Variables and Conditional Logic

Bill Patrianakos

Agenda

- Quiz
- Intro to Ruby & irb
- Variables
    - Numbers
    - Strings
    - Booleans
- Method Basics
- Conditional Logic
- Lab Time

Bill, remember to ask...

One a scale of 1 to 5...

- How comfortable is everyone using the command line?
- How confident do you feel in your ability to add items to a git repository and push them to GitHub?
- Does the concept of a fork make sense to you?

Quiz

1. How do I change directories using the command line?
2. What is Git and GitHub?
3. How do I add files to git?
4. What is a GitHub Repository?
5. What is the correct way to push changes to your GitHub repo?

- a. git pull origin master
- b. git commit -m "push to GitHub"
- c. git push origin master

Adding a new Git remote URL

- Let's first review the basic git concepts
- Keep your fork in sync with the main class repository
- Get the current/next week's class slides, code samples, etc. easily
- Beware of merge conflicts
- `git remote add upstream https://github.com/generalassembly-studio/CHI_BEWD1.git`
- `git pull upstream master`

Future workflow for class repo

- Work and save changes in your local repository
- Add and commit your work (git add, git commit -m)
- Pull down changes from upstream (git pull upstream master)
- Push to your fork (git push origin master)

Ruby

A programming language

- An open source programming language
- Easy to read and natural to write
- Created by Yukihiro Matsumoto (aka Matz) with the goal of building a language FOR developers
- Regularly maintained and evolved (recently reached version 2.0.0)

Rails

A web application framework

- Open source web application framework that is built in Ruby
- Allows you to create web applications that query a database.
- Created by DHH (David Heinemer Hansson) to simplify the task of building web applications, with the help of conventions

Ruby & Rails

Ruby first.

- It will be easier to navigate a Rails project once we have a basic understanding of Ruby.

- We will first teach you how to write simple Ruby scripts as stand-alone applications

- Once we have become familiarized with Ruby, we will start building Rails applications (which are essentially groups of Ruby script files that work together)

Computational Thinking

What does it mean to program?

"Learning about “for” loops is not learning to program, any more than learning about pencils is learning to draw."

—Bret Victor, Learnable Programming

Computational Thinking

- I run a store which uses a simple POS system to ring up customers and keep track of my profits
- The system needs to add up the cost of all items a customer has chosen to buy
- The system should apply a discount if the customer uses a coupon
- The system will then tell me the total cost of the items purchased

Try it yourself

- How do you think about the problem?
    - Think it through in your head
        - Create a **Mental Model**
    - Develop a step by step plan
        - This is an **Algorithm**
    - Figure out the quickest, simplest way to the solution
        - **Efficiency**
- **Using computational thinking, how would you give a robot directions from this room to leave the building**

Recap

Its about changing how you think

- Think in logical steps to solve a problem
- Use Ruby keywords to help solve those problems
  - Conditional Logic
  - Iteration

Programming Fundamentals

- In order to start writing our own Ruby programs, we need to learn some of the basic fundamental tools

- Specifically, we need to learn:
  - Variables
  - Methods
  - Conditions

- We will first learn the basics on their own, and then try to apply our skills in a simple interactive Ruby script

Saving Values

Using Variables

- We can tell our program to remember values for us to use later on
- The action of saving a value to memory is called **assignment**
- The entity we use to store the value is called a variable
- The action of getting the value from a variable is called accessing the variable
- We will use all the above techniques to store values into variables, and generate new values using existing variables

Variables

Storing Values

```
>> name = "Steven"
 => "Steven"
>> age = 2013 - 1983
 => age # 30
```

Data Types

- The types of different values we support include numbers, text, and other more complex ones we'll see in the future

- Ruby has its own names for these:

```
1                       #Fixnum
1.99                    #Float
'Hi! String here!'      #String
"I'm a string tool"     #String
```

Variables

Let's learn how to assign and access simple integer and string variables

Open up IRB to begin

Saving Code

Using Methods

- The same way we can store VALUES in memory by using variables...
- We can store CODE in memory by using methods.
- In other words, we can train the program to 'remember' a set of commands, and give that set of tasks a command name
- Then, we can call that command by name and the program will perform those tasks

Math in Ruby

Ruby Arithmetic Operators

Operator	Meaning	Example
+	Addition	8 + 10
-	Subtraction	10 – 8
*	Multiplication	12 * 2
/	Division	10 / 5
%	Modulus	10 % 6

Methods

- Let's learn how to call simple methods, and how to declare our own methods.

- Open up IRB to get started

Methods Recap

- Methods let us train the program to 'remember' a set of code to perform later

- Making a new method is called declaring a method

- Declaring a method does NOT run the method immediately

- If the method takes in variables to use while it is doing its tasks, those are called parameters

Teddit - Strings

Teddit is a news aggregator we will build during this course.

See you in 15 minutes

More Data Types

Booleans

- Besides strings and integers, Ruby also has a Boolean data type

- A boolean is a simple value that is either true or false

- When different data types are compared to each other, the result of that comparison is a boolean result (e.g. 5 < 7 => true)

Boolean

Logic Operators

Operator	Description	Example (a =4 and b= 2)
==	Equal	a == b *false*
!=	Not Equal	a != b *true*
>	Greater than	a > b *true*
<	Less than	a < b *true*
>=	Greater than or equal to	a <= b *false*
< =	Less than or equal to	a <= b *false*
⇔	same value? return 0 less than? return -1 greater than? return 1	a <=> b *1*
.eql?	same value and same type?	1.eql?(1.0) *false*

Booleans

See Week2/Examples/booleans.rb

Variables & Data Types

Recap

Data Types

- Number
- Float (number with decimals)
- String
- Booleans

Variables

- Store values
- Can be passed to methods as parameters

Conditional Logic

Decision Time

It's either TRUE or FALSE (like booleans)

If you are greater than 18 you are an adult

```
if age > 18
    puts "You are an adult"
end
```

Conditional Logic

Multiple Conditions

```
guess = 7
if guess > 5
    puts "Too high!"
elseif guess < 5
    puts "Too Low!"
else
    puts "You've guessed my hidden digit!"
end
```

Conditional Logic

Multiple Conditions

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non zero then then condition becomes true.	(A && B) is true.
||	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false.

Lab Time

- Conditional Teddit
- See teddit_conditionals.rb in the Homework folder for week 2
- Work on this for the next 15 minutes. I'll be here if you have questions or need help

Homework

Secret Number

Let's see a demo of Secret Number

- HW 1 - Secret Number

    - Secret number is a game we will incrementally build for homework during the Ruby portion of the course. Players must guess a secret number and your program will provide feedback. ---

EXIT TICKET

Please please please don’t leave before you fill it out! Link is in the Repository README file