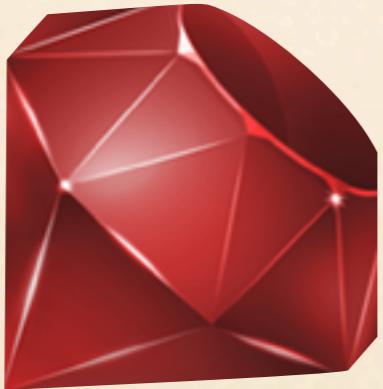


# NIRD

## RUBY ON RAILS

---

2 DAY BootCAMP





# RUBY 1 DAY

---

THE CORE LANGUAGE

# HELLO!

WELCOME!

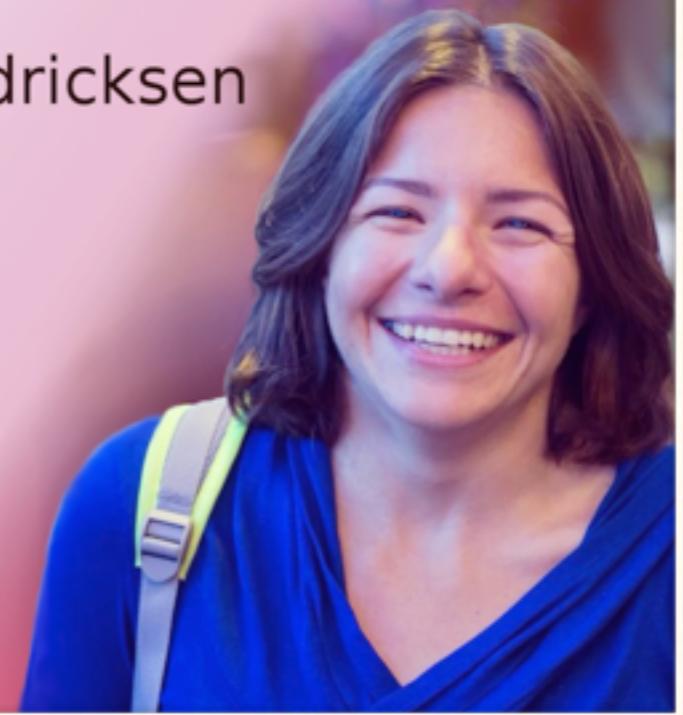
Renée Hendrickson

✉ renée@nird.us

🐦 @nirdllc

𝑺𝒐cial reneedv

[nird.us](http://nird.us)



# NIRD

# INTRODUCTIONS

- ❖ Name
- ❖ Coding Experience
- ❖ What would you like to learn?



# CURRICULUM

## RUBY I DAY

- ❖ Ruby
- ❖ Core-Syntax
- ❖ Standard Libraries
- ❖ Common Patterns and Best Practices

## RAILS I DAY

- ❖ RoR Toolset
- ❖ Rails Walk-Through
- ❖ Gems and Deployment

# SCHEDULE

Times	Day 1 Ruby	Day 2 Rails
9am - 10:30	Intro and Rspec	Tools and Generators
Break		
10:45 - 12	Basic Objects and Collections	Routing and Controllers
Lunch		
1pm - 2:30	RegExp and Files	Active Record
Break		
2:45 - 4:30	Blocks and Method Missing	Gems

# LEARNING GOALS

- ❖ Basic Ruby
- ❖ Common Patterns and Practices
- ❖ Where to go for help

# LEARNING PATH

- ❖ Experimentation
- ❖ Test First Teaching
- ❖ Ask Questions
- ❖ Resources for finding answers

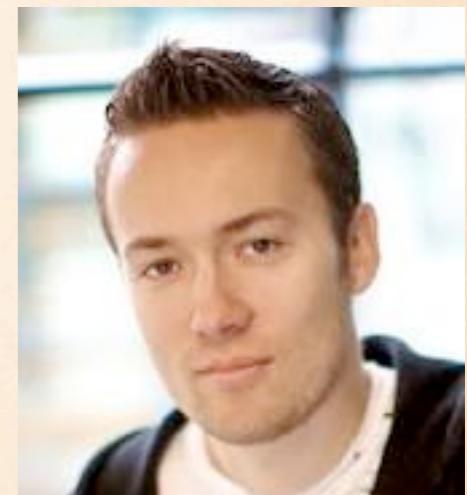
# STRUCTURE

- ❖ Explanation (me talking)
- ❖ Demonstration (me typing)
- ❖ Experimentation (in-class coding)
- ❖ Questions



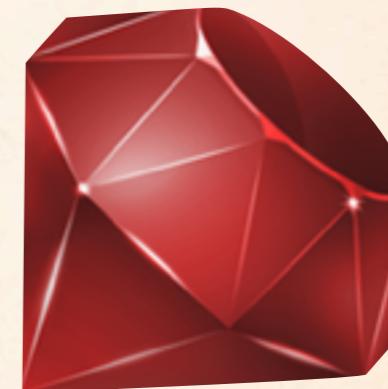
# RUBY VS. RAILS

- ❖ Ruby is the Language -
  - ❖ developed 1993
  - ❖ Yukihiko Matsumoto “Matz”
  
- ❖ Rails is the Framework -
  - ❖ developed 2004
  - ❖ David Heinemeier Hansson “DHH”



# THE RUBY WAY

- ❖ Written in C
- ❖ Influenced by SmallTalk & Pearl
- ❖ Ruby is for People
- ❖ MINSWAN
- ❖ OSS (Open Source Software)
- ❖ RubyGems

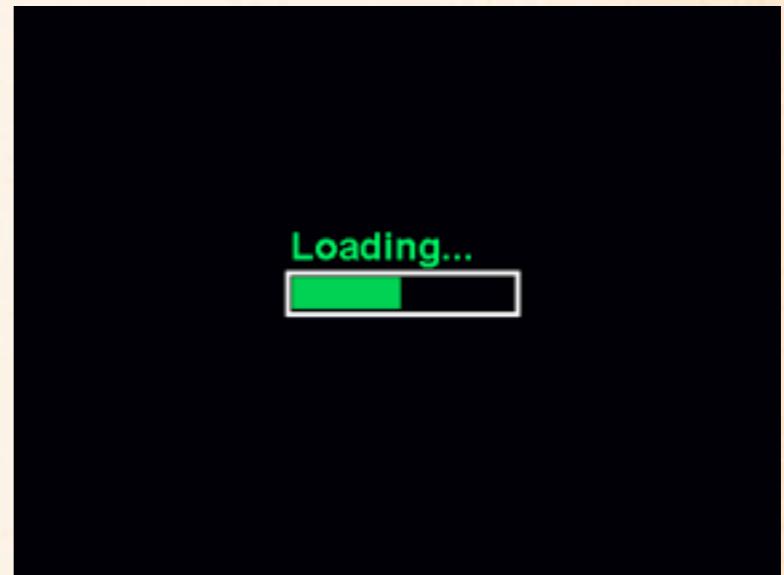


# THE RUBY LANGUAGE

- ❖ Dynamically Typed
- ❖ Interpreted (Standard Lib)
- ❖ Runtime Modifications
- ❖ Object Oriented
- ❖ Blocks and Lambdas
- ❖ Regular Expressions

# INSTALL CHECK

- ❖ Command Prompt
- ❖ Git
- ❖ Ruby
- ❖ Rspec
- ❖ Editor (Syntax Highlighting)



# COMMAND LINE

- ❖ Command Prompt (CLI)
  - ❖ Mac: Terminal, iTerm, etc...
  - ❖ Unix: bash, zsh, csh, etc....
  - ❖ Windows: Command Prompt w/ Ruby on Rails

# EDITOR

- ❖ Sublime Text 2 / TextMate
- ❖ RubyMine
- ❖ KomodoEdit
- ❖ Eclipse
- ❖ Vi / Vim / Emacs

# SUBLIME TEXT 2

- ❖ Command line alias for opening a file or folder:

- ❖ Win: doskey subl="C:\Program Files\Sublime Text 2\sublime\_text.exe" \$\*
- ❖ Mac: ln -s "/Applications/Sublime Text 2.app/Contents/SharedSupport/bin/subl" ~/bin/subl
- ❖ Linux: sudo ln -s /opt/SublimeText2/sublime\_text /usr/bin/subl

Command line alias

Win: doskey subl="C:\Program Files\Sublime Text 2\sublime\_text.exe" \$\*

Mac: ln -s "/Applications/Sublime Text 2.app/Contents/SharedSupport/bin/subl" ~/bin/subl

Linux: sudo ln -s /opt/SublimeText2/sublime\_text /usr/bin/subl

How to Open a folder

# GIT

- ❖ which git
  - ❖ ex: /usr/local/bin/git
- ❖ git --version
  - ❖ ex: git version 1.7.10.2

# RUBY

- ❖ Ruby -v
  - ❖ 1.9.2, 1.9.3, 2.0.0 (1.8.7)
- ❖ gem list
- ❖ Bundler, Rake, rubygems, etc...

# RVM

## RUBY VERSION MANAGER

- ❖ rvm list
  - ❖ rvm use
  - ❖ rvm gemset list
- 
- ❖ rvm.io



# RUBY VERSIONS

- ❖ MRI, YARV, Standard Library, C-Ruby
- ❖ JRuby - Java
- ❖ Rubinius - Pure Ruby
- ❖ MagLev - SmallTalk
- ❖ MacRuby - Objective-C
- ❖ IronRuby - C# (for .Net)

# RUBY INTERPRETER

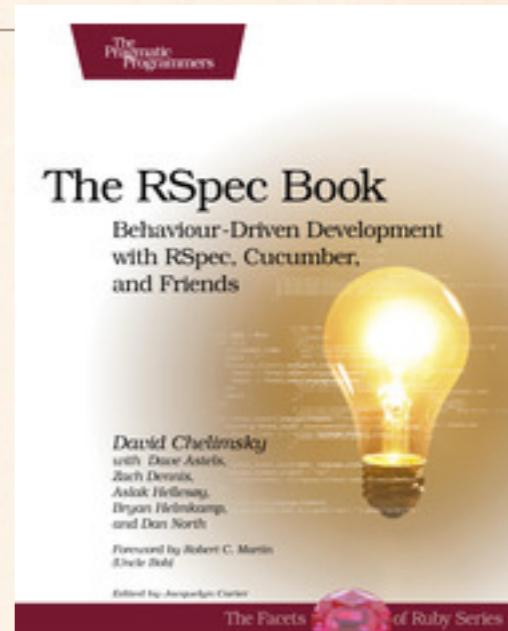
- ❖ irb
- ❖ ex:
- ❖ 1.9.3-p194 :001 > 1+2
- ❖ => 3
- ❖ 1.9.3-p194 :002 > exit

# RSPEC

- ❖ gem install rspec
  - ❖ ex: Fetching: rspec-core-2.11.1.gem (100%)
- ❖ gem list
  - ❖ ex:
    - ❖ diff-lcs (1.1.3), rspec (2.11.0), rspec-core (2.11.1), rspec-expectations (2.11.3), rspec-mocks (2.11.3)
- ❖ rspec -v
  - ❖ ex:
    - ❖ 2.11.1

# RSPEC INTRO

- ❖ Code that Tests code!
- ❖ Test Driven Development
- ❖ Red-Green-Refactor
- ❖ Test First Teaching
- ❖ Ruby Domain Specific Language (DSL) for testing



# RSPEC SPECIFICS

- ❖ Describe (What) / Context (When)
- ❖ It (Does)
- ❖ Should
- ❖ Matchers
- ❖ <http://rubydoc.info/gems/rspec-expectations/2.4.0/RSpec/Matchers>

# RSPEC DEMO!

- ❖ make a new rspec file
- ❖ run rspec
- ❖ walk through rspec\_spec.rb

rspec rspec\_spec.rb

# RSPEC FORMATTING

- ❖ Mac/Linux: alias rspec="rspec --color --format nested"
- ❖ Win: doskey rspec="rspec --color --format nested" \$\*

# IN-CLASS EXERCISE

- ❖ Run exercises/rspec\_spec.rb
- ❖ Fix the broken in-class test
- ❖ Add the 3 pending in-class tests (don't forget the **do/end** and **should!**)

# BREAK!

---

10-15MIN

# RSPEC REVIEW

- ❖ Code that Tests Code!!
- ❖ Describe a feature, set an expectation.
- ❖ Red-Green-Refactor

# RUBY

- ❖ Everything is an Object!
- ❖ Type Doesn't matter! (`respond_to?`)
- ❖ Speaking Ruby : Interpreter Needed!
- ❖ MRI, YARV, JRuby

# RUBY GEMS

- ❖ Packaged Ruby Code
- ❖ Open Source
- ❖ iPhone user: “There’s an app for that!”
- ❖ Ruby programer: “There’s a **gem** for that!”

# IRB

- ❖ Interactive Ruby!
- ❖ Our Ruby **Interpreter!**
- ❖ We speak ruby, it translates!
- ❖ Understand Ruby code, evaluates it, gives us the result  
**(inspect)**

# RUBY

- ❖ Everything evaluates to something
- ❖ `nil` is NULL or nothing or Not Set
- ❖ Every Object `respond_to?` `inspect`
  - ❖ `=> true`

# OBJECTS

- ❖ Ruby: everything is an object
- ❖ Everything is an instance of a class
- ❖ Ruby knows about certain classes of things: Standard Library

# STRING

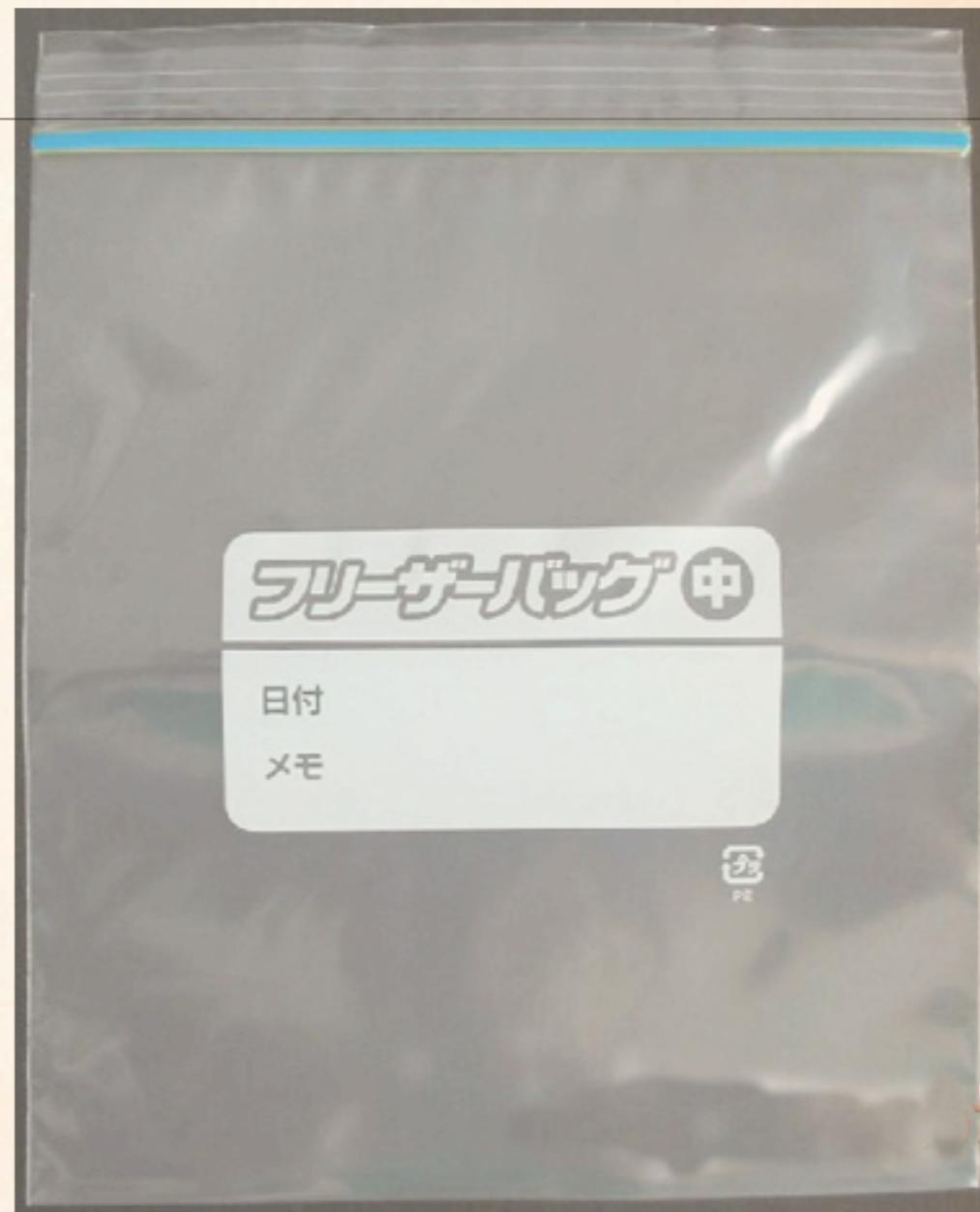
- ❖ String class : part of the standard library
- ❖ How Ruby understands text
- ❖ Instances
  - ❖ `String.new` or `"` or `""` or `%Q{}` or `%q{}`

# STRING CLASS

- ❖ <http://www.ruby-doc.org/core-1.9.3/String.html>
- ❖ Methods : Messages we can send, what an instance will respond to
- ❖ Chomp, \*, +, include?, upcase

# VARIABLES

- ❖ Named location
- ❖ Points to something
- ❖ Contents Can Change



# STRING INTERPOLATION

- ❖ my\_name = “Renée”
- ❖ “Hello #{my\_name}”

# PUTS / GETS

- ❖ put string
- ❖ get string
- ❖ puts “Hello World”
- ❖ user\_input = gets
- ❖ user\_input = gets.chomp

# NUMBERS

- ❖ FixNum
- ❖ I is always I! (unlike strings)
- ❖ Floats (BigDecimal)
- ❖ 4.2
- ❖ 3.141516



# RANGES

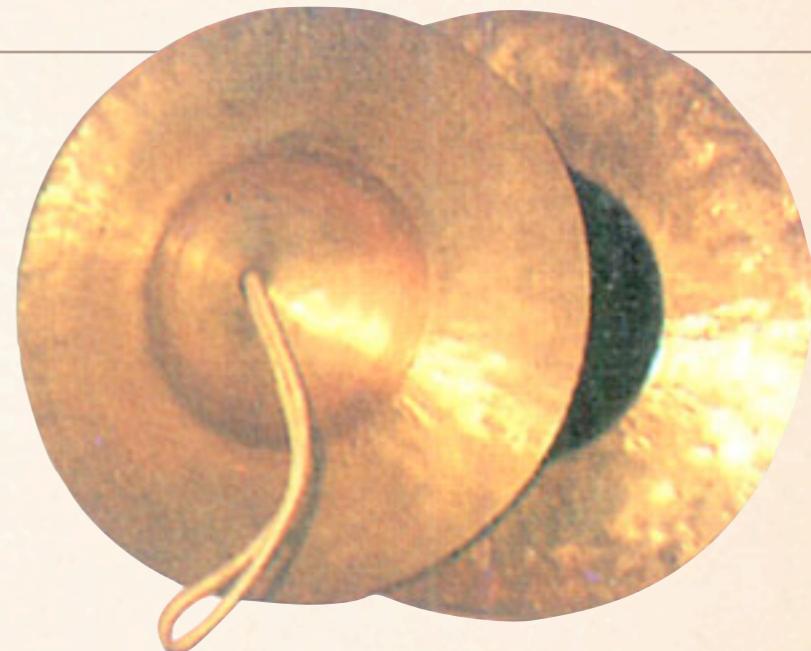
❖ I..5

❖ I...5

❖ a..z



# SYMBOLS



- ❖ :symbols\_are\_fun
- ❖ A static name
- ❖ Stored like numbers, named like strings!

# CONDITIONALS

- ❖ if, else
- ❖ unless, else
- ❖ and && (BEWARE: precedence!!)
- ❖ or || (BEWARE: precedence!!)
- ❖ ||=
- ❖ conditional ? then : else

# MAKING OBJECTS

- ❖ Tell Ruby About Something New!
- ❖ A new class of stuff!

# WHAT IS A BOOK

- ❖ All Books have....??
- ❖ TDD

# DEFINING NEW CLASSES

- ❖ Initialize
- ❖ attributes / readers and writers
- ❖ Scope
  - ❖ Instance variables
  - ❖ Instance methods

# INHERITANCE

- ❖ Parent-Child
- ❖ SubTypes



# MODULES

- ❖ Sharing Behavior
- ❖ Namespacing
- ❖ DRY Code
- ❖ Composition



# INHERITANCE EXERCISE

- ❖ monster.rb
- ❖ Monster
  - ❖ Vampire (vampire.rb)
  - ❖ Zombie
  - ❖ Werewolf

# MODULE EXERCISE

- ❖ named\_thing.rb
- ❖ Vampire's have names
- ❖ Human's have names

# COLLECTIONS



# ARRAY

- ❖ Ordered List
- ❖ Referenced by Order(index)



# HASH

- ❖ Key, Value Store
- ❖ Referenced by Name(key)

**star selections**

**2005 Lincourt Santa Barbara County (\$25)**  
Hints of allspice and sweet oak inform the bright red cherry flavors of this graceful red, made with fruit from vineyards in California's Santa Rita Hills.

**2005 Raptor Ridge Willamette Valley (\$30)**  
Most of the grapes in this elegant, multi-vineyard blend come from the Stony Mountain Vineyard, perched on a hillside in Oregon's Yamhill County.

**2004 Woodenhead Russian River Valley (\$36)**  
Co-owner Nikolai Stez (former assistant winemaker at cult producer Williams Selyem) produces stunning Pinot Noirs, like this hauntingly floral bottling.

**2005 Scott Paul La Paulée (\$40)**  
Scott Paul Wright leads two lives, importing high-quality Burgundies and making his own eponymous wines, like this lightly herbal, raspberry-rich cuvée.

# ARRAY AND HASH DEMO

- ❖ numbers = [5,4,7,1,3,2,6]
- ❖ numbers[0]
- ❖ numbers.size
- ❖ more\_numbers = {"one" => 1, "two" => 2, 3 => "three"}
- ❖ more\_numbers["one"]

# EXERCISE

- ❖ Calculator Advanced Spec

# ENUMERABLE

- ❖ Module for Collection Specific Actions
- ❖ Search
- ❖ Sort
- ❖ Iterate (Enumerate)



# ENUMERABLE

- ❖ select -> gets a subset
- ❖ map -> gets the same size
- ❖ a = [1,2,3]
- ❖ a.select{|i| i.odd?} => [1, 3]
- ❖ a.map{|i| i.odd?} => {true, false, true}
- ❖ a.map {|i| i+1} => [2,3,4]

# RUBY TO \_SQL

- ❖ \$monsters.find\_all{|i| i[:name] == "Vampire"}
  - ❖ Select \* from monsters where name = "Vampire";
- ❖ \$monsters.map{|i| i[:name]}
  - ❖ Select name from monsters;
- ❖ \$monsters.find\_all{|i| i[:legs] == 2}.map{|i| i[:name]}
  - ❖ Select name from monsters where legs = 2

# INJECT

```
total = 0
```

```
[1,2,3].each do |i|
```

```
    total += i
```

```
[1,2,3].inject(:+)
```

```
end
```

```
=> 6
```

```
total
```

```
=> 6
```

# DEMO ENUMERABLE

- ❖ load ‘monsters.rb’
- ❖ \$monsters
  
- ❖ How many monsters are nocturnal?
- ❖ What are the names of the monsters that are nocturnal?
- ❖ How many legs do all our monsters have?
- ❖ What are the 2 most common dangers and vulnerabilities of our monsters?

# LUNCH

---

I HOUR

# REGEXP!

---

REGULAR EXPRESSIONS IN RUBY

# PATTERNS

- ❖ Find a pattern in text

# /FIND ME/

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

# 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| ‘\_’}

# EXERCISE!

❖ pig\_latin\_spec.rb

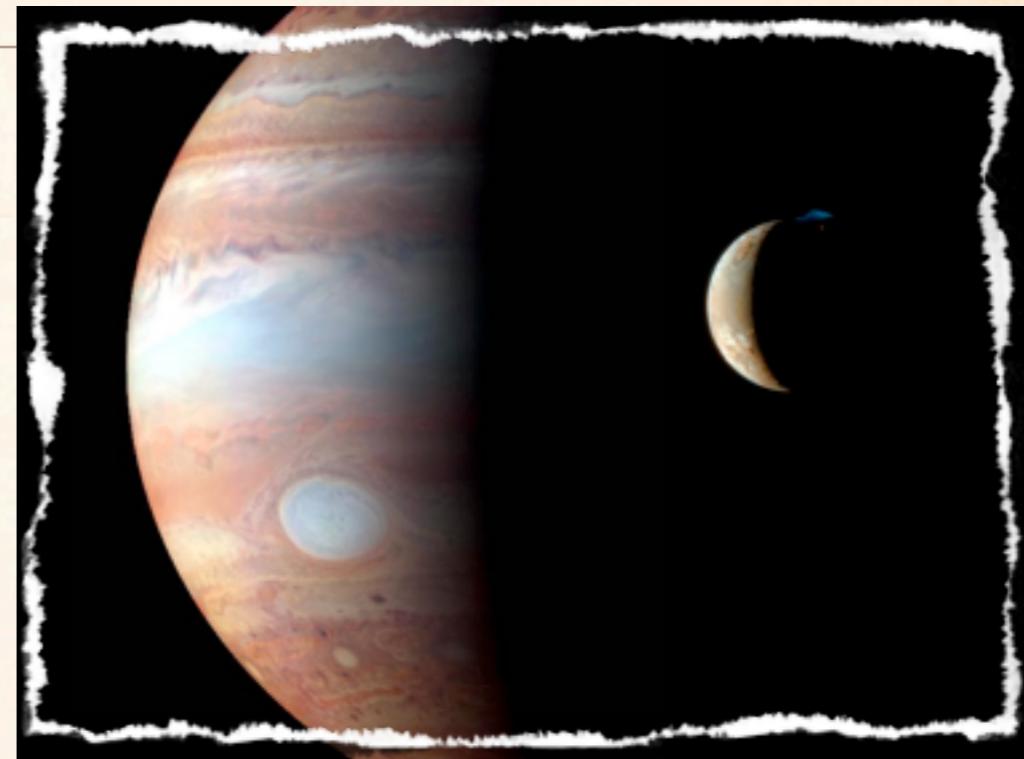
# IO, FILES, AND RAKE

---

STREAMS AND TASKS

# 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

# BREAK

---

10 - 15 MIN

# ADVANCED RUBY

---

WITH GREAT POWER...

# BLOCKS

- ❖ A method with no name!
- ❖ Dynamic method definition

# BLOCK SYNTAX

- ❖ {
- ❖ do
  - ❖ <code>
- ❖ end

# BLOCKS:

- ❖ yield
- ❖ call
- ❖ block\_given?

# BLOCKS WITH PARAMETERS

- ❖ yield(x)
- ❖ call(x)
- ❖  $\{|x|\}$
- ❖ do |x|
  - ❖ <code>
- ❖ end

# BLOCKS DEMO

- ❖ Time My Code!
- ❖ timer\_spec.rb

# BLOCKS EXERCISE

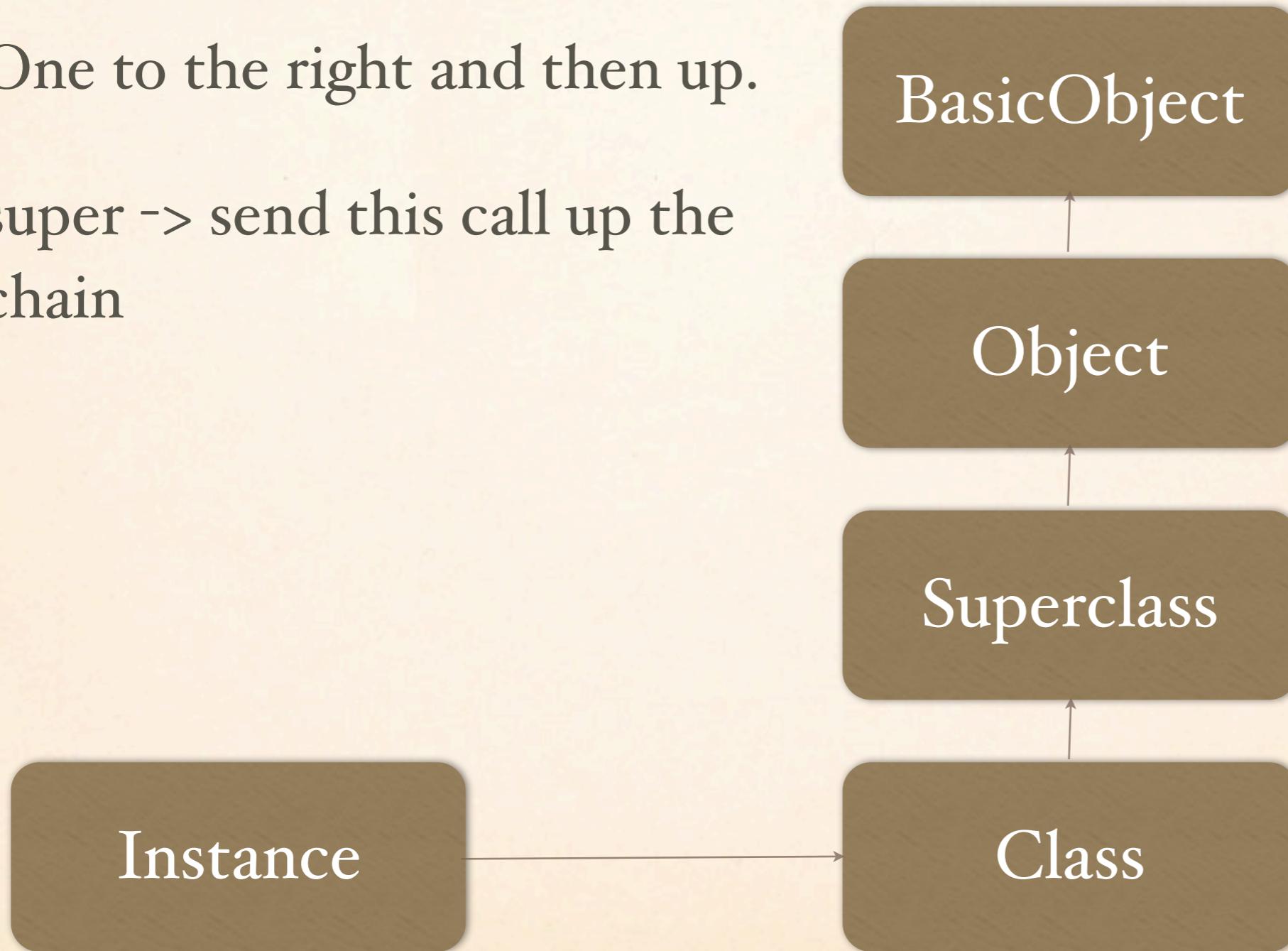
- ❖ Do My Work!
- ❖ worker\_spec.rb

# METHOD MISSING!

- ❖ Up the chain, then back!
- ❖ The final resting place of method calls (most of the time!)
- ❖ It's Magic, and you can too! :)

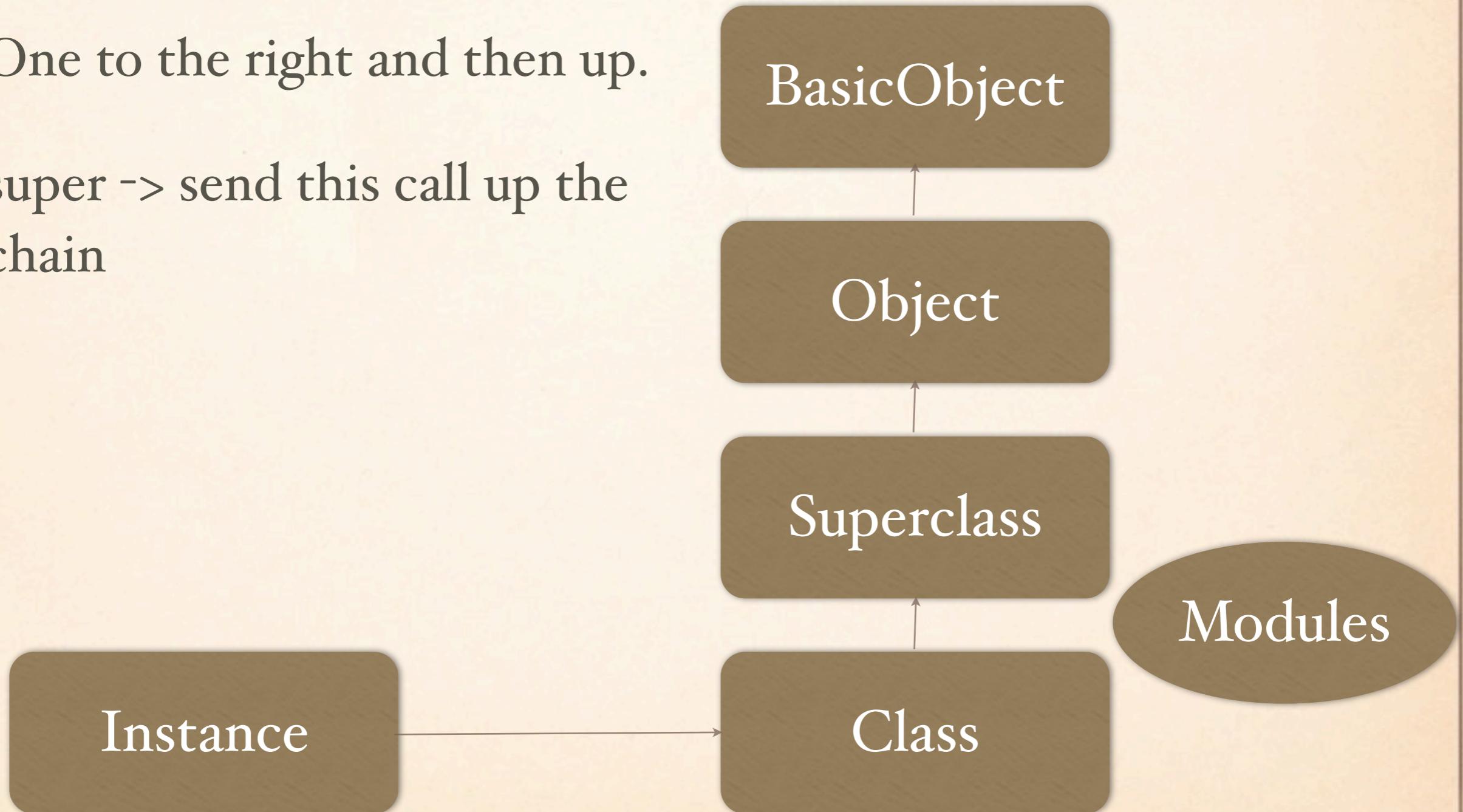
# RUBY CALL CHAIN

- ❖ One to the right and then up.
- ❖ `super` -> send this call up the chain



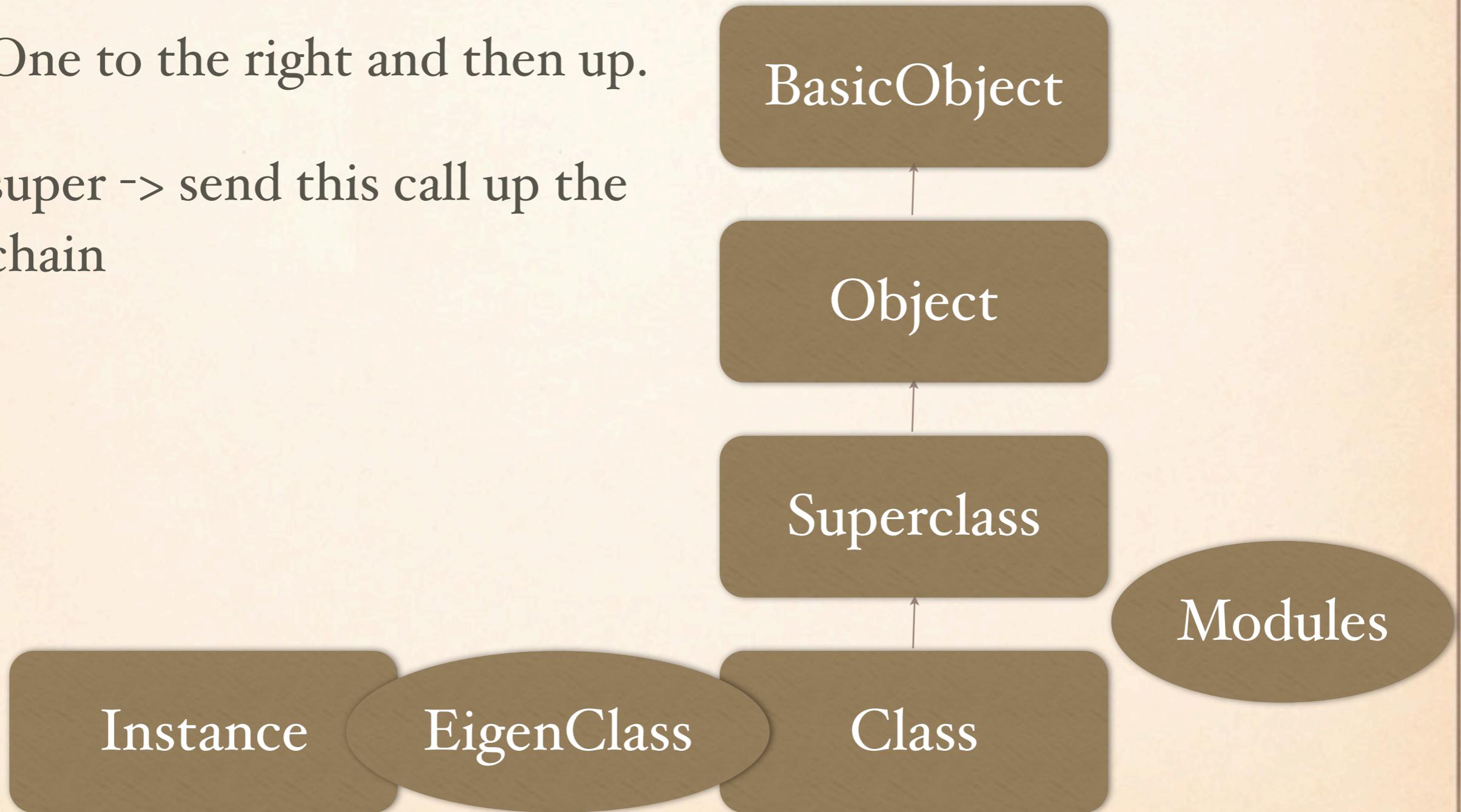
# RUBY CALL CHAIN

- ❖ One to the right and then up.
- ❖ `super` -> send this call up the chain



# RUBY CALL CHAIN

- ❖ One to the right and then up.
- ❖ `super ->` send this call up the chain



# METHOD MISSING

```
def method_missing(sym, *args, &block)
  puts "You asked for #{sym} with #{args.join(" ")}"
  super
end

**def respond_to?
```