Julian – panthers stuff… has ideas and wants to start a business.

Chantelle – blonde behind me. Accountant, but was bored.

Matty – brunette in back

Chris – salesman, glasses and beard. Has a business for B2B sales, wants to write app/website for this.

5g –

---Ctrl+C exits a program without exiting Terminal

**String** – a sequence of text enclosed in ‘ ‘ or “ “

If you put a number in quotations, it’s a string not an integer

**Integer** – Whole numbers (no decimal)

**Float** – Numbers with decimals

**Boolean** – only true or false, a variable set as true or false to get into or out of loops

**Array** – ordered list of items separated by commas. Store info to draw on later.

**Hash** – unordered collection of key/value pairs… examples: cars = {‘Ford’ => ‘Mustang’, ‘Bugati’,

Strings and arrays have index… starts with 0 (not 1).

Methods – pre-set code, you can set your own and some of it is built in.

Do “.whatever”

String.methods gives you a list of built-in methods (must capitalize “S” so it recognizes it as a class)

Variable – always use lowercase

this\_is\_snake\_case

ThisIsCamelCase

Pwd – print your working directory

Cd – change directory, such as “cd desktop/tts”

Ls – lists everything in the folder

Cd .. – back a folder

Touch – create a new file, include .rb extension

.chomp – for “puts” without the line break

**Concatenation vs Interpolation**

Concatenate – uses + sign to add strings and variables together

*puts "Hello " + response + ", how are you?"*

Interpolate – Uses symbols #{ }

*puts "Hello #{response}, how are you"*

**Data Type Conversion**

.to\_s – converts to string

.to\_i – to integer (gets rid of decimals if they’re there)

.to\_f – to float

my.number = “1”

my.number.to\_i

**If/Else**

Elsif continued if you need more than one

**Ruby Comparison Symbols**

== equals

!= does not equal

< less than

<= less than or equal to

>

>=

&& is AND

|| is OR

Loops

MAKE SURE YOU PUT AN END or you get an infinite loop and crash the terminal

.times = do something some value of times:

*3.times do*

*puts “Beetlejuice”*

*end*

Don’t ever use quotations, it has to be an integer

Until = a variable is incremented and decremented and an action is performed until a condition is true.

*Num = 1*

*Until num == 10*

*Puts num*

*Num += 1*

*end*

+= adds by 1

While

Num = 1

While num < 10

Puts num

Num += 1

End

Arrays – a collective data type

Housed within brackets

Array = [“yes”, “no” etc

Array.shuffle

Array.shuffle! (! Permanently changes it)

Arr = [“Atlanta”,”Augusta”,”Columbus”]

Arr.first

“Atlanta”

arr.pop

arr.push(“Vidalia”) – lets you add to an array

arr<<”Vidalia” – Same as above

arr.reverse – reverses order of array

besides push and pop, none are permanent

Hashes are similar to arrays, but arrays are stored by index, hashes by “key”/”Value” pairs

*My\_hash = {“key” => “value”}*

*My\_other\_hash = {“Name” => “Aaron”, “Role” => “Instructor”, “Age” => 34}*

Each key and value are separated by a hash-rocket

A hash is contained with curly brackets

.each loop lopos through a collection of data

*my\_array = [1,2,4,5]*

*my\_array.each do |x|*

*puts x*

*end*

x can be anything you want, just make sure you use it inside the code block

**hash.each**

*state1 = {“name”=>”Georgia”,”Capital”=>”Atlanta”,”Population”=>10097343,”Area”=>59425}*

*state1.each do |key, value|*

*puts “#{key}: #{value}”*

*end*

*$ ruby state\_hash.rb*

*Name: Georgia*

*Capital: Atlanta*

*Population: 10097343*

*Area: 59425*

Codequizzes.com/ruby

Do quizzes 1, 2, and 5

Don’t do #1 on Level 3 homework