BEWD 13

LESSON 1 - INTRO

WHERE IS RUBY?

Basecamp®

goodreads

API LIBRARIES IN RUBY



VIRTUALLY EVERY MAJOR TECH COMPANY HAS A RUBY API LIBRARY

AGENDA

1 - REVIEW THE SYLLABUS

2 - GITHUB FETCH OF THE LATEST FROM THE LESSON REPO

3 - TWO CODE ALONGS!!!

SYLLABUS

HTTPS://GITHUB.COM/GA-STUDENTS/BEWD_SF_13/BLOB/MASTER/SYLLABUS.MD

GIT TIME!

```
git pull upstream master
 - pulls the latest version from the `mother_ship`

git push origin +master
 - pushes the latest version from the `mother_ship` to your forked version

git branch
 - displays all branches.
 - The branch you are working will look like this `* master`


git branch lesson_one
 - creates a new branch called lesson_one

git checkout lesson_one
 - changes your current branch to the `lesson_one` branch
```

KEYS TO SUCCESS

- ONE BRICK AT TIME
- DEBUG WITH PRY EVERY TIME
- CODE PROLIFICALLY

FIRST CHALLENGE!

CODE ALONG!!

AWESOME!!

WRITE A METHOD THAT PRINTS THE FOLLOWING:

- "AWE" - IF THE NUMBER IS DIVISIBLE BY 3
- "SOME" - IF THE NUMBER IS DIVISIBLE BY 5
- "AWESOME" - IF THE NUMBER IS DIVISIBLE BY 3 AND 5
- "THIS NUMBER IS (THE NUMBER). IT IS NOT COMPLETELY AWESOME"
  - IF THE NUMBER DOES NOT MEET ANY OTHER CONDITION

RUBY DOCS FOR THE STRING CLASS

HTTP://RUBY-DOC.ORG/CORE-2.3.0/INTEGER.HTML

LET'S BUILD IT!

CODE ALONG!!

AWESOME!

```ruby
require 'pry'

def awesome(number)
  if number % 3 == 0 && number % 5 == 0
    "Awesome"
  elsif number % 3 == 0
    "Awe"
  elsif number % 5 == 0
    "Some"
  else
    number
  end
end

def awesome_seeker(high_value)
  1.upto(high_value) do |number|
    puts awesome(number)
  end
end


awesome_seeker(100)
```

SECOND CHALLENGE!

CODE ALONG!!

REVERSE IT!

- WRITE OUR OWN 'REVERSE' METHOD
-USE IT TO DETERMINE IF A WORD IS A PALINDROME

RUBY DOCS FOR THE STRING CLASS

HTTP://RUBY-DOC.ORG/CORE-2.3.0/STRING.HTML

LET'S BUILD IT!

CODE ALONG

REVERSE IT!

```ruby
def my_reverse(string)
  char = string.downcase.chars
  word = ""
  until char.length == 0
    word << char.pop
  end
  word.capitalize
end

def is_palindrome?(word)
  if word.downcase == my_reverse(word).downcase
    "Yay! A Palindrome!"
  else
    "Shucks, Not A Palindrome"
  end
end

#####
puts "Please provide a word \n"
word = gets.chomp

puts my_reverse(word)
puts is_palindrome?(word)
```

HOMEWORK

1 - WGR - READ CHAPTER 1 THRU 5

2 - BONUS! COMMAND LINE MURDER MYSTERY

- CLONE -> HTTPS://GITHUB.COM/VELTMAN/CLMYSTERY
  - START BY READING THE INSTRUCTIONS.
  - IT'S OKAY TO READ THE CHEAT SHEET