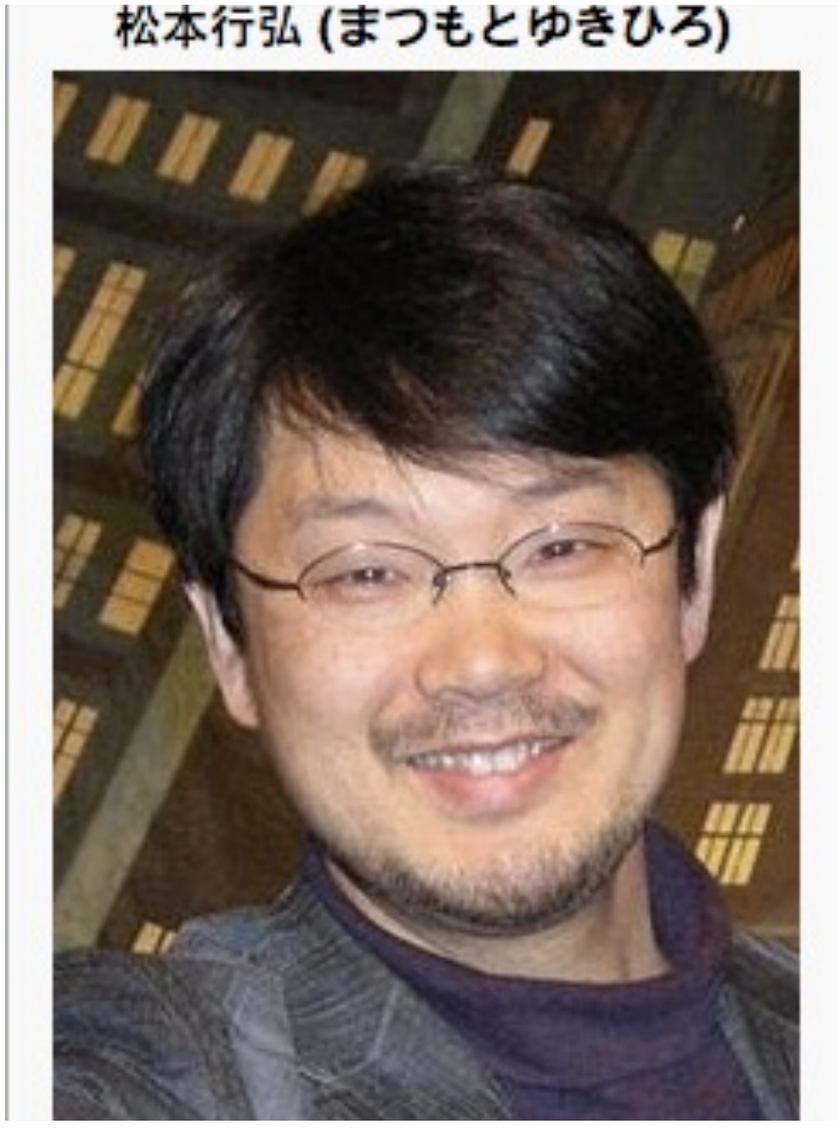
Ruby Basics

Like Git Much?

- gist (https://gist.github.com)
- http://ndpsoftware.com/git-cheatsheet.html
- https://gist.github.com/4570933

Who Is He?



Object-Oriented Language

- Modeling real-world objects (car, person)
  - Class of objects (car v.s. bmw, toyota)
    - Attributes (make, model, color, horse-power)

Object Types

- Number (Integer, Float)
- String
- Foo
- Class

```
zlu@zlu-mba:~/tmp/myproject (master)$ irb
1.9.3-p327 :001 > 1.class
 => Fixnum
1.9.3-p327 :002 > 1.0.class
 => Float
1.9.3-p327 :003 > "abc".class
 => String
1.9.3-p327 :004 > class Foo; end
 => nil
1.9.3-p327 :005 > f = Foo.new
 => #<Foo:0x007fc611ac3850>
1.9.3-p327 :006 > f.class
 => Foo
1.9.3-p327 :007 > Foo.class
 => Class
```

Variables

- Reference to an object (or a thing)
  - office_address = ‘1 Broadway, New York’
  - memory allocation
  - addressing
- Naming convention
  - Class Studio
  - Attribute studio_reviews

Constants

- A rather static reference to an object.
- Always in upper case.
- Variable can change
  - Reference (address) can change

```
1.9.3-p327 :008 > s = 's'
 => "s"
1.9.3-p327 :009 > s = 't'
 => "t"
1.9.3-p327 :010 > S = 's'
 => "s"
1.9.3-p327 :011 > S = 't'
(irb):11: warning: already initialized constant S
 => "t"
1.9.3-p327 :012 >
```

Arithmetics

```
1.9.3-p327 :012 > 1 + 2
 => 3
1.9.3-p327 :013 > a = 1
 => 1
1.9.3-p327 :014 > a += 2
 => 3
1.9.3-p327 :015 > a * 2
 => 6
1.9.3-p327 :016 > a *= 2
 => 6
1.9.3-p327 :017 > a / 2
 => 3
```

Type Coercion

```
1.9.3-p327 :026 > a /= 2
 => 3
1.9.3-p327 :027 > a
 => 3
1.9.3-p327 :028 > a /= 2
 => 1
1.9.3-p327 :029 > a = 3
 => 3
1.9.3-p327 :030 > a /= 2.0
 => 1.5
```

String Concatenation

```
1.9.3-p327 :031 > "s" + "tring"
 => "string"
1.9.3-p327 :032 > "string" - "ing"
NoMethodError: undefined method `-' for "string":String
        from (irb):32
        from /Users/zlu/.rvm/rubies/ruby-1.9.3-p327/bin/irb:16:in `<main>'
```

String Interpolation

```
1.9.3-p327 :033 > s = "string"
 => "string"
1.9.3-p327 :034 > another_s = "another #{s}"
 => "another string"
1.9.3-p327 :035 >
```

Equality

```
1.9.3-p327 :048 > true
 => true
1.9.3-p327 :049 > true.class
 => TrueClass
1.9.3-p327 :050 > false
 => false
1.9.3-p327 :051 > false.class
 => FalseClass
1.9.3-p327 :052 > 1 == 1
 => true
1.9.3-p327 :053 > 1 != 1
 => false
1.9.3-p327 :054 > 1 > 0
 => true
1.9.3-p327 :055 > 1 < 0
 => false
1.9.3-p327 :056 > 1 >= 1
 => true
```

Boolean Operators

```
1.9.3-p327 :057 > true && false
 => false
1.9.3-p327 :058 > true || false
 => true
1.9.3-p327 :059 > !true
 => false
1.9.3-p327 :060 > !false
 => true
```

Conditionals

```
1.9.3-p327 :061 > if true
1.9.3-p327 :062?>   puts 'foo'
1.9.3-p327 :063?>   else
1.9.3-p327 :064 >     puts 'bar'
1.9.3-p327 :065?>   end
foo
 => nil
```

if v.s. unless

```
1.9.3-p327 :066 > unless true
1.9.3-p327 :067?>   puts 'foo'
1.9.3-p327 :068?>   else
1.9.3-p327 :069 >     puts 'bar'
1.9.3-p327 :070?>   end
bar
 => nil
```

case statement

```
1.9.3-p327 :071 > s = "s"
 => "s"
1.9.3-p327 :072 > case s
1.9.3-p327 :073?>   when "t"
1.9.3-p327 :074?>   puts "t"
1.9.3-p327 :075?>   when "s"
1.9.3-p327 :076?>   puts "hello world"
1.9.3-p327 :077?>   end
hello world
 => nil
```

Outside of command line

```
zlu@zlu-mba:~/tmp/myproject (master)$ echo 'puts "hello world"' > hello.rb
zlu@zlu-mba:~/tmp/myproject (master)$ ruby hello.rb
hello world
zlu@zlu-mba:~/tmp/myproject (master)$
```

ruby - interpreted language

ruby interpreter

`which ruby`

String I/O

```
zlu@zlu-mba:~/tmp/myproject (master)$ echo "s = gets; puts s" > foo.rb
zlu@zlu-mba:~/tmp/myproject (master)$ ruby foo.rb
hello world
hello world
zlu@zlu-mba:~/tmp/myproject (master)$
```

Loops

```
puts "what is 2 to the 16th power?"
answer = gets.chomp.to_i

while (2**16) != answer
  print "wrong! try again. "
  answer = gets.chomp.to_i
end
puts "good job!"
```

Method

- Define a method with keyword def
- Method name
  - Begins with underscore or lowercase letter
  - Ends with letter, ?, !, or =
  - “s”.length - 1.even? - “s “.chop!
  - Methods does a specific thing

Method Parameters

```
zlu@zlu-mba:~/tmp/myproject (master)$ irb
1.9.3-p327 :001 > def area(height, width)
1.9.3-p327 :002?>   height * width
1.9.3-p327 :003?>   end
 => nil
1.9.3-p327 :004 > area(5, 4)
 => 20
```

Lab - Calculator

- Build a basic calculator supporting +,-,x,/

```
(a)dd, (s)ubtract, (m)ultiply, (d)ivide: a
first number: 4
second number: 5
9.0
```

```
(a)dd, (s)ubtract, (m)ultiply, (d)ivide: d
first number: 5
second number: 2
2.5
```

Lab - Calculator

```
(b)asic, (a)dvanced, or (q)uit: a
(p)ower, (s)qrt: p
first number: 2
second number: 3
8.0
```

```
(b)asic, (a)dvanced, or (q)uit: q
zlu@zlu-mba:~/projects/ga/WDI_Curriculum/Week_01/
```

Daily Review

- Ruby is an object-oriented programming language
- Ruby has a powerful type system: numbers, string
- Ruby syntax: variable, constant, assignment
- Loops, methods

Hw - Trip Calculator

- Inputs
  - Distance
  - Miles per gallon
  - Cost per gallon
  - Speed
  - For every 1 MPH over 60 MPH, reduce the the MPG by 2 MPG. (i.e. a car that normally gets 30 mpg would only get 28 mpg if its speed were 61 mph. Yes this gets silly at high speed where mpg goes to zero or gets negative.)
- Output: "Your trip will take 3.5 hrs and cost $125