

Terminal Application Assignment

Coder Academy.

Hello and welcome to my terminal application assignment for Coder Academy 2019. In this terminal app I will be going through how and why I built a multiple question quiz for my assignment.

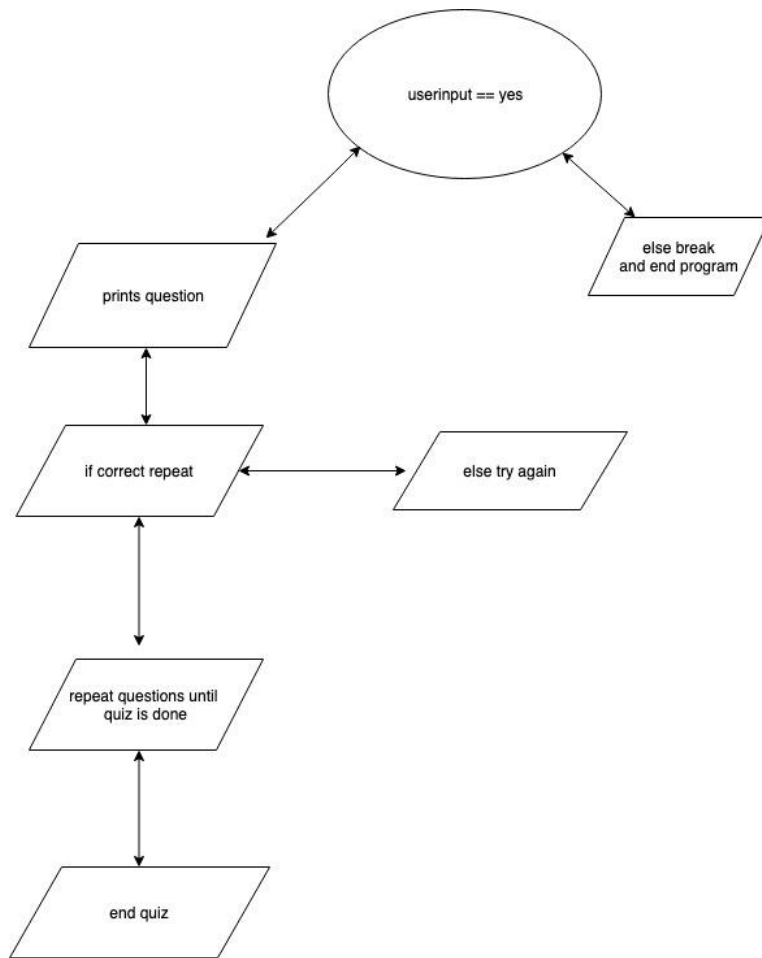
Description.

This questionnaire will provide users with a select number of questions about general history that include topics like the world wars and American presidency.

The target audience for my app would ideally be people interested in history but it can also be used to help people who want to learn about history. The app will go through 10 questions that will have something to do with history.

Motivation.

I decided to make this multiple question quiz for my terminal application assessment for Coder Academy because I love history whether it be about wars or just general history, it's all interesting to me. So, I thought why not share my interest and make a quiz about history where the user enters in what they think will be the answer.



Features and Function

terminal_app.rb is the main file and where the user interaction is set.

questions_answers.rb is where all of my questions and answers that the user reads and interacts with are set into a hash inside of an array.

In my program I had to use different variables and loops to get my program to work. I used a variable called "questions" to make an array for all of my questions and answers to go into. I then had to use an until loop and assign a variable called game_over and set it to false which makes the program loop until the user exits out of the quiz. Inside of the until loop I used if loops to test if the user was inputting the right commands.

Status Updates

1. Got my loop to print my welcome message and the user to input "Yes or No"
2. Added my first question into an array but found out that I needed to use a hash inside of an array.
3. Implemented more questions and answers into the hash.
4. Kept on working on my loop to figure out how to be able to display the question.
5. Got the Question to print but also ended up printing the answers which was not ideal.
6. I then got the help of a few classmates who told me what I should do instead and it worked. I was printing the questions and the answers separately.
7. I then continued working on my loop but this time I got a gem which allowed me to make the user select what answer they would like to pick.
8. I worked more on my loop to get the gem to work and in the end, I was left with my final product full with colour and user selection

Quiz Process

When the user starts the program, it'll be greeted with a message asking if they would like to take the quiz or not. By typing "Yes" the program will then take them to the first question. The program will run through the quiz as the user enters the answers and print out with "Correct" or "Wrong" depending on the answer the user chooses. After the user is finished the quiz they will be given a chance to either go back and take the quiz again or they can just exit the program.

Gems

Tty::Prompt – For selecting the different answers with (up/down arrows)

Colorize – For adding color to the program

Bundler

Instructions for use

1. Install Ruby (if you have HomeBrew you can use your terminal if not install homebrew first then come back)
2. Use Homebrew to install the latest version of Ruby

> brew install ruby

3. Once you've installed Ruby, you'll need to get the gem files
4. Go to [RubyGems.org](https://rubygems.org) and search for the gems and install them (You may have to use "sudo" to manually install them)

> sudo gem install tty-prompt

> sudo gem install colorize

> sudo gem install bundler

5. Once you have installed the gems, you'll want to use your terminal to add a gemfile into your working directory

> bundle init (This will create a gemfile into your working directory)

> bundle gem add "gemname" (This will add the gem into your gemfile)

6. Once you have finished adding the gems into your gemfile you can now use them in your code as long as they are in the same directory by using the keyword "require".

Timeline

In the beginning I started off with a while unit thinking that while everything was true then the loop would just continue to print the questions and asking the user to enter the answers. But like many I ran into problems with getting my code to output either the questions or letting the user enter their answer. I also ran into problems with figuring out how to add the questions into a hash and getting that to display in the terminal but with the help of my classmates and teachers I figured out how to add the questions and answers into a hash and then display them in the terminal. I then applied my gems to make the program pop and I was done after all of the syntax errors and load errors I was finished all I had to do now was the documentation. I then started the testing process by using Test Unit Development to test that what my code was doing was running smoothly. I ran into some issues mainly with just naming things and placing things in wrong places. But once I figured out where things should go from help from my classmates and teachers, I got the testing working and then it was just a process of elimination to get it all sorted.

Trello.

You can view my user story at <https://trello.com/b/EpCeDjic/terminal-app>

Glossary

- Variable - a value or option that changes over time or based on user input.
- Hash - a collection of key-value pairs
- Terminal - an interface that allows you to access the command line from the GUI.
- Gems - a module/Library that you can install and use in every project on your server.