Ruby

EduRamp ELS

The B2B Learning Specialist

Ruby

- Ruby is a programming language, while Ruby on Rails is a web application framework – a collection of pre-written code that simplifies website building
- Rails extends the Ruby language and solves everyday problems so you don’t have to reinvent the wheel

Install Ruby

- Install it from http://rubyinstaller.org/ or https://www.ruby-lang.org/en/downloads/
- *Make* sure you install Ruby 2.0 or Ruby 2.1
- In your PowerShell (Terminal) program, run ruby –v

First Exercise

- Exercise 1
- ruby Ex1.rb

Numbers and Math

- Every programming language has some kind of way of doing numbers and math
- Ex2.rb
- Next exercise has lots of math symbols
  - + plus
  - - minus
  - / slash
  - * asterisk
  - % percent
  - < less-than
  - > greater-than
  - <= less-than-equal
  - >= greater-than-equal

irb

- Type irb in your terminal
- You can do Ruby math operations here

Variables and Names

- Ex3.rb
- Ex4.rb

Strings and Text

- A string is usually a bit of text you want to display to someone or "export" out of the program you are writing
- Ruby knows you want something to be a string when you put either " (double-quotes) or ' (single-quotes) around the text
- You saw this many times with your use of puts when you put the text you want to go inside the string inside " after the puts to print the string
- Ex5.rb
- Ex6.rb

EduRamp ELS
The B2B Learning Specialist

Escape Sequences

Escape	What it does.
\\	Backslash ()
\'	Single-quote (')
\"	Double-quote (")
\a	ASCII bell (BEL)
\b	ASCII backspace (BS)
\f	ASCII formfeed (FF)
\n	ASCII linefeed (LF)
\r	ASCII Carriage Return (CR)
\t	ASCII Horizontal Tab (TAB)
\uxxxx	Character with 16-bit hex value xxxx (Unicode only)
\v	ASCII vertical tab (VT)
\ooo	Character with octal value ooo
\xhh	Character with hex value hh

Asking Questions

- What we want to do now is get data into your programs
- Ex7.rb

Prompting People for Numbers

- Ex8.rb

Parameters, Unpacking, Variables

EduRamp
The B2B Learning Specialist

- In this exercise we will cover one more input method you can use to pass variables to a script
- Ex9.rb
- Run ruby ex9.rb first 2nd 3rd
- The ARGV is the "argument variable," a very standard name in programming that you will find used in many other languages
- This variable holds the arguments you pass to your Ruby script when you run it

EduRamp ELS
The B2B Learning Specialist

Prompting and Passing

- Let's do one exercise that uses ARGV and gets.chomp together to ask the user something specific
- Ex10.rb

EduRamp ELS
The B2B Learning Specialist

Reading Files

- You know how to get input from a user with gets.chomp or ARGV
- Now you will learn about reading from a file
- This exercise involves writing two files. One is the usual ex11.rb file that you will run, but the other is named ex11_sample.txt
- This second file isn't a script but a plain text file we'll be reading in our script
- What we want to do is "open" that file in our script and print it out
- However, we do not want to just "hard code" the name ex11_sample.txt into our script

Reading Files

- ruby ex11.rb ex11_sample.txt

Reading and Writing Files

- Here's the list of commands you can give to your files:
  - close -- Closes the file. Like File->Save.. in your editor.
  - read -- Reads the contents of the file. You can assign the result to a variable.
  - readline -- Reads just one line of a text file.
  - truncate -- Empties the file. Watch out if you care about the file.
  - write('stuff') -- Writes "stuff" to the file.
  - seek(o) -- Move the read/write location to the beginning of the file.

Reading and Writing Files

- Ex12.rb
- ruby Ex12.rb test.txt

More Files

- We'll write a Ruby script to copy one file to another
- It'll be very short but will give you ideas about other things you can do with files
- Ex13.rb
- Run this one with two arguments: the file to copy from and the file to copy it to
- ruby ex13.rb test.txt new_file.txt

Names, Variables, Code, Functions

EduRamp
The B2B Learning Specialist

- Functions do three things:
  - They name pieces of code the way variables name strings and numbers.
  - They take arguments the way your scripts take ARGV.
  - Using 1 and 2, they let you make your own "mini-scripts" or "tiny commands.“
- You can create a function by using the word def in Ruby
- Ex14.rb

function checklist

- Did you start your function definition with def?
- Does your function name have only characters and _ (underscore) characters?
- Did you put an open parenthesis (right after the function name?
- Did you put your arguments after the parenthesis (separated by commas?
- Did you make each argument unique (meaning no duplicated names)?
- Did you put a close parenthesis) after the arguments?
- Did you indent all lines of code you want in the function two spaces?
- Did you end your function with end lined up with the def above?

function checklist

- When you run ("use" or "call") a function, check these things:
  - Did you call/use/run this function by typing its name?
  - Did you put the (character after the name to run it?
  - Did you put the values you want into the parenthesis separated by commas?
  - Did you end the function call with a) character?
  - Functions that don't have parameters do not need the () after them, but would it be clearer if you wrote them anyway?

EduRamp ELS
The B2B Learning Specialist

Functions and Variables

- The variables in your function are not connected to the variables in your script
- Ex15.rb

Functions and Files

- Ex16.rb
- ruby Ex16.rb test.txt

Functions Can Return Something

- You have been using the = character to name variables and set them to numbers or strings
- We're now going to look at how to use = and a new Ruby word return to set variables to be a value from a function
- Ex17.rb

The Truth Terms

- In Ruby we have the following terms (characters and phrases) for determining if something is "true" or "false.“
  - && (and)
  - || (or)
  - ! (not)
  - != (not equal)
  - == (equal)
  - >= (greater-than-equal)
  - <= (less-than-equal)
  - true
  - false

EduRamp
The B2B Learning Specialist

Boolean Practice

- Run irb and try the following:
  - true && true
  - false && true
  - 1 == 1 && 2 == 1
  - "test" == "test"
  - 1 == 1 || 2 != 1
  - true && 1 == 1
  - false && 0 != 0
  - true || 1 == 1
  - "test" == "testing“
  - 1 != 0 && 2 == 1
  - "test" != "testing"
  - "test" == 1
  - !(true && false)
  - !(1 == 1 && 0 != 1)
  - !(10 == 1 || 1000 == 1000)
  - !(1 != 10 || 3 == 4)
  - !("testing" == "testing" && "Zed" == "Cool Guy")
  - 1 == 1 && (!("testing" == 1 || 1 == 0))
  - "chunky" == "bacon" && (!(3 == 4 || 3 == 3))
  - 3 == 3 && (!("testing" == "testing" || "Ruby" == "Fun"))

What If

- Ex18.rb

Else and If

- In the last exercise you worked out some if-statements
- Ex19.rb

Making Decisions

- Ex20.rb

EduRamp ELS
The B2B Learning Specialist

Loops and Arrays

- programs also need to do repetitive things very quickly
- We are going to use a for-loop in this exercise to build and print various arrays
- Arrays are a container of things that are organized in order from first to last
- Ex21.rb

```
hairs = ['brown', 'blond', 'red']
eyes = ['brown', 'blue', 'green']
weights = [1, 2, 3, 4]
```

While Loops

- A while-loop will keep executing the code block under it as long as a boolean expression is true
- Ex22.rb

EduRamp ELS
The B2B Learning Specialist

Accessing Elements of Arrays

- animals = ['bear', 'tiger', 'penguin', 'zebra']
- bear = animals[0]
- Ex24.rb

Branches and Functions

- Ex23.rb

EduRamp ELS
The B2B Learning Specialist

Hashes, Oh Lovely Hashes

- A Hashmap (or "hash") is a way to store data just like a list, but instead of using only numbers to get the data, you can use almost anything
- This lets you treat a hash like it's a database for storing and organizing data
- Ex25.rb