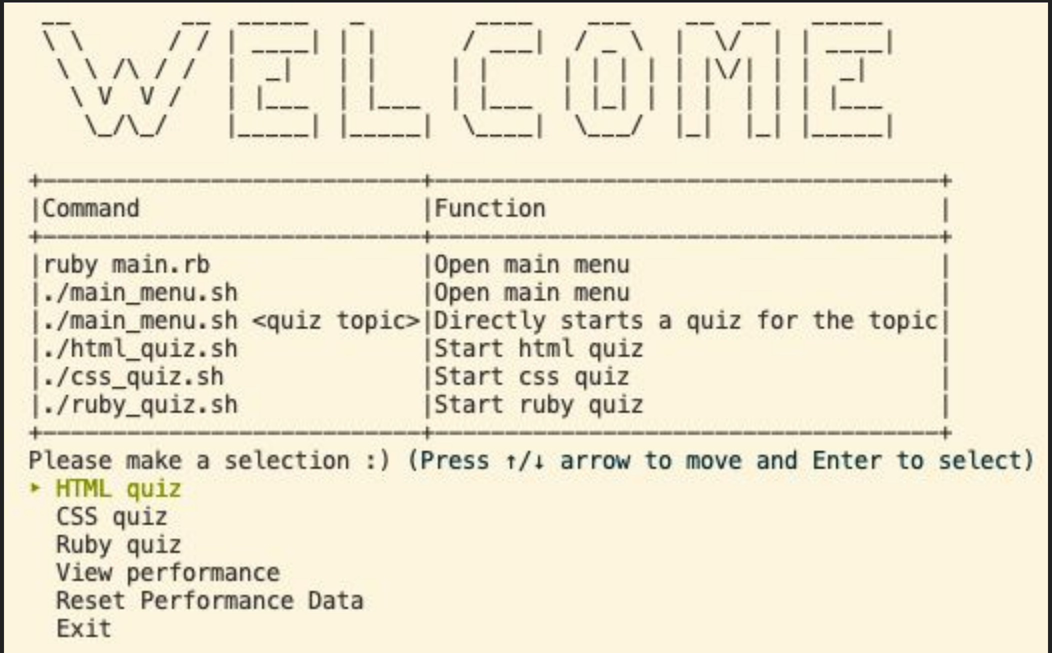
Multiple Choice Quiz App

Riki Fujihara

Main Features

Main Menu:

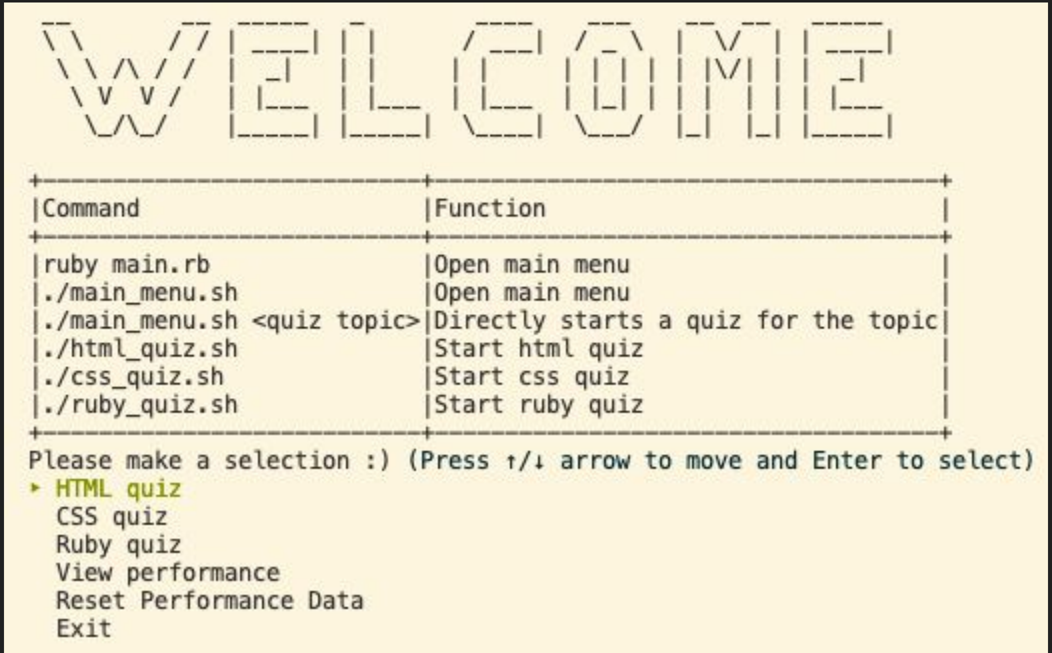
- List of available commands
- Quiz Selection
- Viewing Performance
- Resetting Performance



```
 __        __  _____   _        ____    ___    __  __   _____
 \ \      / / | ____| | |      / ___|  / _ \  |  \/  | | ____|
  \ \ /\ / /  |  _|   | |     | |     | | | | | |\/| | |  _|
   \ V  V /   | |___  | |___  | |___  | |_| | | |  | | | |___
    \_/\_/    |_____| |_____|  \____|  \___/  |_|  |_| |_____|

+----------------------------+-------------------------------------+
|Command                     |Function                             |
+----------------------------+-------------------------------------+
|ruby main.rb                |Open main menu                       |
|./main_menu.sh              |Open main menu                       |
|./main_menu.sh <quiz topic> |Directly starts a quiz for the topic|
|./html_quiz.sh              |Start html quiz                      |
|./css_quiz.sh               |Start css quiz                       |
|./ruby_quiz.sh              |Start ruby quiz                      |
+----------------------------+-------------------------------------+
Please make a selection :) (Press ↑/↓ arrow to move and Enter to select)
‣ HTML quiz
  CSS quiz
  Ruby quiz
  View performance
  Reset Performance Data
  Exit
```

List of available commands

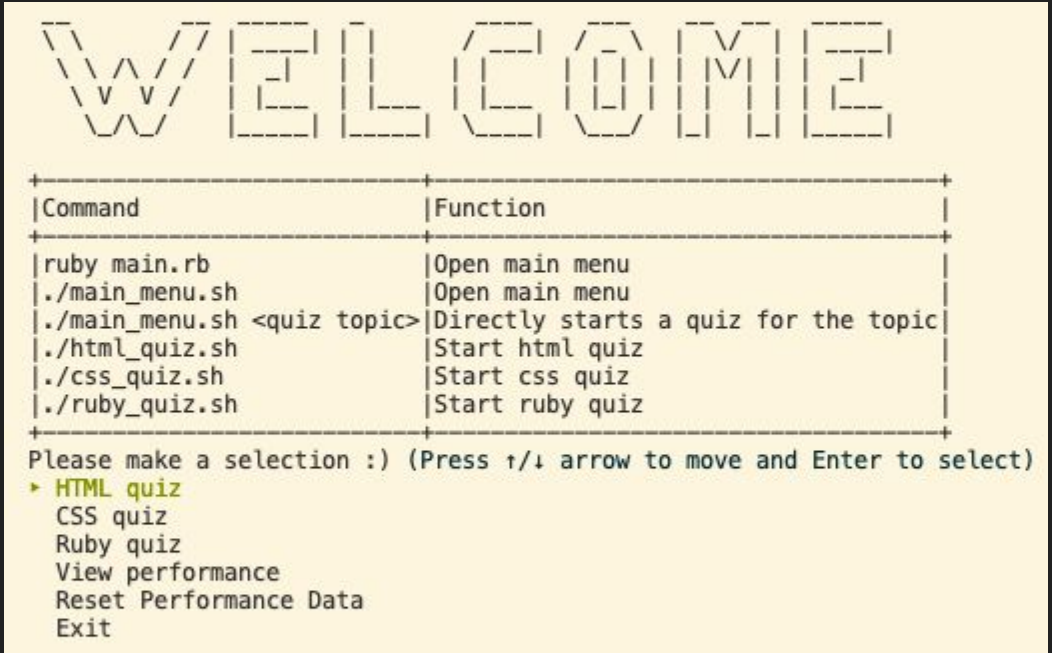
- Various shell scripts and their functions



```
__        _______ _     ____ ___  __  __ _____
\ \      / / ____| |   / ___/ _ \|  \/  | ____|
 \ \ /\ / /|  _| | |  | |  | | | | |\/| |  _|
  \ V  V / | |___| |__| |__| |_| | |  | | |___
   \_/\_/  |_____|_____\____\___/|_|  |_|_____|

+---------------------------+------------------------------------+
|Command                    |Function                            |
+---------------------------+------------------------------------+
|ruby main.rb               |Open main menu                      |
|./main_menu.sh             |Open main menu                      |
|./main_menu.sh <quiz topic>|Directly starts a quiz for the topic|
|./html_quiz.sh             |Start html quiz                     |
|./css_quiz.sh              |Start css quiz                      |
|./ruby_quiz.sh             |Start ruby quiz                     |
+---------------------------+------------------------------------+
Please make a selection :) (Press ↑/↓ arrow to move and Enter to select)
‣ HTML quiz
  CSS quiz
  Ruby quiz
  View performance
  Reset Performance Data
  Exit
```

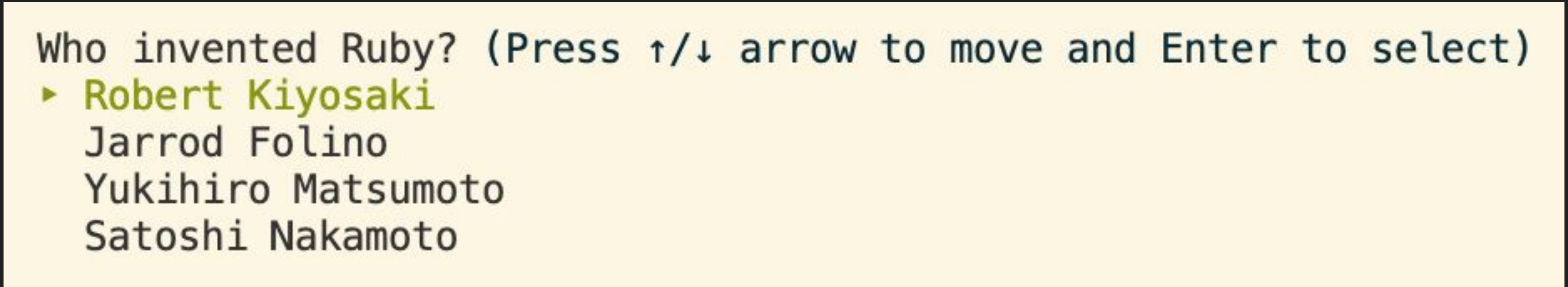
Quiz Selection

- HTML quiz
- CSS quiz
- Ruby quiz



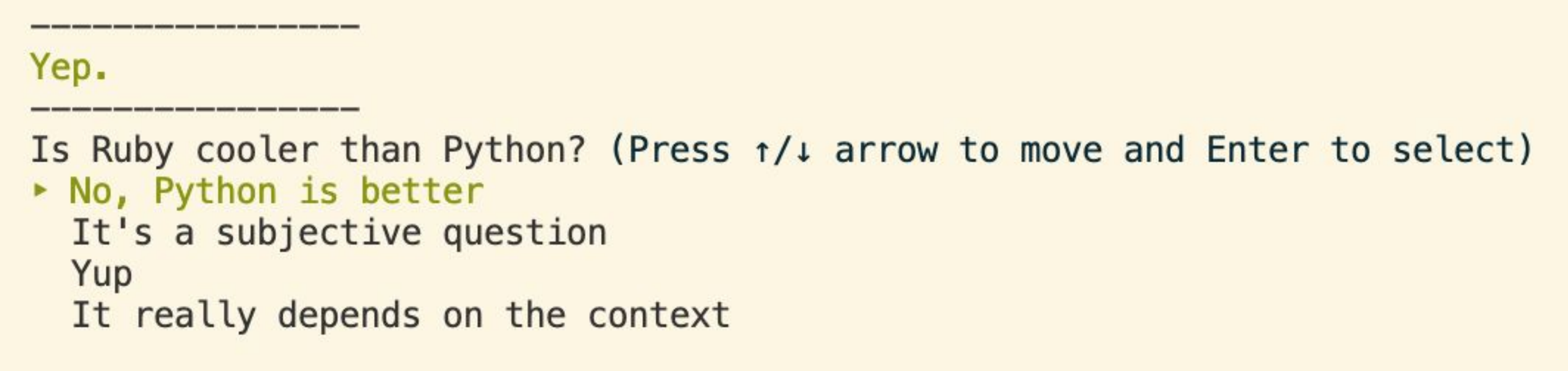
```
 __          ________ _      _____ ____  __  __ ______
 \ \        / /  ____| |    / ____/ __ \|  \/  |  ____|
  \ \  /\  / /| |__  | |   | |   | |  | | \  / | |__
   \ \/  \/ / |  __| | |   | |   | |  | | |\/| |  __|
    \  /\  /  | |____| |___| |___| |__| | |  | | |____
     \/  \/   |______|______\_____\____/|_|  |_|______|

+-----------------------------+------------------------------------+
|Command                      |Function                            |
+-----------------------------+------------------------------------+
|ruby main.rb                 |Open main menu                      |
|./main_menu.sh               |Open main menu                      |
|./main_menu.sh <quiz topic>  |Directly starts a quiz for the topic|
|./html_quiz.sh               |Start html quiz                     |
|./css_quiz.sh                |Start css quiz                      |
|./ruby_quiz.sh               |Start ruby quiz                     |
+-----------------------------+------------------------------------+
Please make a selection :) (Press ↑/↓ arrow to move and Enter to select)
‣ HTML quiz
  CSS quiz
  Ruby quiz
  View performance
  Reset Performance Data
  Exit
```

Quiz Display



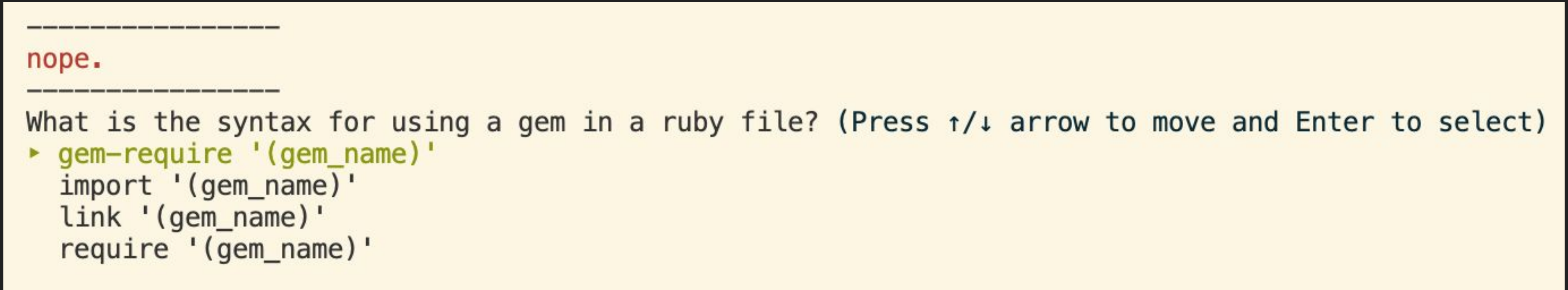
```
Who invented Ruby? (Press ↑/↓ arrow to move and Enter to select)
‣ Robert Kiyosaki
  Jarrod Folino
  Yukihiro Matsumoto
  Satoshi Nakamoto
```

Quiz Display



```
----------------
Yep.
----------------
Is Ruby cooler than Python? (Press ↑/↓ arrow to move and Enter to select)
‣ No, Python is better
  It's a subjective question
  Yup
  It really depends on the context
```

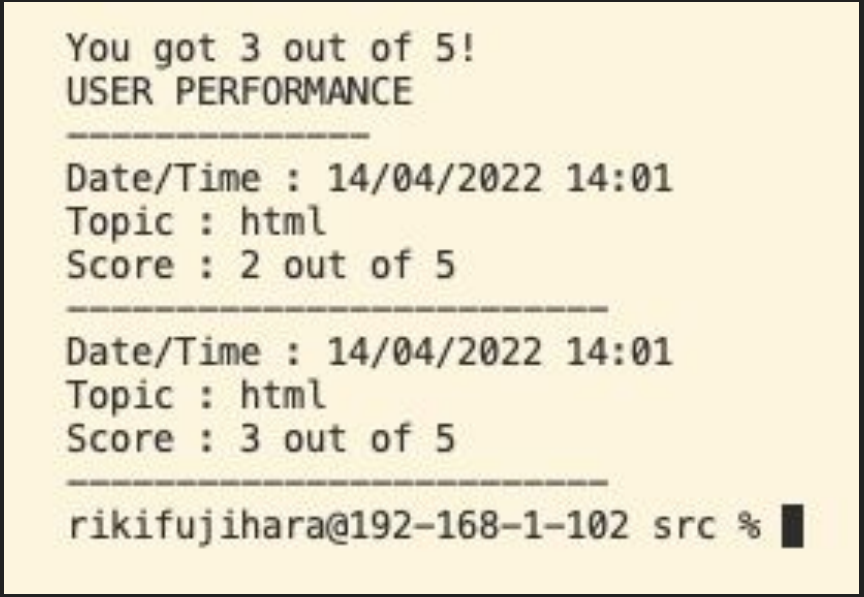
Quiz Display



```
----------------
nope.
----------------
What is the syntax for using a gem in a ruby file? (Press ↑/↓ arrow to move and Enter to select)
‣ gem-require '(gem_name)'
  import '(gem_name)'
  link '(gem_name)'
  require '(gem_name)'
```

Viewing performance

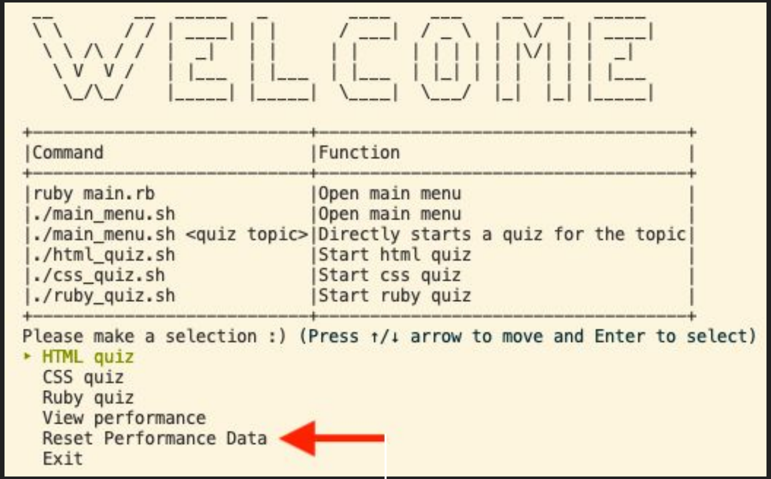
- Score for quiz
- Date and time of when quiz was completed
- Topic of quiz
- Score out of 5



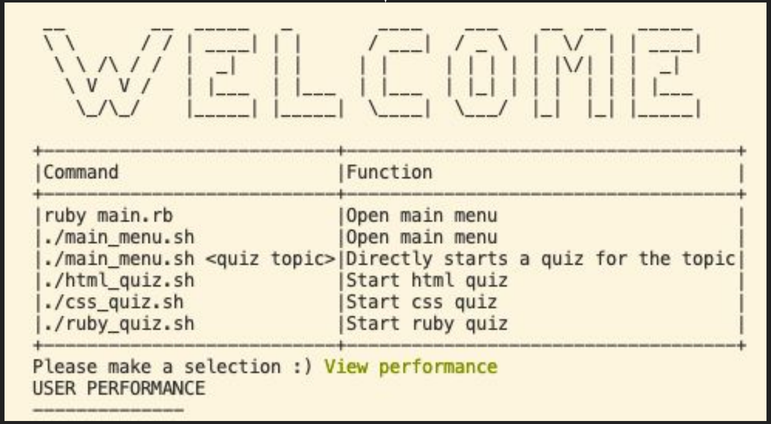
```
You got 3 out of 5!
USER PERFORMANCE
--------------
Date/Time : 14/04/2022 14:01
Topic : html
Score : 2 out of 5
-------------------------
Date/Time : 14/04/2022 14:01
Topic : html
Score : 3 out of 5
-------------------------
rikifujihara@192-168-1-102 src %
```

Resetting performance

- User can select ‘Reset Performance Data” from the main menu
- All performance data will be reset which you can check when viewing performance

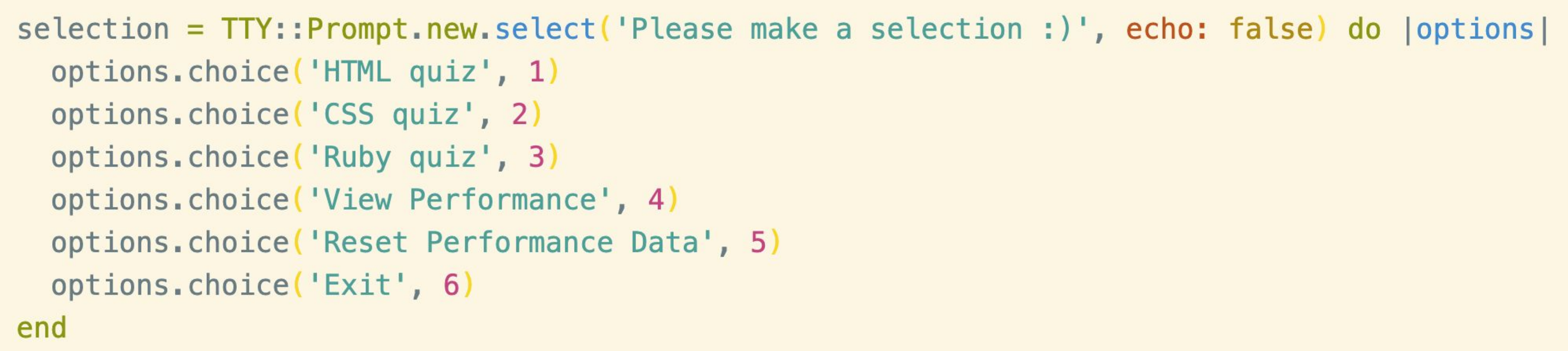


```
__        __  _____   _        ____    ___    __  __   _____ 
\ \      / / | ____| | |      / ___|  / _ \  |  \/  | | ____|
 \ \ /\ / /  |  _|   | |     | |     | | | | | |\/| | |  _|  
  \ V  V /   | |___  | |___  | |___  | |_| | | |  | | | |___ 
   \_/\_/    |_____| |_____|  \____|  \___/  |_|  |_| |_____|

+---------------------------+---------------------------------+
|Command                    |Function                         |
+---------------------------+---------------------------------+
|ruby main.rb               |Open main menu                   |
|./main_menu.sh             |Open main menu                   |
|./main_menu.sh <quiz topic>|Directly starts a quiz for the topic|
|./html_quiz.sh             |Start html quiz                  |
|./css_quiz.sh              |Start css quiz                   |
|./ruby_quiz.sh             |Start ruby quiz                  |
+---------------------------+---------------------------------+
Please make a selection :) (Press ↑/↓ arrow to move and Enter to select)
‣ HTML quiz
  CSS quiz
  Ruby quiz
  View performance
  Reset Performance Data
  Exit
```



```
__        __  _____   _        ____    ___    __  __   _____ 
\ \      / / | ____| | |      / ___|  / _ \  |  \/  | | ____|
 \ \ /\ / /  |  _|   | |     | |     | | | | | |\/| | |  _|  
  \ V  V /   | |___  | |___  | |___  | |_| | | |  | | | |___ 
   \_/\_/    |_____| |_____|  \____|  \___/  |_|  |_| |_____|

+---------------------------+---------------------------------+
|Command                    |Function                         |
+---------------------------+---------------------------------+
|ruby main.rb               |Open main menu                   |
|./main_menu.sh             |Open main menu                   |
|./main_menu.sh <quiz topic>|Directly starts a quiz for the topic|
|./html_quiz.sh             |Start html quiz                  |
|./css_quiz.sh              |Start css quiz                   |
|./ruby_quiz.sh             |Start ruby quiz                  |
+---------------------------+---------------------------------+
Please make a selection :) View performance
USER PERFORMANCE
----------------
```

Code: TTY prompt



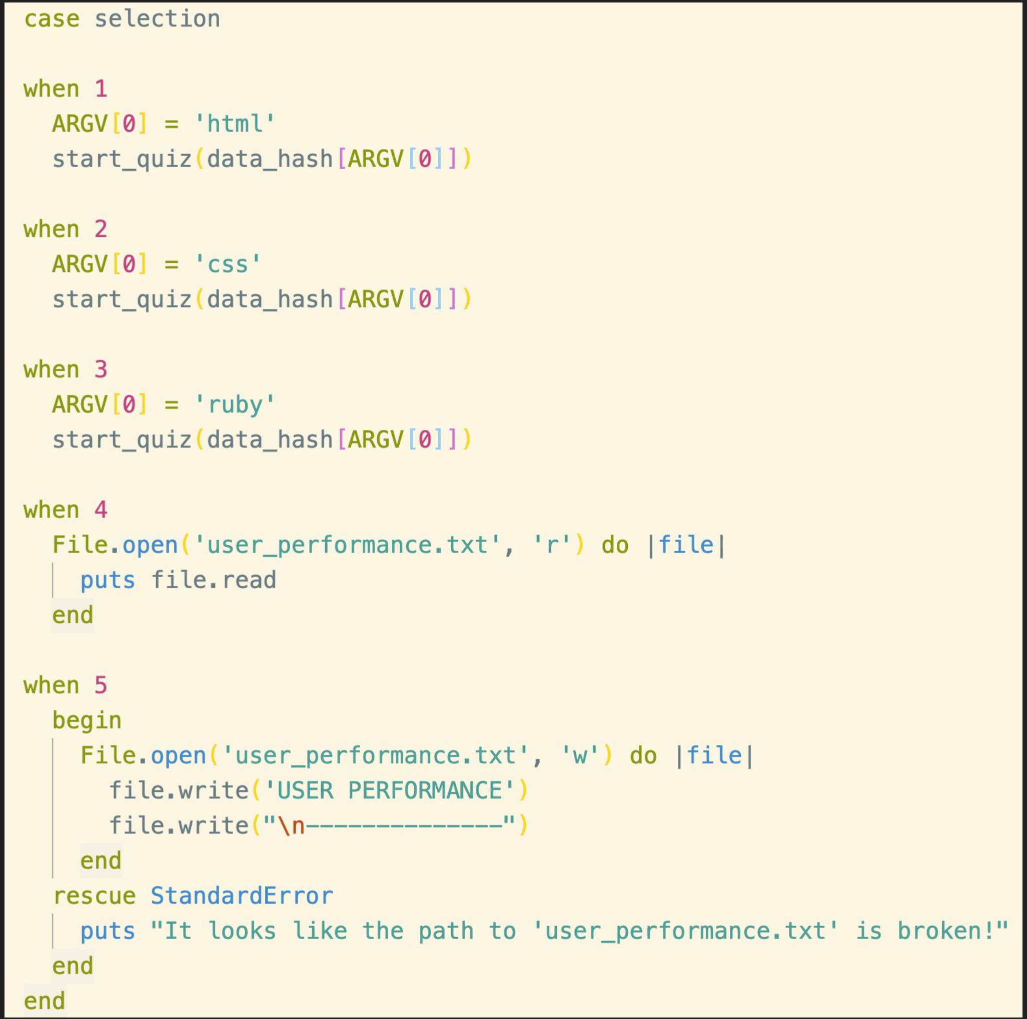
```ruby
selection = TTY::Prompt.new.select('Please make a selection :)', echo: false) do |options|
  options.choice('HTML quiz', 1)
  options.choice('CSS quiz', 2)
  options.choice('Ruby quiz', 3)
  options.choice('View Performance', 4)
  options.choice('Reset Performance Data', 5)
  options.choice('Exit', 6)
end
```

- Each selection option has displayed text and a value that will be assigned to ‘selection’ if it is selected by the user

Code:

Case statement

- Case statement used to execute different code based on user’s selection
- This is represents all of the options in the main menu

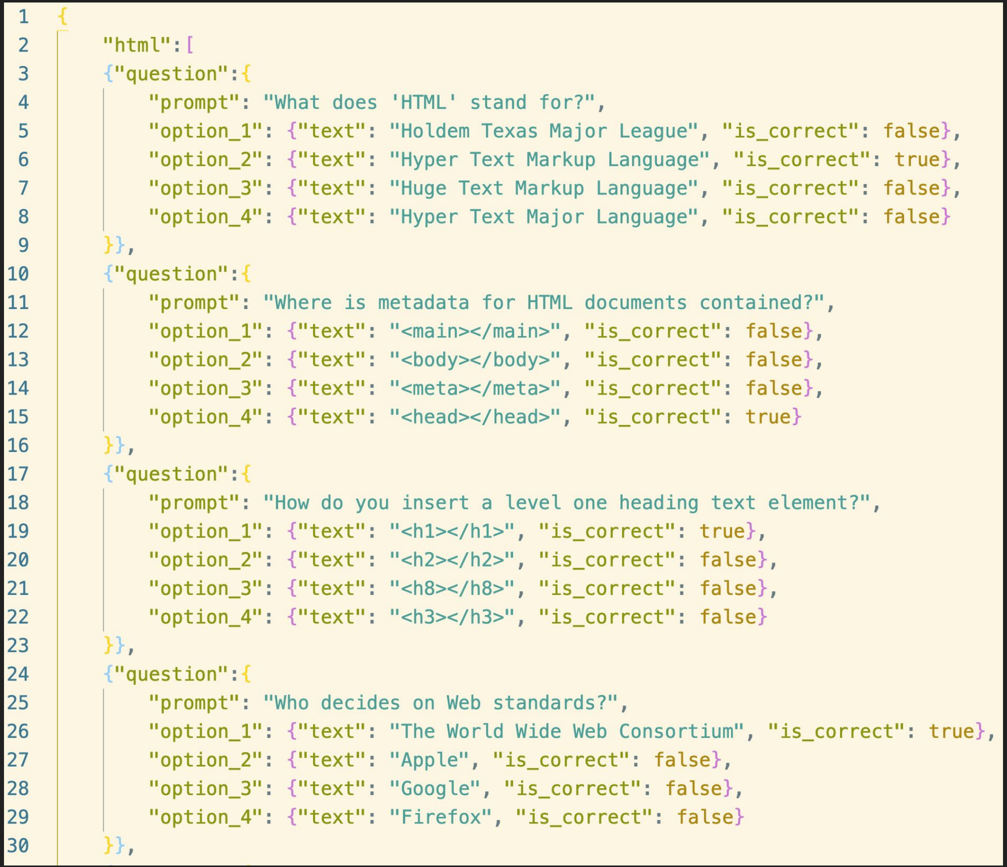


```ruby
case selection

when 1
  ARGV[0] = 'html'
  start_quiz(data_hash[ARGV[0]])

when 2
  ARGV[0] = 'css'
  start_quiz(data_hash[ARGV[0]])

when 3
  ARGV[0] = 'ruby'
  start_quiz(data_hash[ARGV[0]])

when 4
  File.open('user_performance.txt', 'r') do |file|
    puts file.read
  end

when 5
  begin
    File.open('user_performance.txt', 'w') do |file|
      file.write('USER PERFORMANCE')
      file.write("\n--------------")
    end
  rescue StandardError
    puts "It looks like the path to 'user_performance.txt' is broken!"
  end
end
```

Code

Json file stores quiz data

- The top level hashes have quiz topics as keys
- Those keys have an array of question hashes as values
- Those hashes contain more hashes with prompts, option text and whether each option is correct or not.

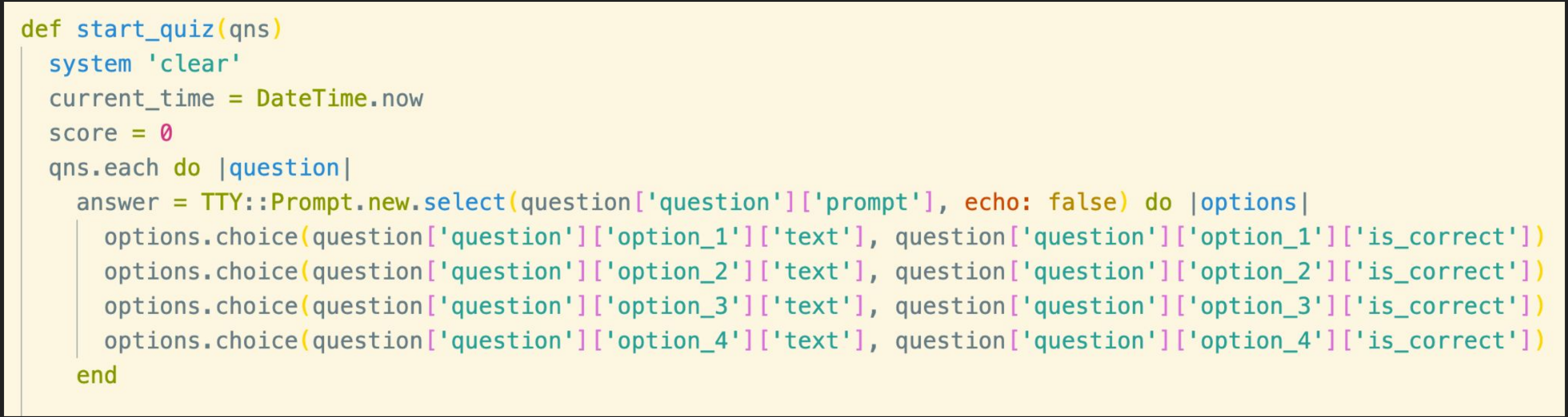


```
{
    "html":[
    {"question":{
        "prompt": "What does 'HTML' stand for?",
        "option_1": {"text": "Holdem Texas Major League", "is_correct": false},
        "option_2": {"text": "Hyper Text Markup Language", "is_correct": true},
        "option_3": {"text": "Huge Text Markup Language", "is_correct": false},
        "option_4": {"text": "Hyper Text Major Language", "is_correct": false}
    }},
    {"question":{
        "prompt": "Where is metadata for HTML documents contained?",
        "option_1": {"text": "<main></main>", "is_correct": false},
        "option_2": {"text": "<body></body>", "is_correct": false},
        "option_3": {"text": "<meta></meta>", "is_correct": false},
        "option_4": {"text": "<head></head>", "is_correct": true}
    }},
    {"question":{
        "prompt": "How do you insert a level one heading text element?",
        "option_1": {"text": "<h1></h1>", "is_correct": true},
        "option_2": {"text": "<h2></h2>", "is_correct": false},
        "option_3": {"text": "<h8></h8>", "is_correct": false},
        "option_4": {"text": "<h3></h3>", "is_correct": false}
    }},
    {"question":{
        "prompt": "Who decides on Web standards?",
        "option_1": {"text": "The World Wide Web Consortium", "is_correct": true},
        "option_2": {"text": "Apple", "is_correct": false},
        "option_3": {"text": "Google", "is_correct": false},
        "option_4": {"text": "Firefox", "is_correct": false}
    }},
```

Code:

Running the quiz

- Iterator combined with TTY prompt to access values in the JSON file
- This method takes in an array of questions as an input (so you can pass in which quiz you want to run)
- For each question, it will display the prompt, all of the options, and store whether the user's answer was correct or not

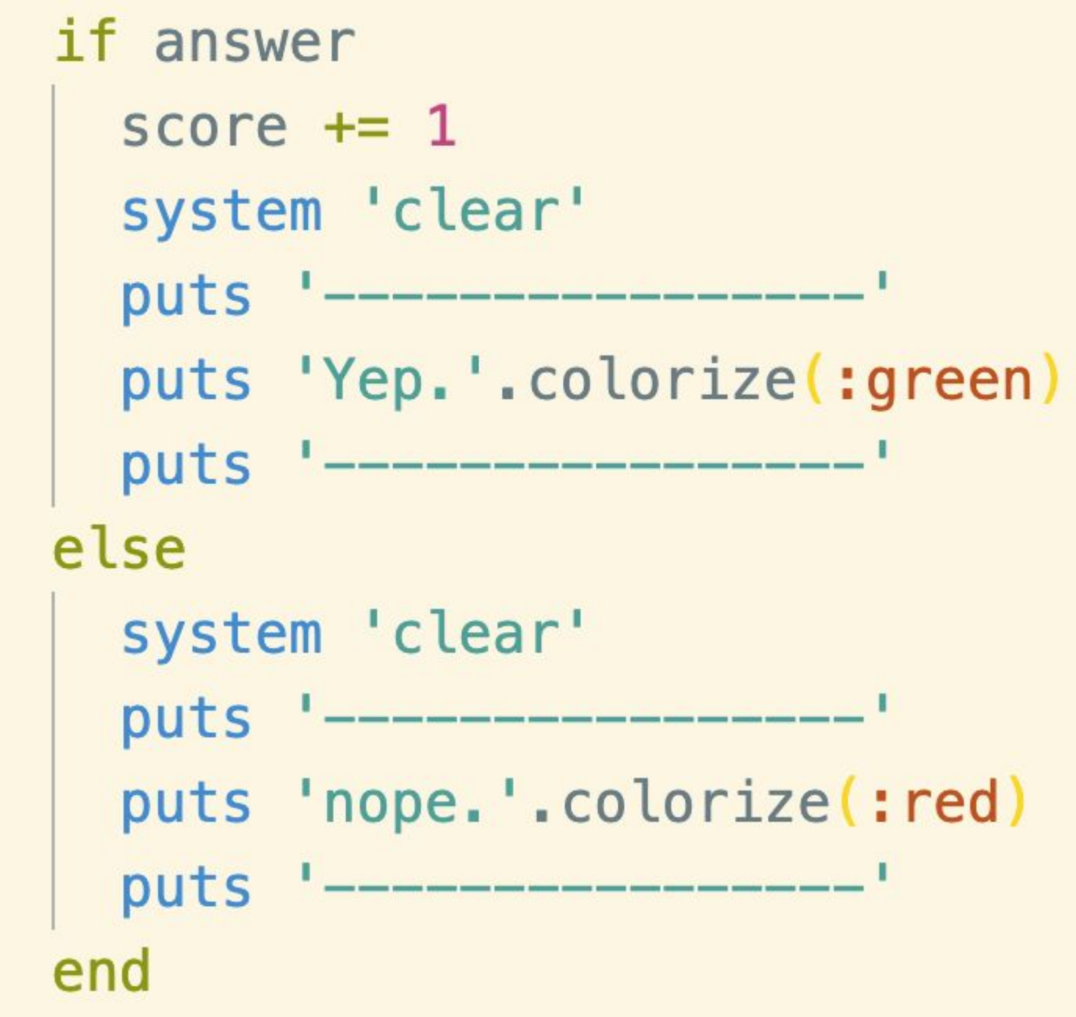


```
def start_quiz(qns)
  system 'clear'
  current_time = DateTime.now
  score = 0
  qns.each do |question|
    answer = TTY::Prompt.new.select(question['question']['prompt'], echo: false) do |options|
      options.choice(question['question']['option_1']['text'], question['question']['option_1']['is_correct'])
      options.choice(question['question']['option_2']['text'], question['question']['option_2']['is_correct'])
      options.choice(question['question']['option_3']['text'], question['question']['option_3']['is_correct'])
      options.choice(question['question']['option_4']['text'], question['question']['option_4']['is_correct'])
    end
```

Code:

Incrementing the score

- ‘if’ statement used to respond to whether or not the user’s answer is correct

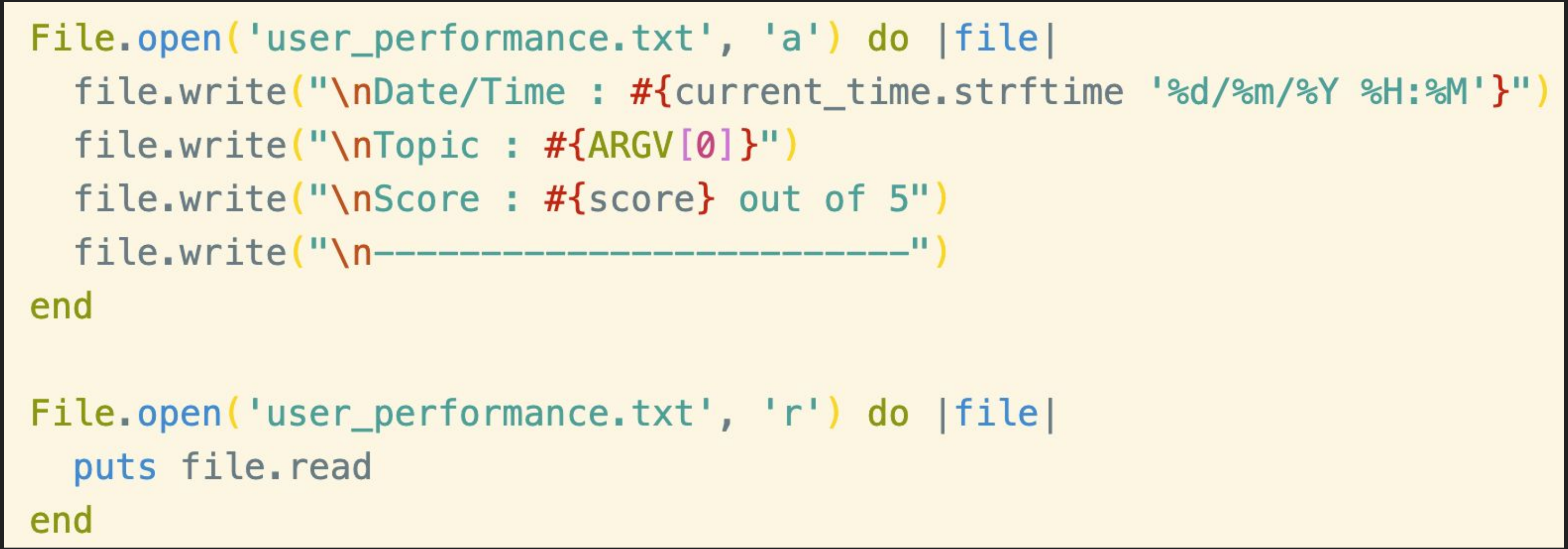


```
if answer
  score += 1
  system 'clear'
  puts '----------------'
  puts 'Yep.'.colorize(:green)
  puts '----------------'
else
  system 'clear'
  puts '----------------'
  puts 'nope.'.colorize(:red)
  puts '----------------'
end
```

Code:

Working with .txt

- Writing user performance to text file
- Displays updated text file



```
File.open('user_performance.txt', 'a') do |file|
  file.write("\nDate/Time : #{current_time.strftime '%d/%m/%Y %H:%M'}")
  file.write("\nTopic : #{ARGV[0]}")
  file.write("\nScore : #{score} out of 5")
  file.write("\n-------------------------")
end

File.open('user_performance.txt', 'r') do |file|
  puts file.read
end
```

Live Walkthrough

Development Review

- I originally stored the quiz data in objects with their own instance variables
- Accessing values from a JSON file was great practice for understanding hashes and arrays

Development Review - favourite parts

- Debugging
- Successfully using an iterator on a JSON file
- Hacking together mockups

Thanks for listening :)