Learn to Program with Ruby

An element in the UPLVLS

About me

- Senior undergrad in CS+Math at UW-Madison, console cowboy
- Undergraduate Projects Lab coord
- Using Ruby ~2 years
- Several projects
- Some Ruby on Rails (web dev)
- Attended Madison+Ruby 2014

About you

- Have a computer or a friendly partner
- Have basic math and reasoning skills (sorry preschoolers)
- Want to learn to program, or want to learn a new language
- Are intrigued by the power and freedom that comes with having programming knowledge

Why learn to program?

- Employment
- Practicality
- Improve critical and abstract thinking
- Become a better problem solver
- Become part of a community
- Be a rockstar
- Make your dreams come true!

How computers work

A layered system of discrete, finite computations on memory as proposed by Turing and Von Neumann.

A bunch of “stupido” robots that speak in 0’s and 1’s that update numbers in mailboxes based on simple commands.

How do programs work?

- Program = human-readable file of text characters
- Interpreter = a program that reads a chunk of another program and performs an action
    - Compilers read the whole program and then output a blob of machine code
- 0’s and 1’s sent to the CPU

About Ruby

- Created in Japan in the 90s by a guy named Matz
- Focus on programmer happiness and human-readability (i.e. English)
- Dynamic, strong types
- Object-oriented, with functional elements
- Multiple ways to do things
- Lively, inclusive community
- Elixir, Crystal, Rails, APIs, and even Harmony...

My Opinions

- vs Python
  - + Dynamicism
  - (+ OOP) & (+ FP) => great readability
  - - “Science”
- vs JavaScript
  - + Better design
  - + User-friendliness
  - + Stable/mature
  - - Web (until wasm)
- Types of “speed”
  - Run time
  - Compile time
  - Implementation time
  - Debug time
  - Testing time
  - Fun time!

Lingo

- Variable
  - A box containing something
- Integer (Fixnum)
  - A number that has no fractional component
- Float
  - A number with fractional components
- String
  - A sequence of characters
- Method
  - A procedure that takes arguments and returns a value (i.e. a function)
- Object
  - A collection of data and methods that responds to messages. In Ruby, everything is an object!
- Class
  - A blueprint for manufacturing objects

Objects are basically...

messages

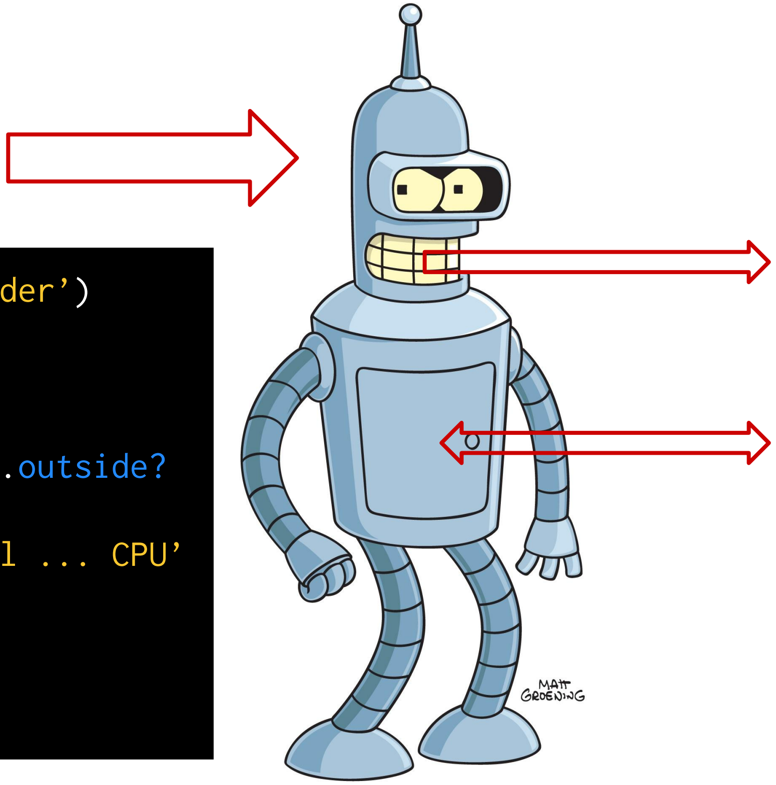
```
bender = Robot.new('Bender')

bender.drink!

bender.smoke! if bender.outside?

s = 'Kiss my shiny metal ... CPU'

bender.shout(s)
```



Example

```
# pass msg reverse to obj earth, store in box bizarro
bizarro = ‘earth’.reverse
puts bizarro # use puts for outputting things
# pass msg upcase to obj bizarro
loud_bizarro = bizarro.upcase
puts loud_bizaro
# what do you think this does?
angry_loud_bizarro = loud_bizarro + (‘!’ * 5)
puts angry_loud_bizarro
```

Example

```
x = 6

greeting = ‘hello’

char_count = greeting.size

if x - char_count > 0
  print greeting + ‘ world’
end
```

- x is a box containing 6
- word is box containing ‘hello’
- char_count is a box containing words’s size at the time of that command
- What happens?

Now for the fun part...

- Head to http://repl.it/languages/Ruby
    - Or use `ruby` || `irb` if you have them
- Work with a partner if you can!
- See the gist for a guide, or feel free to explore!
- If you have questions, let me know!
- Internet search engines are your friend!