Terminal Trivial

Walkthrough >>>

- Opens with ascii title graphic prompting user to press any key to continue

- Username is entered and validated and then option to change it or exit the app is given

- Default setting is 6 general knowledge multiple choice questions on easy setting.

- Quiz runs and then score is printed out in rainbow ascii graphic

- Printout of results, including correct and incorrect. User prompted to press any key and results disappear before asking to play again or exit.

```
 _____              (_)        _____ _      (_)       (_)          _
|__   __|             (_)       |_   _(_)     (_)       (_)         | |
   | | ___ _ __ _ __ ___ _ _ __   __ _| |   | |_ __ ___   _  __ _| |
   | |/ _ \ '__| '_ ` _ \| | '_ \ / _` | |   | | '__| \ \ / / |/ _` | |
   | |  __/ |  | | | | | | | | | | (_| | |   | | |  | |\ V /| | (_| | |
   |_|\___|_|  |_| |_| |_|_|_| |_|\__,_|_|   |_|_|  |_| \_/ |_|\__,_|_| (_)
```

You entered: LUKE

Is this correct? (Use ↑/↓ arrow keys, press Enter to select)
‣ Yes
  No
  Exit

Which type of rock is created by intense heat AND pressure?
(Use ↑/↓ arrow keys, press Enter to select)
‣ Metamorphic
  Sedimentary
  Igneous
  Diamond

You scored:

2 out of 6

What is the largest organ of the human body?

Skin :-)

Who is the main protagonist of Dead Space?

Commander Shepard => Isaac Clarke

On the show "Rick and Morty", in episode "Total Rickall", who was a parasite?

Pencilvester :-)

What was the first feature-length computer-animated movie?

Tron => Toy Story

The "British Invasion" was a cultural phenomenon in music where British boy bands became popular in the USA in what decade?

30's => 60's

Working with the Open Trivia DB API

```ruby
require 'httparty'
require 'htmlentities'

class QuestionBank
  attr_reader :prompts, :correct_answers, :incorrect_answers, :q_amount, :difficulty
  attr_accessor :q_index

  def initialize(url = nil)

    #if command line argument not given, use default url
    if !url
      url = "https://opentdb.com/api.php?amount=6&difficulty=easy&type=multiple"
    end

    #error handling if internet not connected
    begin
      response = HTTParty.get(url)
    rescue SocketError
      puts "SocketError! Check internet connection!"
    end
```

- HTTParty to simplify API call

- Error handling if issue with internet

- Listed is the default url that asks for 6 random multiple choice questions from any category

Decoding HTML Entities & populating quiz arrays

```ruby
#create difficulty instance variable
@difficulty = response.parsed_response["results"][0]["difficulty"].upcase

#create instance array of question prompts
@prompts =
  response.parsed_response["results"].map {
    |index| index["question"] }

#create instance array of correct answers
@correct_answers = response.parsed_response["results"].map {
  |index| index["correct_answer"] }

#create instance array of incorrect answers
@incorrect_answers = response.parsed_response["results"].map {
  |index| index["incorrect_answers"] }

  @q_amount = @prompts.length
  @q_index = 0

#remove HTMLEntities from array variables
html = HTMLEntities.new

@prompts = @prompts.map {|item| html.decode(item)}
@correct_answers = @correct_answers.map {|item| html.decode(item)}
@incorrect_answers = @incorrect_answers.map {|row| row.map {|item| html.decode(item)}}
```

- Created 2 arrays and 1 nested array of trivia data

- Used HTMLEntities gem to easily decode the &quot? Etc from the data.

Using ARGVs

- Provide an alternative API url asking for 15 HARD questions when '-g' flag is used.

```
#set difficulty to 'GOD' level using ARGV
def process_argv(option)
  case option
  when "-g"
    @url = "https://opentdb.com/api.php?amount=&difficulty=hard&type=multiple"
  end
end
```

TriviaGame Class

```ruby
class TriviaGame

  def initialize
    setup
  end

  def reset
    setup
  end

  private

  def setup

    @god_mode = ARGV[0]

    @username = ""    #set @username variable

    @cursor = TTY::Cursor   #set @cursor variable

    welcome #display welcome message and title graphic

    username #input and check username

    quiz  #run quiz questions

    play_again #gives user to exit or play again

  end
end
```

- Create an instance variable for the ARGV in a setup method in the main TriviaGame class.

- Wrapping the game in a TriviaGame class like this was the solution I found to re-initialize the game instance.

Any Questions?