Learn To Program

In Ruby

Class Overview

- Strings, Numbers, Variables
- Arrays & iterators, Methods
- Hashes & iterators, Methods
- Objects & Classes
- Sinatra, HTML, CSS
- Orange tree
- Files
- Blocks

What is a Program?

What is a Program?

- Instructions to your computer.
- Programs are written in a code that computers understand.

Platforms

A platform is a special program called an operating system.

It provides a structure for other programs to run.

Operating Systems:

- Windows
- Linux
- Mac

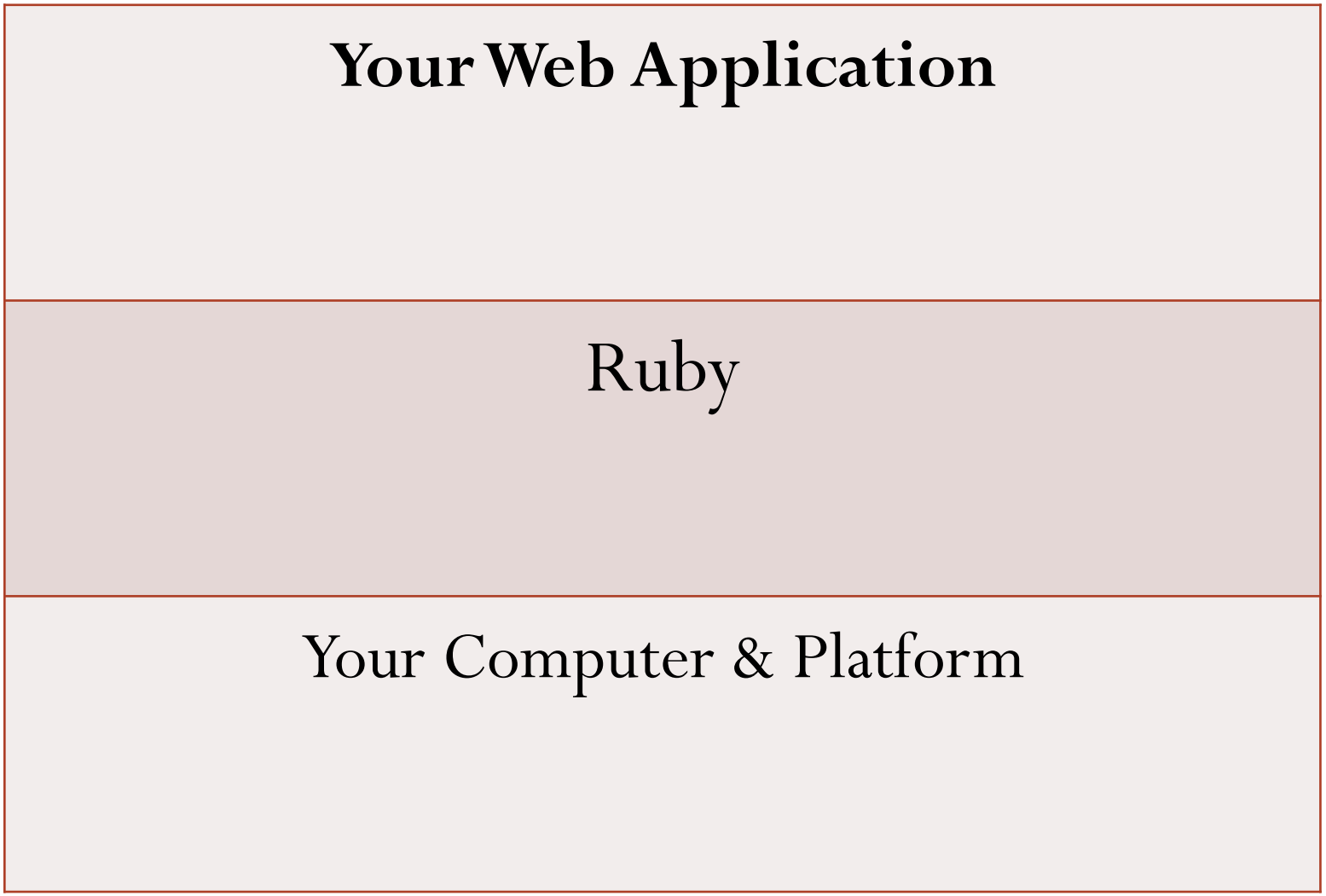
Some Programs You Use

Platform = Your computer

- Microsoft Office
- Firefox
- Itunes
- Photoshop

Platform = Server owned or rented by a company

- Google
- eBay


Your Web Application
Ruby
Your Computer & Platform

How Do I Write a Program?

- Learn about customer's requirements
- Translate to "stories"
- Pick a story that seems doable
- Write code that does it
- Show your work to the customer, get feedback
- Based on feedback, adjust your stories
- When a story is done, go back to "pick a story"
- Repeat until app is finished!

Tools for Writing Code

- ruby
- irb
- using ruby gems
- command line
- html / css
- deploying a web app on heroku
- git
- tdd, testing
- rspec, sinatra

Let’s Do Some Ruby!

Data Types

- Programmers create replicas of the real world inside of computers. We need a way to describe the world's people, places and things.
- The first step is to allow for different behaviors of data.
  - You want a person represented in your program to have a name
    - Use a string
  - You want to count all the people in your program.
    - Use an integer
- Some Common data types: String, Integer, Array, Symbol, Hash

Wordy Data Types:

- String
  - “I am a string. I can contain letters, spaces, numbers, variables, and things like#$%^&&*! The quotation marks at the beginning and end are the way a computer recognizes where I begin and where I end”
  - “hello world”

- Symbol
  - :i_am_a_symbol_i_can_contain_letters_and_underscores
  - :i_am_used_to_give_variables_of_other_data_types_a_name

Number Data Types

- Integer
  - 5
- Float
  - 4.566

Collection Data Types

- Array
  - `["string", 333, 4.2, "another string", :color]`
- Hash
  - `{"key" => "value", :name => "Sally", :age => 14}`

Looking Closer at Basic Data Types

- Integers
- Floats
- Strings
- Symbols

Numbers

- Integer
- Float

Four Basic Operations

```
>> 1+1
=> 2
>> 4-3
=> 1
>> 2*5
=> 10
>> 8/2
=> 4
>> -5 + 3
=> -2
```

Use Float for Decimal Precision

```
>> 23/2
=> 11
>> 23.0/2.0
=> 11.5
```

Testing Numbers

```
>> 4<9
=> true
>> my_age == 27
=> true
>> my_age != 27
=> false
```

Strings

Stringing together characters

Quotes

```
>> "hello”
=> "hello"
>> 'hello’
=> "hello"
>> hello
>>NameError: undefined local variable
  or method `hello' for
  main:Objectfrom (irb):25
```

Concatenation

```
>> "hello" + " " + "world"
=> "hello world"
```

Puts

- Stands for “put” to “s”tring
- Takes an argument (a number, string, or other object) and prints the argument to the screen in the form of a string.
- When you are running a web application, puts will print the string to the server log
- **Warning: It returns nil!** (nil means no value)

Puts Returns nil

```
>> "hello, what is your name"
=> "hello, what is your name"

>> puts "hello, what is your name"
hello, what is your name
=> nil
```

Exercise

- You are writing a program that has the user's city and state in separate database fields. The user's city is "Austin" and their state is "Texas". Concatenate the city and state together with a ", " in between.

Strings and Numbers Together

Multiply a string

```
>> puts "repeat me 3 times\n" * 3
repeat me 3 times
repeat me 3 times
repeat me 3 times
=> nil

>> puts 10 * "repeat me 10 times”
TypeError: String can't be coerced
  into Fixnumfrom (irb):42:in `*'from
  (irb):42
```

Exercise

- Write a program that repeats the string “I do believe in fairies” 16 times.

Numbers Inside of Strings

```
>> "a string with just 1 number in it”
=> "a string with just 1 number in it"
>> "1 string with 2 numbers in it”
=> "1 string with 2 numbers in it"
>> "34”
=> "34"
>> "34" + "1”
=> "341"
>> "34 + 1”
=> "34 + 1"
```

These won't work:

```
>> "34" + 1
TypeError: can't convert Fixnum into
  Stringfrom (irb):36:in `+'from
  (irb):36


>> "3" * "9”
TypeError: can't convert String into
  Integerfrom (irb):40:in `*'from
  (irb):40
```

Single and Double Quotes

```
>> 'this won't work'
```

because the quote in "won't" is not escaped

TWO SOLUTIONS:

1. double quotes around a string with a single quote in it
2. escape the single quote with a slash

```
>> "this won't not work"
=> "this won't not work"
>> 'this won\'t not work'
=> "this won't not work"
```

Variables

```
>> my_age = 27
=> 27
>> your_age = 43
=> 43
>> our_ages = my_age + your_age
=> 70
>> your_age = 44
=> 44
>> our_ages
=> 70
>> our_ages = my_age + your_age
=> 71
```

String Interpolation

```
>> animal = "cow”
=> "cow"
>> "the #{animal} ran away”
=> "the cow ran away"
>> 'the #{animal} ran away’
=> "the \#{animal} ran away"
```

Variables with String Interpolation

```
>> birthday = "March 3rd”
=> "March 3rd"
>> name = "Sally Smith”
=> "Sally Smith"
>> string = "#{name}s birthday is #{birthday}”
=> "Sally Smiths birthday is March 3rd"
>> birthday = ”September 30th”
=> ”September 30th”
>> string = "#{name}s birthday is #{birthday}”
=> "Sally Smiths birthday is September 30th"
```

Exercises

- Write a program that compliments the user's eye color. If the user's name is Sally and her eye color is brown, the program should say:
  - "Wow Sally, your brown eyes sure do sparkle!"
- Write a program that calculates how old you are in seconds

gets & chomp

- gets is a Ruby method that takes command line input from the user's keyboard and makes it available to your program.
- chomp removes the new line character (\n) from the end of gets.

```
puts "what's up"
answer = gets.chomp
```

who_took_my_cheese.rb

More Exercises

- Full name greeting:
  - Write a program that asks for a person's first name, then middle and then last. Finally, it should greet the person using their full name.
- Bigger better favorite number:
  - Write a program that asks for a person's favorite number. Have your program add 1 to it and then suggest the result as a bigger and better favorite number.

A few words about Classes & Objects

- Everything in Ruby is an object.
- Objects are instances of Classes

```
>> 5.class
 => Fixnum

>> "hello".class
 => String

>> 4.5.class
 => Float

>> :hello.class
 => Symbol
```

Making a new Class

```
class Daisy
end
```

Instanciating the class:

```
d = Daisy.new
```

Making a Methods for Our Class

```
class Daisy
   def name
         “Daisy”
   end
end

d = Daisy.new
d.name
```

Exercise

- Make five new methods on the Daisy class:
  - num_petals should return 30 (as an integer)
  - color should return white (as a symbol)
  - smell should return delicious (as a string
  - age should return 2 days (as a string)
  - height should return 10 inches (as a string)

- Instanciate your daisy class and try calling all your methods on it.

Homework

- Read the following:
  - http://pine.fm/LearnToProgram/?Chapter=01
  - http://pine.fm/LearnToProgram/?Chapter=02
  - http://pine.fm/LearnToProgram/?Chapter=03
  - http://pine.fm/LearnToProgram/?Chapter=04
- Complete exercises not finished in class
- Make a program that tells the first line of a knock knock joke, records the joke reciever's answer ("who's there"), and reply with a witty punchline. Some inspiration:
  - http://www.google.com/search?q=knock+knock+jokes&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a