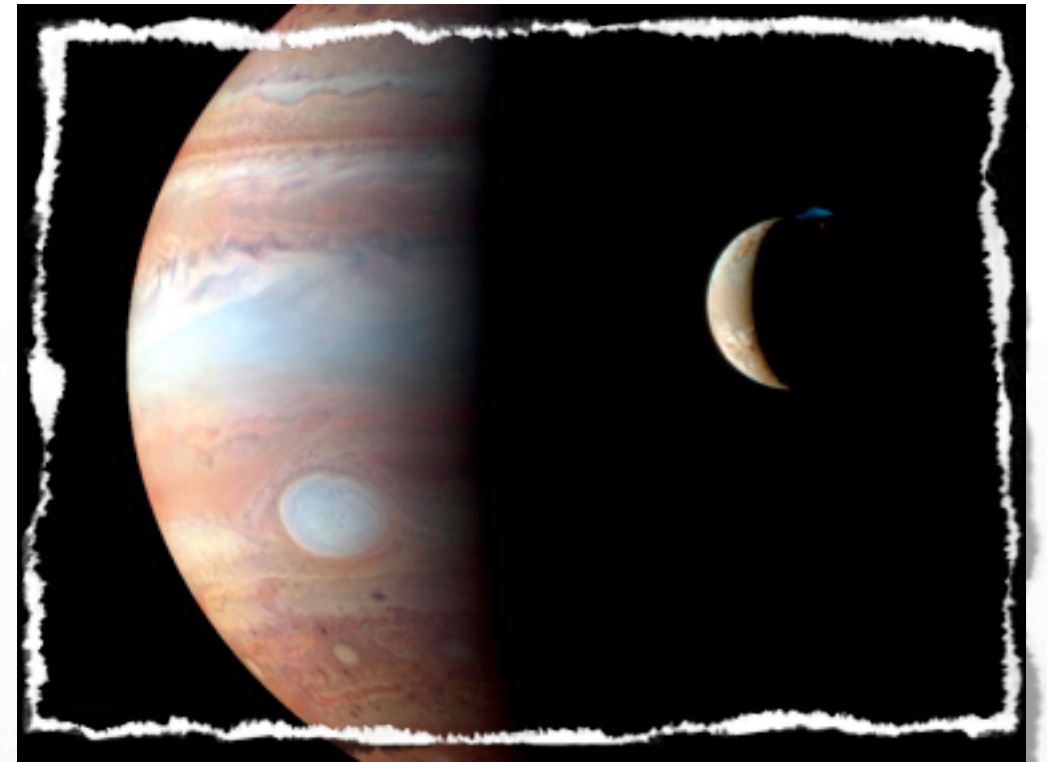
- Worker Spec





# IO

- Input and Output
- Reading in, Writing out
- Working with the external (real!) world





# Files

- `file = File.open("test_file", "r+")`
- `"r"` Read-only, starts at beginning of file (default)
- `"r+"` Read-write, starts at beginning of file
- `"w"` Write-only, truncates existing file to zero length or creates a new file for writing
- `"w+"` Read-write, truncates existing file to zero length or creates a new file for reading and writing





# Files with Blocks

- `File.open("test_file", "r+") do |r|`
  - `r << "Add some text"`
- `end`
- `f = File.open("roster.txt")`
- `f.each { |line| puts line }`
- `f.map { |line| line.split(',').first }`



# Files

- `f = File.open("roster.txt")`
- `f.gets` (get a line)
- `f.pos` (character position)
- `f.rewind` (`pos = 0`)
- `f.read` (file as a string)
- `f.readlines` (array of each line as a string)





# Directories

- `Dir.pwd`
- `Dir.chdir("..")`
- All files: `Dir["*"]` All hidden files: `Dir[".*"]`
- `d = Dir.new(".")`
- `Dir.mkdir("test")`
- `d.entries`
- `d.count`
- `d.each{|f| puts f}`



# Rakefile

- Ruby Make
- Ruby scripting
- Ordered Tasks (configuration!)





# Rake tasks

- `task :name => [:dependency] do`
- `<ruby code>`
- `end`



# Exercise

- Create a task that reads all the lines in names and outputs them
- Create a task that creates a class directory
- Create a task dependent on the class directory task that makes a directory in the class directory for each name in names





# RegExp!

- **Regular Expressions in Ruby**



# Patterns

- **Find a pattern in text**





# Use

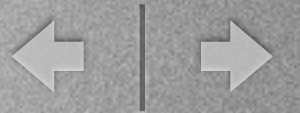
- Test a String
- Change a String



# /find me/ - Testing

- ✧ “Hello World” =~ /World/
- ✧ pattern = /World/
- ✧ pattern.match “Hello World”





# sub / gsub - Changing

- `sub(/pattern/,"text")` - replace the first
- `gsub(/pattern/,"text")` - replace all



# Rubular

<http://www.rubular.com/>





# TitleCase, snake\_case

- ❖ “this is supposed to be a title”.gsub(/\b\w/) { |w| w.upcase}
- ❖ “this should be a snake case”.gsub(/\b\W/){ |b| ‘\_’}



# Homework

- Do the midterm!
- Ask me questions!





# Review

- Git
- Objects
- RSpec
- Complex Data

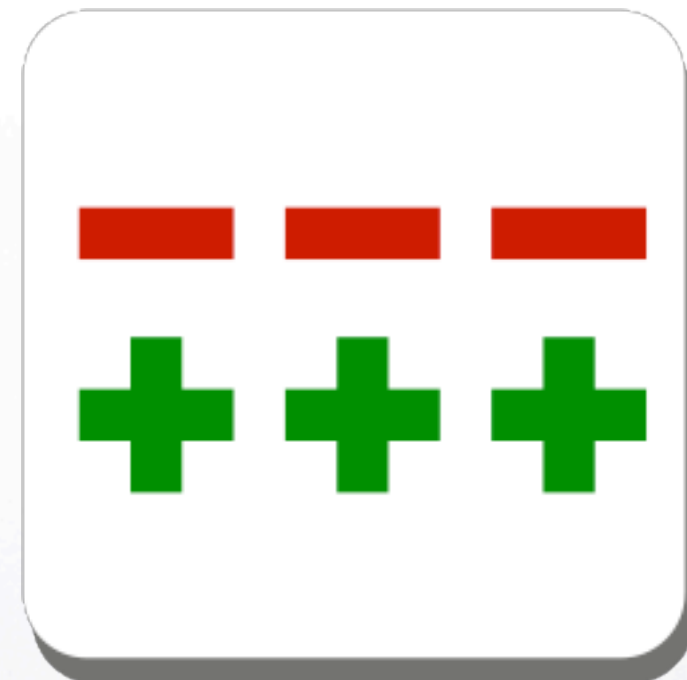
A rating form titled "RATING" is shown. It has a red pen resting on it. The form contains a checklist with five items, each with a checkbox and a description. The first item, "Exceptional", is checked with a red checkmark. The other items are "Exceeds requirements", "Meets requirements", "Needs improvement", and "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

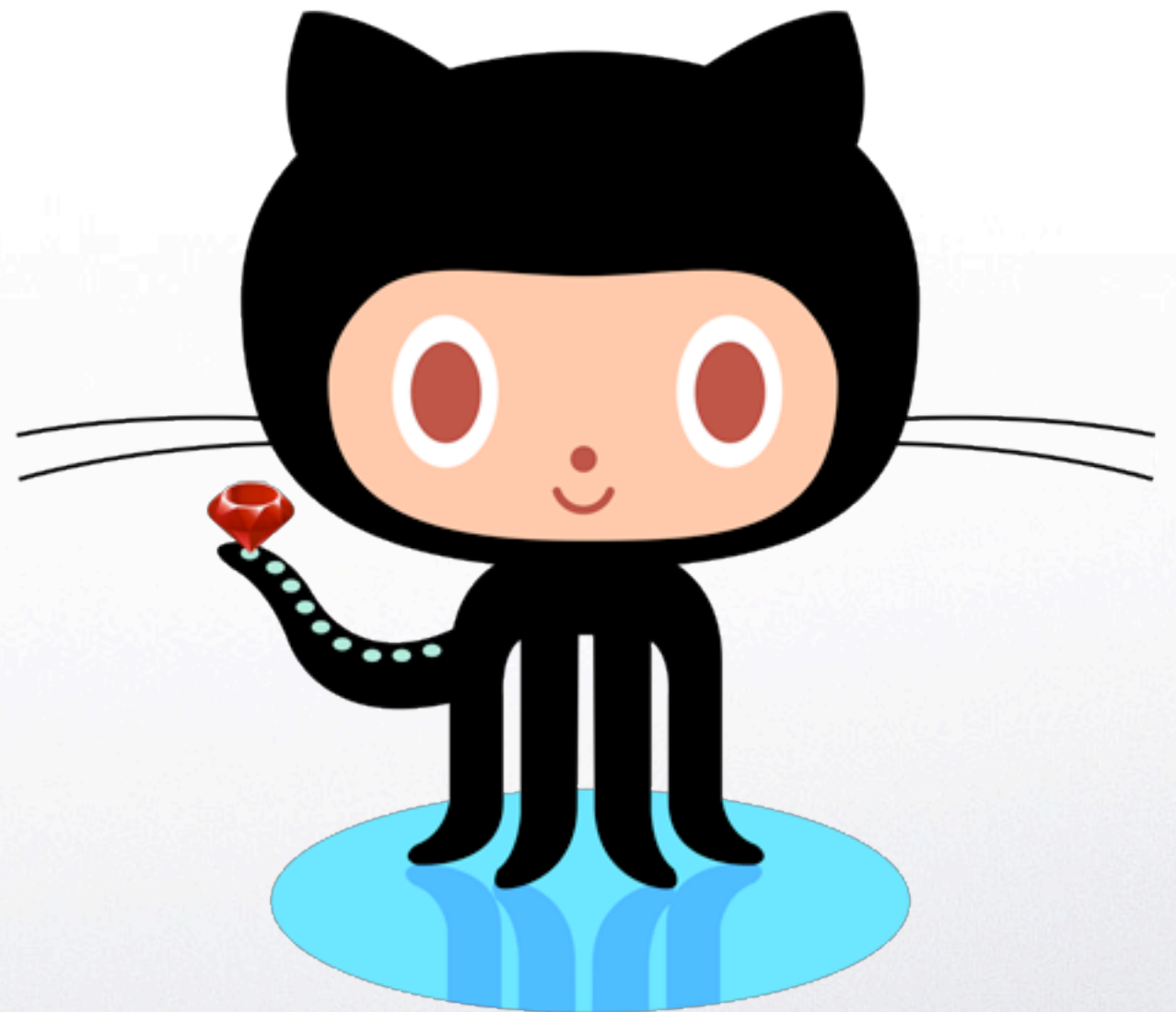


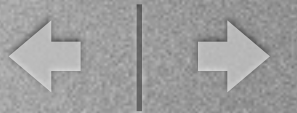




# Github / Git Remotes

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



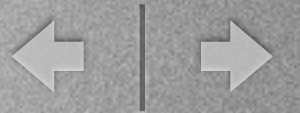


# Objects

- new / initialize
- method signatures
- relationships







# RSpec

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

