DOM Manipulation in JavaScript Guided Project

Introduction

This week we will be **a mini quiz app - Test Your Knowledge** in JavaScript to help us gain understanding of DOM Manipulation in JavaScript.

We will be working with the basic concepts of DOM Manipulation like getting element by id, updating element text, etc

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

Our task is as follows -

1. **Initializing Quiz Parameters**: At the beginning of the script, you are tasked with initializing several variables that define the quiz's current state. These variables include **currentQuestion**, **score**, **numOfQuestions**, **numOfChoices**, and **questionCountElement**. Your task is to set **currentQuestion** and **score** to their initial values, calculate **numOfQuestions** based on the length of the **questions** array, set **numOfChoices** to the number of answer choices per question, and obtain a reference to the HTML element with the id "question-count" and store it in **questionCountElement**.

2. **Displaying the Current Question**: In the **displayQuestion()** function, you are responsible for updating the quiz interface to display the current question. This function should retrieve the question text from the **questions** array and display it in the HTML element with the id "question." Additionally, it should update the question count by displaying the current question number out of the total number of questions in the format "currentQuestion/numOfQuestions" in the HTML element with the id "question-count." Lastly, it should populate the answer choice buttons with the options for the current question.

3. **Checking the User's Answer**: In the **checkAnswer(choiceIndex)** function, your task is to determine whether the user's selected answer (given by **choiceIndex**) is correct for the current question. Compare the selected choice to the correct answer index stored in the **questions** array. If the answer is correct, increment the user's score. After checking the answer, update the current question count displayed in the HTML element with the id "question-count." If there are more questions, call the **displayQuestion()** function to move to the next question; otherwise, call the **endQuiz()** function to finish the quiz.

4. **Updating the User's Score**: In the **updateScore()** function, your responsibility is to update the displayed score on the quiz interface. If the current question is the last question in the quiz, display the final score in the format "Final Score: score / numOfQuestions" in the HTML element with the id "score." Otherwise, if the quiz is still ongoing, display the current score in the format "Current Score: score."

5. **Ending the Quiz**: The **endQuiz()** function handles the conclusion of the quiz. When this function is called, your task is to clear the question area by updating the HTML element with the id "question-container" to display a message indicating that the quiz has been completed. Additionally, remove the answer choice buttons by clearing the HTML element with the id "choices-container." Ensure that the current question count displayed in the HTML element with the id "question-count" is also cleared.

6. **Restarting the Quiz**: The **restartQuiz()** function allows users to restart the quiz. When this function is called, your task is to reset the **currentQuestion** and **score** variables to their initial values (0). Additionally, update the displayed score on the quiz interface by calling the **updateScore()** function. Reset the HTML elements for the question and answer choices to their initial state, and call the **displayQuestion()** function to start the quiz from the beginning.