to Github :

**…or create a new repository on the command line**

echo "# Ruby-Programming" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/skmorshed75/Ruby-Programming.git

git push -u origin main

### …or push an existing repository from the command line

git remote add origin https://github.com/skmorshed75/Ruby-Programming.git

git branch -M main

git push -u origin main

Ruby Programming Learning (freeCodeCamp.org)

- Windows Installation

- https://rubyinstaller.org/



- select all 3 options

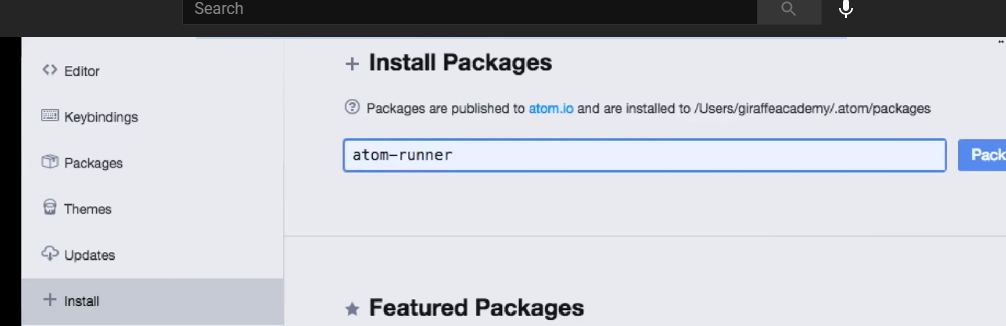
- goto cmd

- ruby -v (check ruby version)

- Hello World Setup (text editor)

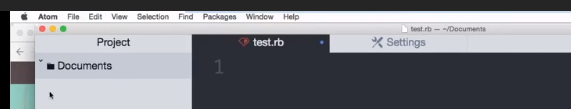
- Text Editor (atom.io)

- Preferences -> settings -> install -> atom-runner

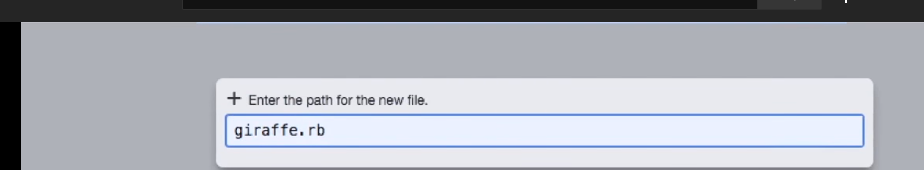


- File -> add Project Folder -> select the folder you want to keep as project -> open

- the folder will show in left windows



- create new file



- .rb extension is required to create ruby file

- Open the file - write print "Hello World"

print "Hello World" - show text in same line

puts "Hello World" - show text in new line

variable should be descriptive like character\_name, character\_age

Example variables :

character\_name = "Jeal"

character\_age = "40"

puts ("There once was a man named "+character\_name)

puts ("he was " + character\_age + " years old")

puts ("he really liked the name "+character\_name)

puts ("but didn't like being " + character\_age)

Data Types :

name = "Mike" #String

age = 70 #integer

gpa = 3.2 #float

ismale = true #boolean

istall = false

flaws = nil #null

Class 5 : Uses of Strings

puts "Giraffe Academy"

puts "\"Giraffe\" Academy" #Insert Quotation

puts "Giraffe\nAcademy" #create a new line

phrase = "Giraffe Academy"

puts phrase.upcase() # Uppercase

puts phrase.downcase() #Lowercase

phrase = " Giraffe Academy " #Space Trimming

puts phrase.strip()

phrase = "Giraffe Academy" #Count no of Characters

puts phrase.length()

phrase = "Giraffe Academy" #Search keyword

puts phrase.include? "Academy"

phrase = "Giraffe Academy" #Show characters

puts phrase[0] #show a character

puts phrase[0,3] #show a range of characters

puts phrase.index("G") #show index

puts phrase.index("r") #show index

Class 6 - Math and Numbers

puts 5+9

puts 5/10

puts 5\*10

puts 5-10

puts 10 % 3 #remainder

num = 10

puts ("My fav number is " + num.to\_s) #to string

num = -20

puts num.abs() #Positive number

num = 20.3154

puts num.round() #20 nearest number

num = 20.11

puts num.ceil() #21 maximum number

num = 20.21

puts num.floor() #20 minimum number

puts Math.sqrt(36) #6.0

puts Math.log(1) #0.0

Class - 7 : Getting User Input :

Preferences -> Settings -> Install -> platformio-ide-terminal

chomp() - to remove new line -- gets.chomp()

Class - 8 : Calculator :

puts "Enter a number :"

#num1 = gets.chomp()

num1 = gets.chomp().to\_f

puts "Enter another number :"

#num2 = gets.chomp()

num2 = gets.chomp().to\_f

#puts num1.to\_i + num2.to\_i

#puts num1.to\_f + num2.to\_f #floating point number ie. 6.44

puts num1 + num2 #floating point number ie. 6.44

Class - 9 : Mad Libs Game

Class - 10 : Array :

friends = Array["Kevin", "Karen", "Oscar", "Andy"]

puts friends # show all

puts friends[2] # show index no. 2 Oscar

puts friends[-1] # show index from last

puts friends.length # show the length of Array

puts friends.include? "Karen" # Search a keyword,show true if found

puts friends.reverse() # show from last

puts friends.sort() # display ascending order

friends = Array.new

friends[0] = "Michael"

friends[5] = "Nick"

puts friends

**Class - 11 : Hashes :**

# Key, Value

states = {

"Pennsilvania" => "PA", # or we can write :Pennsilvania => "PA"

#:Pennsilvania => "PA",

1 => "One",

"New York" => "NY",

"Michigan" => "MI"

}

#puts states

#puts states["Michigan"]

#puts states[:Pennsilvania]

puts states[1]

**Class - 12 : Method :**

method block :

def sayhi

end

**Class - 13 : Return Statement :**

xx

**Class - 14 : If Statement :**

xx

**Class - 15 : Building a better calculator :**

xx

**Class - 16 : Case Expression :**

xx

**Class - 17 : While Loops :**

xx

**Class - 18 : Building a Guessing Game :**

xx

**Class - 19 : For Loops :**

xx

**Class - 20 : Exponent Method :**

xx

**Class - 21 : Comments :**

single line comment : #

multiline comments : =begin COMMENTS =end

**Class - 22 : Reading Files :**

- File.open("test.txt", "r") do |file| : Open file

- file = File.open("test.txt", "r") : Open a file with file variable

- puts file.read() or puts file.readlines() : Read file

- puts file.read().include? "Jim" #search Jim text : Search a string

- puts file.readline() #print first line : Print first line of the file

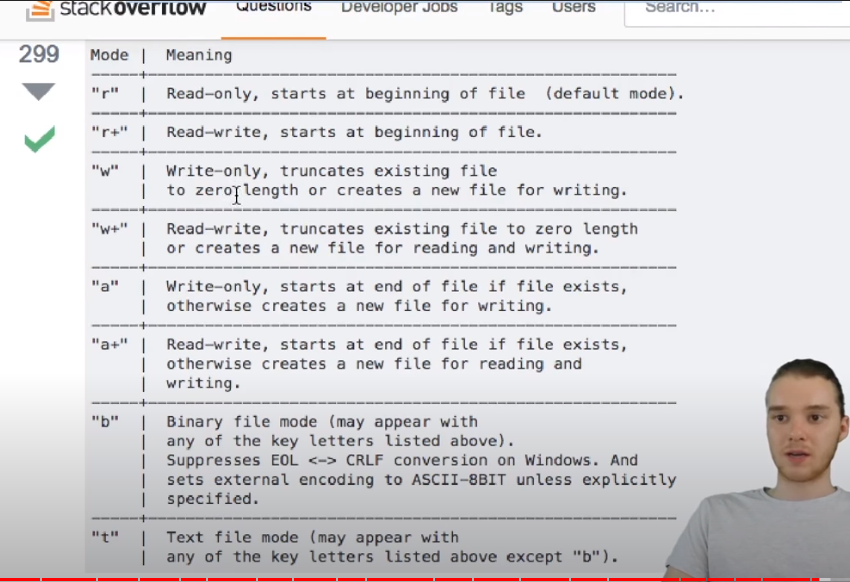
- puts file.readchar() #print first character of a line : Print first character of a line

for line in file.readlines()

puts line #read all lines

end

**Class - 23 : Writing Files :**



- File.open("test.txt", "a") do |file| : Open a file in read mode

- file.write("\nOscar, ", "Acconting") : Insert data into file

- File.open("index.html", "w") do |file| : Open a file in write mode with newly creation

- file.write("<h1>Hello World! </h1>") : Insert data into file

- File.open("test.txt", "r+") do |file| : Open a file in read / write mode

- file.readline() : write data after first line

- file.write("Hi") : Insert data

**Class - 24 : Error Handling :**

begin

puts lucky\_nums["dog"]

num = 10/0

rescue ZeroDivisionError

puts "Division by 0 Error"

rescue TypeError => e

puts e

end

**Class - 25 : Classes & Objects :**

- Classes is just like a custom data type

class Book

attr\_accessor :title, :author, :pages

end

book1 = Book.new()

book1.title = "Harry Potter"

book1.author = "JK Rowling"

book1.pages = 499

puts book1.pages

**Class - 26 : Initialize Method :**

- xxx

**Class - 27 : Object Methods :**

-

**Class - 28 : Quiz :**

-

**Class - 29 : Inheritance :**

-

**Class - 30 : Modules :**

-

**Class - 31 : IRB - Interactive Ruby :**

- cmd

- irb -v : to see whether irb installed and irb version number

