Introduction to Ruby

History of Ruby

History of Ruby

- First released in 1993
- Version 1 in 1996
- Version 1.8 in 2003
- Rails released in 2005
- Mac OS X starts having Ruby by default in 2007
- Currently at Version 2.5.1

Philosophy of Ruby

- "Making programmers happy"
- There is more than one way to do it
- There is no perfect programming language
- Principle of least astonishment

Important Links

- The Ruby programming language
- Ruby on Github
- Ruby Docs
- Yukihiro "Matz" Matsumoto
  - Twitter
  - Github
- AirBnB Styleguide
- The Ruby Style Guide

Installation of Ruby

Installation Overview (Windows Only!)

Follow the steps here

Make sure you use RVM!

Installation Overview (Mac Only!)

1. Get some developer tools
2. Install RVM
3. Include RVM in your startup scripts and PATH
4. Install and use a version of Ruby
5. Install common gems

Developer Tools

```
xcode-select --install
```

Install RVM

```
curl -sSL https://get.rvm.io | bash -s stable

# Restart the terminal

rvm get stable --auto-dotfiles
```

Install Ruby

Find the most recent version here

```
rvm install ruby-2.5.1
rvm --default use 2.5.1
```

Let's test that it works!

```
ruby -v
rvm -v

which ruby
```

Let's install some common "Gems"

```
# Useful
gem install pry

# Not overly useful...
gem install lolcat
brew install fortune
brew install cowsay
brew install ponysay
brew install cmatrix
```

Some common commands

- `ruby -v`
- `which ruby`
- `ruby file.rb`
- `pry`
- CTRL + D

Ruby Building Blocks

Data Types

Data Types

- Strings
- Numbers
- Symbols
- Methods (like functions)
- Arrays
- Hashes (like objects)

Strings

```
'Hello'

'It's a string'

"Hello"

"The person said "How are you?""

"This is a string".methods
```

Interpolation with Double Quotes

```
result = 2 + 2

"The result is #{result}"

# This will only work with double quotes!
```

Numbers in Ruby

```ruby
1.1

17.68

1234

1_560_142
# Underscores are ignored
# Very useful for readability though
```

Arithmetic in Ruby

```ruby
10 + 4
10 - 6
10 * 12
10 / 12

10 < 12
12 > 10
10 >= 10
12 <= 12

10 == 10
10 != 9
```

Variables

Variables in Ruby

```ruby
this_is_ruby = true
this_is_a_string = "Yes, it is"
this_is_a_number = 1241
this_is_a_number += 1
this_is_a_number -= 1

empty_array = []
empty_hash = {}

name = "Gilberto"

# snake_case in Ruby, camelCase in JS
```

Conditionals

`if` Conditionals

```
if 42 > 13
  p "42 is a bigger number"
end
```

if, elsif, else

```
name = "Groucho"

if name == "Harpo"
  # Do something
elsif name == "Chico"
  # Do something else
else
  # Do something else
end
```

`if` Conditionals

```
p "42 is bigger" if 42 > 13
```

`unless` Conditionals

```
x = 1

unless x > 2
  puts "x is less than 2"
else
  puts "x is greater than 2"
end

code_to_perform unless conditional
```

`case` Statement

```
num_of_wheels = 1

case num_of_wheels
when 1
  p "Unicycle"
when 2
  p "Bicycle"
when 4
  p "Car"
else
  p "I'm not sure"
end
```

case Statement

```
hour = 9

case
when hour < 12
  p "Good Morning"
when hour > 12 && hour < 17
  p "Good Afternoon"
else
  p "Good Evening"
end
```

case Statement

```
hour = 9

case hour
when 0..12
  p "Morning"
when 13..17
  p "Afternoon"
else
  p "Night"
end
```

Logical Operators

```
&&
||
!
```

Debugging in Ruby

```
name = "Jacques"

p name
puts name
```

```
require 'pry'

# ...

binding.pry
```

In-class Exercise / Homework

Have a go at these exercises

Loops

The `while` loop

```
while conditional
  # Statements to execute
end

while true
  puts "This is a great idea"
end

i = 0
while i < 5
  puts "I: #{ i }"
  i += 1
end
```

The `until` loop

```
until conditional
  # Statements to execute
end
```

```
i = 0
until i == 5
  puts "I: #{ i }"
  i += 1
end
```

The `for` loop

```
# These aren't regularly used in Ruby

for i in 0..5
  puts "I: #{ i }"
end
```

Loops: Iterators

```ruby
5.times do
  puts "Wow"
end

5.times do |i|
  puts "I: #{i}"
end

5.downto(0) do |i|
  puts "I: #{ i }"
end

5.upto(10) do |i|
  puts "I: #{ i }"
end
```

Generating Random Numbers

```
# Generates a number between 0 and 1
Random.rand

# Generates a random number up to 10 (including zero and 10)
Random.rand(10)

# Generates a number between 5 and 10 (also includes them)
Random.rand(5..10)

# Does not include 10
Random.rand(5...10)
```

In-class Exercise / Homework

Have a go at these exercises

Methods

Methods

```
def hello
  p "Hello World"
end

hello
hello()
```

Methods

```
def hello( name )
  p "Hello #{ name }"
end

hello "Roget"
hello( "Roget" )
```

Methods: Implicit Returns

```ruby
def add( first, second )
  result = first + second
  # no need for `return`: the value of the
  # method's last line is implicitly returned
  result
end
```

```ruby
def add(first, second)
  first + second
end

def add( first, second )
  result = first + second
  return result
end
```

Methods: Default Parameters

```
def add(x = 0, y = 0)
  x + y
end

add(4, 5)
add(4)
add
```

Homework

- Finish off the in-class exercises
- Then, these exercises