Learn to Program with Ruby

RailsBridge Triangle 2015

Goals

- You we'll be able to read Ruby code
- You won't memorize anything in these slides

Operating System

"An operating system (OS) is software that manages computer hardware and software resources and provides common services for computer programs."

Applications

“A set of one or more programs designed to carry out operations for a specific application”

It is dependent on system software to execute

* Chrome
* Firefox
* Word
* Spotify

Language

- Ruby
- Python
- Perl
- Java
- C++

Many others.

“A programming language is a formal constructed language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behavior of a machine or to express algorithms.”

Q: How is a computer language similar to a human language, like English or Spanish? How is it different?

* Most languages are complete, just like you can express anything in English or Spanish, you could write any application with most programming languages.

* Programming language’s grammar are usually very strict when compared to human languages.

Library

- A collection of reusable code to accomplish a generic activity.

Examples

* User/password handling
* Credit card processing
* Image manipulation
* Currency handling

Framework

- Collection of reusable code to facilitate development of a particular product or solution.

Very vague, let’s jump into a concrete example instead.

Ruby, Ruby on Rails and Gems

- Ruby is the language
- Gems are Ruby's libraries
- Ruby on Rails is a framework

Ruby on Rails

- Ruby on Rails is written in Ruby
- Ruby on Rails contains many Ruby gems
- Ruby on Rails is a framework for building websites

Ruby Philosophy

- “A programmer’s best friend”
- *“I believe people want to express themselves when they program.*

  They don't want to fight with the language.

  Programming languages must feel natural to programmers.

  I tried to make people enjoy programming and concentrate on the fun and creative part of programming when they use Ruby.” - Matz (Yukihiro Matsumoto), Ruby creator

Tagline on their website.

http://linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html

Ruby is a Scripting Language

- Scripting languages:
  - Don't require a compiler.
  - Have an interpreter - runs "on the fly"
  - Easy to change frequently
- Python, Perl, and JavaScript are scripting languages too.
- Java and C++ are examples of compiled languages.

TODO add notes

Let’s start coding

Open your Terminal

- Also known as “command line”, “console”, “shell”
- Windows: **git bash**
- Mac and Linux: **Terminal**

Prompt

- Terminals show a line of text when you login & after a command finishes
- It's called the *prompt*, and customarily ends with a dollar sign
- Whenever instructions start with `$`, type the rest of the line into the terminal.
- Let's give the terminal a `command`, to open Interactive Ruby (IRB): `irb`

Windows Users! Some people have experienced trouble with backspace, delete, and arrow keys working properly in irb - what a pain! If you run into this problem, use this command instead to launch irb.

```
$ irb --noreadline
```

irb: Interactive Ruby

- IRB has its own prompt, which customarily ends with `>`

```
$ irb

>
```

- You can use `Control-C` to exit IRB any time or type **`exit`** on irb's prompt:

```
> exit

$
```

- Now you're back to the terminal's prompt.

Variables

- It has a name, so we can refer to it
- Holds information that can be changed
- Setting the variable equal to something is called “assignment”

Variable Assignment

- Variables are assigned using a single equal sign =
- The right side is evaluated first, then the value is assigned to the variable named on the left side of the equals

```
apples = 5

bananas = 10 + 5

fruits = apples + bananas

bananas = fruits - apples
```

What happened on each line? Is it that what you expected?

Variable Naming

Create a variable whose name has:

- All letters: `folders`
- All numbers: `2000`
- An underscore: `first_name`
- A dash: `last-name`
- A number anywhere: `y2k`
- A number at the start: `101dalmatians`
- A number at the end: `starwars2`

What did you learn?

Common types of information

String

- A string is text, it must be wrapped in a matched pair of quotation marks: `'Single quotes', "Double quotes"`
- Quotes must match: `"Start and end have to match'` (Fails)
- You can add strings together using `+`

Exercise
Create variables called first_name, last_name, and favorite_color.

Assign the variables to strings.
Can you print out a sentence that reads "Hi, my name is (first name) (last name) and my favorite color is (favorite color)." with these variables?

Numbers

- Integers: `0, -105, 898989`
- Floats (an approximation to a real number): `0.0, -105.56, 0.33`
- You can perform operations on both types, such as: `+, -, /, *`

Exercises
Try dividing an integer by an integer. Try dividing an integer by a float. How are the results different?

Create two integer variables called num1 and num2 and assign them your favorite numbers.

Next, compute the sum, difference, quotient, and product of these two numbers and assign these values to variables called sum, difference, quotient, and product, respectively.

Booleans

- A boolean is either `true` or `false`

```
>1 + 1 == 2

=> true

> 1 + 1 == 0

=> false
```

Exercises
Create a variable named favorite_color and assign it to your favorite color.

Create a variable named not_favorite_color and assign it to a different color.

Test to see if these variables are equal.

Collection

- Arrays: `fruits = [“kiwi”, “strawberry”, “plum”]`
  - Retrieving values (indexing) : `fruits[0]`, `fruits[3]`
- Hash: `states = {“NC” => “North Carolina”, “CA” => “California”}`
  - Key/value pairs. Keys must be unique
  - Indexing: `states[“NC”]`

What’s index 0 in fruits? what’s index 3?

Values don’t have to be unique
Key: address of the hash member
Value: variable contained by the member, and located by key name
A hash may also be known as a dictionary, associative array, or map.

Methods

- "If objects (like strings, integers, and floats) are the nouns in the Ruby language, then methods are like the verbs." - Chris Pine's "Learn to Program"
- Methods are called (used) with a `.` (dot)
- As it turns out, `5 + 5` is really just a shortcut way of writing `5.+ 5`. Operators are methods

Exercises
Create a String variable called old_string and assign it the value "Ruby is cool"

Use String methods to modify the old_string variable so that it is now "LOOC SI YBUR" and assign this to another variable called new_string.

Hint: look at the string methods "upcase" and "reverse"

Loop

- Does something repeatedly

```
> fruits.each do |fruit|
*> puts fruit
> end
kiwi
strawberry
plum
=> ["kiwi", "strawberry", "plum"]
```

Exercises
Create an array of 4 places you would like to visit.

Print out each of these places using a loop.

Example:
"I would like to visit Barcelona"
"I would like to visit Antigua"
"I would like to visit Alaska"
"I would like to visit New Orleans"

Conditional

- Does something only if a condition is true

```
> fruits.each do |fruit|
?> puts fruit if fruit == "plum"
> end
plum
=> ["kiwi", "strawberry", "plum"]
```

Exercises
Create an array called "group" that contains the names of some of the people in your Railsbridge group. Make sure you include your own name.

Using your group array, create a conditional that prints "My Name is (your name)" for your name only.

Running your Code

There are various ways to run code through a Ruby interpreter. We were using IRB earlier and now we will use a file.

Running Code from a File

- Create a file called my_program.rb, in your working directory (the folder in which your terminal is currently in)
- Type `puts "Hello World!"` and save the file:
- Run your code:

```
$ ruby my_program.rb

Hello World!

$
```

Object Oriented Programming

OOP

Class

- Describes the generic characteristics of a single type of object
- What things of this type are
- Examples: `Dog, Vehicle, Baby, String`

Object's Methods

- Defines behavioral characteristic
- What the things of the class's type do
- Examples: `bark, start, cry, upcase`

Object's Variables

- Defines attribute characteristic
- What things of class type have
- Examples: `breed, model, age, length`

Instance

- A specific incarnation of the class.
- Examples: buddy, my mom's van, your friend's baby, "Hello World!"

Recap

- Class: generic characteristics of a type
- Methods: define behavioral characteristic
- Variables: define attribute characteristic
- Instance: a specific incarnation of the class