Introduction to Ruby and programming

* Take up Unix Homework from yesterday
* Basic Standard I/O with Ruby
* REPL
* IRB (Pry)
* Methods
* Basic problem solving & debugging

What does a cli stand for

A **CLI** (**command line interface**) is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth

What is the difference between the following commands? Explain each one. Try them out in vagrant if you'd like:

* cd ~
* cd ~/ - slash tells change
* cd .
* cd ..
* cd /
* cd -

|  |
| --- |
| cd ~ home directory (same as cd) |
|  | cd ~/ user home |
|  | cd . will leave you in the same directory you are currently in |
|  | cd .. move you up one directory (parent directory) |
|  | cd / goes to root directory |
|  | cd - will switch you to the previous location |

Q3

Mkdir is saying make a directory –p is saying make a parent directory and after if there is anything its giving the path

Mkdir –p /users/kvirani/lhl/cohort6/class/wld2

Q4

|  |
| --- |
| -cat /usr/share/dict/words | grep "ous$" |
|  | lists the words from /usr/share/dict/words ending in 'ous' |
|  | --cat /usr/share/dict/words | grep "ous$" | wc -l |
|  | is counting the number of lines a word ending in 'ous' occurs in /usr/share/dict/words |

Q5

|  |
| --- |
| ---What is the dollar sign ($) being used for in the grep commands above? What happens if we don't specify that symbol? |
|  | Specifying ends with; elsewise they would look for those characters anywhere in a string |
|  |  |
|  | --This argument for grep ("ous$"), contained within the double quotes, is more than just a string. What is this type of expression called? |
|  | Regular Expressio, regx, regexp |
|  | the dollar sign represents the end of the line not each word |

Q7

What does the > symbol do in the Unix CLI?

Similar to the pipe | but instead of another command on the right side you give it a path

High level concepts

Playing with irb – interactive ruby in terminal

Write irb in terminal

This will write ruby code that will execute immediately

Use cmd ell and it will clear (cmd L)

The “ require ‘pry’ 🡪 we are asking it to get someone elses code.

If it returns true then good, if not then theres a prob

This is used for debugging

Example

Def add(x,y)

Result = x – y

Binding.pry

End

Can write result and get the answer

Write x gives x value

Y gives y value

Write exit to get out of there

What does pry do?

* To get someone elses gem (someone elses code). You write – sude gem install pry-coolline (google pry-coolline github)
* Then asks for password.
* Hallelujah its installed- it now has live syntax highlighting
* After this you do:
  + Def add(x,y)
    - Result = x + y
    - Binding.pry
  + End
* Add (20,20)
* To get the result just write result, or x for x y for y etc
* Exit to leave
* If you want to open something in sublime in terminal do .subl – (google it)

Notes

* When you use puts “Hello” 🡪 what it actually does is puts “hello”.to\_s
* so it turns our text as a string so it can be output like that.
* But if we use p instead of puts 🡪 it gives us the actual type of what it is. So if you write p “Hello”
* It will print “Hello” in quotes. So it prints it in a new line but also shows what is it.
* So use p when your trying to be debugging. Try to use all the time.
* The gets.chomp defaults your input to a string. So if your asking for age and write p age. It will give it in quotes. If we want to get it as an integer we hve to do age = gets.chomp.to\_i

So try

Puts “hey how old are you

Age = gets.chomp

If age > 10

Puts “sort of an adult”

Else

Puts “definetly an adult

End

It gives an error because, gets.chomp gives a string not integer. So do instead (if age.to\_i > 10)

**Ruby always returns the answer to the last line in the code, whether you need it or not. So we don’t expicitly have to write return.**

Binding.pry to debug

Puts name + “ is “ + age.to\_s + “ years old!” instead do 🡪

Puts “#{name} is #{age} years old!”

Hashes – assigns value –

In an array think theres a million different data – name, address, postal .. says to find my number.. how? Have to map the data with a hash and assign values

Keys hold values. Example

Person\_1 = {

Name: “sara”

Age: 23

} 🡪 here name is key sara is value, age is key, value is 23 .

people = [person\_1, person\_2]

people.each do |person|

puts person[:name]

end

prints names.

**Intro To Programming with Ruby**

Today we did:

* A brief lecture overview
* We took up homework from the previous day questions
* Gave an introduction to IRB for just a few brief moments
* Elaborated on IRB's useful functionality but lacking features
  + In order to do debugging we need to require other libraries
  + Show pry's Github page and go over some of their README [https://www.google.ca/search?q=github+pry&oq=github+pry&aqs=chrome..69i57j0j69i60l3j0.1382j0j4&sourceid=chrome&ie=UTF-8](http://mandrillapp.com/track/click/30244704/www.google.ca?p=eyJzIjoiYnM0eHdFQXhFVFRqUjh4THlFZ2FPWG5tY1NZIiwidiI6MSwicCI6IntcInVcIjozMDI0NDcwNCxcInZcIjoxLFwidXJsXCI6XCJodHRwczpcXFwvXFxcL3d3dy5nb29nbGUuY2FcXFwvc2VhcmNoP3E9Z2l0aHViK3ByeSZvcT1naXRodWIrcHJ5JmFxcz1jaHJvbWUuLjY5aTU3ajBqNjlpNjBsM2owLjEzODJqMGo0JnNvdXJjZWlkPWNocm9tZSZpZT1VVEYtOFwiLFwiaWRcIjpcIjdkNmU0YTFhOTA0MTRjZWRiNjEwODg2OGNlZjMxYTBmXCIsXCJ1cmxfaWRzXCI6W1wiOTMzYWE5NjYzODU5N2UxMzE1MTQ4MmI5YWYyYjc1YmUzNjhiZGEyNlwiXX0ifQ" \t "_blank)
  + We did a gem list and show the pry gem on your machine
  + We required pry in an IRB session to do some debugging.
  + We spoke about how its lifeless and boring
  + Give an introduction to the pry REPL.
    - Hurray for syntax highlighting
    - Hurray for built in debugging
    - But wait, why does the syntax highlighting still suck?
    - Using another gem like pry-coolline "Live syntax highlighter to save the day!"[https://github.com/pry/pry-coolline](http://mandrillapp.com/track/click/30244704/github.com?p=eyJzIjoiSUxia2NiWFFYUGtJYUh1M0ZCZGdjMDE2SDU0IiwidiI6MSwicCI6IntcInVcIjozMDI0NDcwNCxcInZcIjoxLFwidXJsXCI6XCJodHRwczpcXFwvXFxcL2dpdGh1Yi5jb21cXFwvcHJ5XFxcL3ByeS1jb29sbGluZVwiLFwiaWRcIjpcIjdkNmU0YTFhOTA0MTRjZWRiNjEwODg2OGNlZjMxYTBmXCIsXCJ1cmxfaWRzXCI6W1wiOGI1MmI3NGVjZmYzMGU5NjZjYjdhNTgyODlkZTU0YmZlZGVkODQ5Y1wiXX0ifQ" \t "_blank)
  + Jumped into the exersises folder and walk through the code from 01 - 10\*.rb as a group exercise.

Link to in class code: [https://drive.google.com/folderview?id=0B5ML1Pmi6uURam9rMXFkZGI5akE&usp=sharing](http://mandrillapp.com/track/click/30244704/drive.google.com?p=eyJzIjoiUGw5UVB5VlY3bl9UWWVVQ2xiREx5N3JJTWFZIiwidiI6MSwicCI6IntcInVcIjozMDI0NDcwNCxcInZcIjoxLFwidXJsXCI6XCJodHRwczpcXFwvXFxcL2RyaXZlLmdvb2dsZS5jb21cXFwvZm9sZGVydmlldz9pZD0wQjVNTDFQbWk2dVVSYW05ck1YRmtaR0k1YWtFJnVzcD1zaGFyaW5nXCIsXCJpZFwiOlwiN2Q2ZTRhMWE5MDQxNGNlZGI2MTA4ODY4Y2VmMzFhMGZcIixcInVybF9pZHNcIjpbXCI4MmUyNzczNTI1NDM1YmJhNWRmM2UyOTMwYTdiZWMwYmE4NjQ5MTJlXCJdfSJ9" \t "_blank)[http://compass.lighthouselabs.ca/days/w1d2/activities/54](http://mandrillapp.com/track/click/30244704/compass.lighthouselabs.ca?p=eyJzIjoidzkyMkhLZEVnbS1wUmNRdTVUalprWElqekd3IiwidiI6MSwicCI6IntcInVcIjozMDI0NDcwNCxcInZcIjoxLFwidXJsXCI6XCJodHRwOlxcXC9cXFwvY29tcGFzcy5saWdodGhvdXNlbGFicy5jYVxcXC9kYXlzXFxcL3cxZDJcXFwvYWN0aXZpdGllc1xcXC81NFwiLFwiaWRcIjpcIjdkNmU0YTFhOTA0MTRjZWRiNjEwODg2OGNlZjMxYTBmXCIsXCJ1cmxfaWRzXCI6W1wiMGNiNDEzMWMxZDAyZjE2Y2YwMWFjOTg0NDhlMWFjOTI1NzAzNDJhOVwiXX0ifQ" \t "_blank)