Activity: Create a Trivia Game

In this activity, we're going to bring together each of the concepts from this module to build a trivia game.

Instructions

1. Click HERE for the link to the repo.
2. `Fork` (not *clone*) it to your **OWN** GitHub account.
3. Now to `clone` the repo to your machine, click the green 'Code' button and then copy the URL.
4. In a new terminal, or Git Bash, go to where you want to clone the repo.
5. Type `git clone` in the terminal or Git Bash, then a space, then paste the URL you copied from your repo. **Example:**

```
git clone https://github.com/HackerUSA-CE/FSI-Create-a-Trivia-Game.git
```
Click here to copy

6. Hit "Enter" or "Return" whichever is on your keyboard.
7. Do the assignment in Visual Studio Code and stage your changes using `git add -A` command.
8. Make at least one commit by using `git commit -m` "write your message here" command. **Example:**

```
git commit -m "changed font size on navbar"
```
Click here to copy

9. Finally push your changes using the `git push` command. **Example:**

```
git push origin main
```
Click here to copy

Our Goal

In our finished trivia game, a user should be able to:

- Provide their name.
- Answer a series of trivia questions.
- Be awarded 10 points for each question they answer correctly.
- See their score at the end of the game.
- (Bonus) Decide if they want to play the game again or not.

We will walk through how to build a trivia game throughout this activity.

Note: There are questions throughout this exercise. Take careful note of them, as they will be helpful for you to ask yourself when you're solving any other programmatic challenge.

The Starter Code

The only significant code in this repository is in `questions.js`.
Open `questions.js` and briefly review it.

> What type of variable is "questions"?

- ○ An object
- ○ An array

"Questions" is an array of _?

- ○ Objects
- ○ Strings
- ○ Numbers
- ○ Arrays

1. Get the user's name.

Open `index.js` in your code editor.
First, we'll want to get and **remember** the user's name.

We can ask for a name with `window.prompt` and then assign it to a variable so we can reference it later:

index.js

```
let userName = window.prompt('Please enter your name:')
```

Click here to copy

At this point, as a user, you should be prompted for your name, which you should be able to enter.

After entering your name, check that it was remembered by typing **userName** into the developer console.

2. Prompt the user with each question.

Next, the user should answer each question in our questions array.

To do something for **each** thing in an array, we'll use a `for` loop:

index.js

```
let userName = window.prompt('Please enter your name:')

for(let i = 0; i < questions.length; i++){

}
```

Click here to copy

Based on our review of the questions array earlier, what would we expect `questions[i]`, an element from the questions array, to be?

Let's use `console.log` to confirm:

index.js

```
let userName = window.prompt('Please enter your name:')

for(let i = 0; i < questions.length; i++){
    console.log(questions[i])
}
```

Click here to copy

Enter your name in the browser to start the game, then check the developer console for `questions[i]`.

What type of variable was logged?

- ○ Object
- ○ String
- ○ Number
- ○ Array

We could continue to refer to this object as `questions[i]`, but our code might get rather challenging to read.

Sometimes, it's helpful to create a variable just to give something a more readable name:

index.js

```
let userName = window.prompt('Please enter your name:')

for(let i = 0; i < questions.length; i++){
    let question = questions[i]

}
```

Click here to copy

Next, we need to prompt the user to answer each question as we iterate through them.

We can use `window.prompt` again, but we can't hard-code a message to prompt the user with, since we want the user to be prompted with different text for each question.

index.js

```
let userName = window.prompt('Please enter your name:')

for(let i = 0; i < questions.length; i++){
    let question = questions[i]
    window.prompt( ? )
}
```

Click here to copy

Let's look more closely at one of the question objects to figure out how we want to present each question to the user.

```
{
    text: `What is the first book of the Old Testament?
    A. Leviticus
    B. Numbers
    C. Genesis
    D. Exodus`,
    correctAnswer: "C"
}
```

Click here to copy

What property of each question object do we want to display to the user?

- ○ text
- ○ correctAnswer

- ○ What's a property?

How could we reference that property within our loop?

- ○ question[text]
- ○ question.text
- ○ questions[i].text
- ○ text.question

We'll also want to save the user's answer to a variable, so as a whole, our prompt would look something like:

index.js

```
let userName = window.prompt('Please enter your name:')

for(let i = 0; i < questions.length; i++){
    let question = questions[i]
    let userAnswer = window.prompt(question.text)
}
```

Click here to copy

Use `console.log` to check `usersAnswer`, then open your browser and answer each of the questions presented.

You should see each answer appear in the developer console when you enter it.

3. Check the user's answer for each question.

Next, we want to check if the user has entered the right answer.

```
{
    text: `What is the first book of the Old Testament?
    A. Leviticus
    B. Numbers
    C. Genesis
    D. Exodus`,
    correctAnswer: "C"
}
```

Click here to copy

How could we reference the correct answer for the current question within our loop?

- ○ question.text
- ○ question.correctAnswer
- ○ correctAnswer.question
- ○ We can't.

With that in mind, let's use another `console.log` to test our `if` condition:

index.js

```
let userName = window.prompt('Please enter your name:')

for(let i = 0; i < questions.length; i++){
    let question = questions[i]
    let userAnswer = window.prompt(question.text)
    if(userAnswer === question.correctAnswer){
        console.log('Correct')
    }
}
```

Click here to copy

In your browser, enter your name to start the game, then enter **C** for the first question. *Correct* should appear in the developer tools.

For the next question, enter **C** again. This is not the correct answer, so *Correct* should not appear a second time.

Note: Because we are checking if the user's input is equal to the correctAnswer, you will need to make sure you enter your answers exactly as they appear in question.js (uppercase letters).

4. Track the user's score.

Next, we need the game to **remember** the user's score, and add to it when they get a question correct.

What could we do to remember the user's score?

- ○ Use another for loop.
- ○ Use another if condition.
- ○ Define a new variable.
- ○ Add the user's score to their name.

We'll reassign the variable whenever we want to add to it, and use `window.alert` to display the score after all the questions have been asked:

index.js

```
let userName = window.prompt('Please enter your name:')
let userScore = 0

for(let i = 0; i < questions.length; i++){
    let question = questions[i]
    let userAnswer = window.prompt(question.text)
    if(userAnswer === question.correctAnswer){
        userScore = userScore + 10
    }
}

window.alert('Your score is: '+userScore)
```

Click here to copy

At this point, you should be able to play through the game and see your score after answering all 10 questions.

What would happen if we moved `window.alert` (line 12) inside of the loop?

- ○ An error would be thrown.
- ○ Nothing would change (everything would still work fine).
- ○ The score would be displayed after each question.

- ○ The score would never be displayed.

(Bonus) Decide to play the game again or not.

Now since we have a functioning trivia game, we want the user to be able to decide whether or not the user wants to play the game again at the end of each round.

Feel free to try this on your own first! We can do this a few ways so here is the logic we will follow in this activity:

- While the user wants to play the game, we want the code that we have written in previous steps to be run (to show the questions, check answer, etc.).
- If the user does not want to play the game, we want to display a message and stop the game.

We can use a function to help us achieve this functionality. Remember that a function is a block of code designed to perform a particular task. A function expression needs these properties:

- the keyword `function`
- a function name
- a pair of parentheses
- a pair of curly braces

For example:

```
function playGame() {
// our code will go here
}
```

Click here to copy

Let's define a function called `playGame` that will show the questions, check if the user answer is correct, and display the user score to the user.

Our code should look something like this:

```
let userName = window.prompt('Please enter your name:')
let userScore = 0

function playGame() {
    for(let i = 0; i < questions.length; i++){
        let question = questions[i]
        let userAnswer = window.prompt(question.text)
        if(userAnswer === question.correctAnswer){
            userScore = userScore + 10
        }
    }
    window.alert('Your score is: '+userScore)
}
```

Click here to copy

Now that we have defined a function, we need to call the function to use it. Let's call the function and test our program to check that the game still works.

```
let userName = window.prompt('Please enter your name:')
let userScore = 0

function playGame() {
    for(let i = 0; i < questions.length; i++){
        let question = questions[i]
        let userAnswer = window.prompt(question.text)
        if(userAnswer === question.correctAnswer){
            userScore = userScore + 10
        }
    }
    window.alert('Your score is: '+userScore)
}
playGame()
```

Click here to copy

Now we want to implement the logic for while the user wants to play the trivia game, call `playGame()`. After each game, check if the user wants to play the game again. If the user wants to play the game again, call `playGame()`. Otherwise, display a message and end the program.

In order to do this, we are going to use a `while loop`. Here is more information on the while loop. A `while loop` loops through a section of code when a specific condition is true.

The syntax is:

```
while (condition) {
    // add code here to run when the condition is true
}
```

Click here to copy

Let's declare a boolean variable called `playAgain` at the start of our program and set equal to `true`. Now we can edit our code so that while `playAgain` is equal to `true`, call the `playGame()` function.

At this point, our code should look something like this:

```
let userName = window.prompt('Please enter your name:')
let userScore = 0
let playAgain = true

function playGame() {
    for(let i = 0; i < questions.length; i++){
        let question = questions[i]
        let userAnswer = window.prompt(question.text)
        if(userAnswer === question.correctAnswer){
            userScore = userScore + 10
        }
    }
    window.alert('Your score is: '+userScore)
}

while (playAgain === true) {
    playGame()
}
```

Click here to copy

In this loop, we can ask for the user's choice (if they want to play the game or not) using `window.prompt` and then assign it to a variable so we can reference it later:

```
while (playAgain === true) {
    playGame()
    let userChoice = window.prompt('Would you like to play the game again? Answer yes or no.')
}
```

Click here to copy

Next we can add the conditional logic in the loop that checks the user choice. If `userChoice` equals yes, we want `playAgain` to have a value of `true`. Otherwise, we want to set `playAgain` to `false` to stop the loop from running again and use `window.alert` to display a message that the game is ending.

Our code in the loop should look something like this:

```
while (playAgain === true) {
    playGame()
    let userChoice = window.prompt('Would you like to play the game again? Answer yes or no.')
    if(userChoice === 'yes'){
            playAgain = true
        } else {
            playAgain = false
            window.alert('Thanks for playing the game!')
        }
}
```

Click here to copy

Nice work. Now we should play the trivia game a few times to check the functionality!

Acceptance Criteria

- When running your application, it should prompt the user to answer questions in succession.
- When running your application, the user's correct answers are logged in the console as correct.

- When running your application, points for each correct answer are awarded and the total score is displayed in the final prompt.

Before submitting, make sure you do a self review of your code, check for formatting, spelling, include comments in your code, and ensure you have a healthy commit history.

Make sure to submit your GitHub repository link on the submission page.