```
1
2   document.addEventListener("DOMContentLoaded", () => {
3       // DOM Elements
4       const levelOfDifficulty = document.querySelectorAll(".level-selection .choice");
5       const takeTriviaBtn = document.querySelector(".take-trivia");
6       const multipleChoice = document.querySelectorAll(".multiple-choice .choice");
7       const verifyBtn = document.querySelector(".verify-answer");
8       const nextTrivia = document.querySelector(".next-trivia");
9       const triviaPage = document.querySelector(".trivia-section");
10      const howToPlayPage = document.querySelector(".how-to-section");
11      const backToLevel = document.querySelector(".back-to-level");
12      const scorePage = document.querySelector(".score-section");
13      const nbaTrivia = document.querySelector(".nba-trivia");
14      const questionPage = document.querySelector(".question-section");
15
16
17      // Initial setup
18      takeTriviaBtn.style.visibility = "hidden";
19      takeTriviaBtn.style.opacity = "0";
20      nextTrivia.style.display = "none";
21
22      // Variables to track trivia progress
23      let questionNumber = 0;
24      let questions;
25      let score = 0;
26      let currentQuestionIndex = 0;
27      let currentQuestion;
28      let sendAnswer = false;
29
30      // Function to clear answer styles
31      const clearAnswers = () => {
32          multipleChoice.forEach(choice => {
33              choice.classList.remove("choice-selected", "correct-answer", "incorrect-answer
34              choice.style.cursor = "pointer";
35          });
36      };
37
38      // Function to clear buttons
39      const clearBtns = () => {
40          verifyBtn.style.display = "block";
41          verifyBtn.style.visibility = "hidden";
42          nextTrivia.style.visibility = "hidden";
43          nextTrivia.style.display = "none";
44          nextTrivia.style.opacity = "0";
45          takeTriviaBtn.style.visibility = "hidden";
46          takeTriviaBtn.style.opacity = "0";
47      };
48
49      // Function to reset trivia
50      const resetTrivia = () => {
51          sendAnswer = false;
52          questionNumber = 0;
53          triviaPage.classList.remove("hiddenPage");
54          triviaPage.style.display = "block";
55          questions = {};
56          score = 0;
57          currentQuestionIndex = 0;
58          currentQuestion = "";
59          nbaTrivia.textContent = "";
60          sendAnswer = false;
61          scorePage.style.display = "none";
62          clearAnswers();
63          clearBtns();
64          takeTriviaBtn.style.visibility = "hidden";
65          takeTriviaBtn.style.opacity = "0";
66      };
67
68      // Event listener for back to level button
69      backToLevel.addEventListener("click", resetTrivia);
70
71      // Event listeners for difficulty level buttons
72      levelOfDifficulty.forEach(choice => {
73          choice.addEventListener("click", () => {
74              levelOfDifficulty.forEach(cho => cho.classList.remove("choice-selected"));
75              choice.classList.toggle("choice-selected");
76              takeTriviaBtn.style.visibility = "visible";
77              takeTriviaBtn.style.opacity = "1";
78          });
79      })
80
81     // Event listener for take trivia button
82      takeTriviaBtn.addEventListener("click", () => {
83      triviaPage.classList.add("hiddenPage");
84      howToPlayPage.classList.add("hiddenPage");
85      questionPage.classList.remove("hiddenPage");
86      setTimeout(() => {
87          triviaPage.style.display = "none";
88          howToPlayPage.style.display = "none";
89          questionPage.style.display = "block";
90      }, 300);
91
92      const selectedLevel = document.querySelector(".choice-selected");
93      nbaTrivia.style.visibility = "visible";
94      let level = selectedLevel.dataset.info;
95
96      // Set NBA Trivia text and icon based on difficulty level
97      nbaTrivia.innerHTML = `<i class="icon"></i><span>${level.charAt(0).toUpperCase() + lev
98      nbaTrivia.classList.add(`level-${level === "simple" ? 1 : level === "intermediate" ? 2
99
100     // Fetch trivia questions based on difficulty level
101     fetch(`./resources/trivia/${level}.json`)
102         .then(response => response.json())
103         .then(data => {
104             questions = data;
105             showNextQuestion();
106         })
107         .catch(error => console.error("Error while fetching questions:", error));
108 });
109
```

(use 'esversion: 6').

72 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

73 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

79 Missing semicolon.

82 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

86 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

92 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

94 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

97 'template literal syntax' is only available in ES6 (use 'esversion: 6').

98 'template literal syntax' is only available in ES6 (use 'esversion: 6').

101 'template literal syntax' is only available in ES6 (use 'esversion: 6').

102 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

103 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

107 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

111 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

112 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

114 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

123 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

125 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

131 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

140 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

149 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

160 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

165 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

165 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

172 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

179 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

181 'template literal syntax' is only available in ES6 (use 'esversion: 6').

184 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

190 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

193 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

203 Missing semicolon.

203 Expected ')' and instead saw ';'.

203 Missing semicolon.

203 Unrecoverable syntax error. (98% scanned).

JSHint

version 2.13.6
(https://github.com/jshint/j

- About (/about)
- Documentation (/docs)
- Install (/install)
- Contribute (/contribute)
- Blog (/blog)

```
110     // Event listeners for multiple-choice buttons
111     multipleChoice.forEach(choice => {
112         choice.addEventListener("click", () => {
113             if (!sendAnswer) {
114                 multipleChoice.forEach(cho => cho.classList.remove("choice-selected"));
115                 choice.classList.toggle("choice-selected");
116                 verifyBtn.style.visibility = "visible";
117                 verifyBtn.style.opacity = "1";
118             }
119         });
120     });
121
122     // Event listener for verifying the answer
123     verifyBtn.addEventListener("click", () => {
124         sendAnswer = true;
125         multipleChoice.forEach(cho => cho.style.cursor = "not-allowed");
126         verifyBtn.style.display = "none";
127         nextTrivia.style.visibility = "visible";
128         nextTrivia.style.display = "block";
129         nextTrivia.style.opacity = "1";
130
131         const answer = document.querySelector(".multiple-choice .choice-selected");
132
133         // Check if the selected answer is correct
134         if (answer.textContent === currentQuestion.answer) {
135             answer.classList.add("correct-answer");
136             score++;
137         } else {
138             answer.classList.add("incorrect-answer");
139             // Highlight the correct answer
140             multipleChoice.forEach(choice => {
141                 if (choice.textContent === currentQuestion.answer) {
142                     choice.classList.add("correct-answer");
143                 }
144             });
145         }
146     });
147
148     // Event listener for moving to the next trivia
149     nextTrivia.addEventListener("click", () => {
150         if (currentQuestionIndex + 1 === questions.questions.length) {
151             questionPage.style.display = "none";
152             scorePage.style.display = "block";
153             scorePage.style.visibility = "visible";
154             document.querySelector(".score").textContent = score;
155             return;
156         }
157         currentQuestionIndex++;
158         clearBtns();
159         sendAnswer = false;
160         multipleChoice.forEach(cho => cho.style.cursor = "pointer");
161         showNextQuestion();
162     });
163
164     // Function to display the next question
165     const showNextQuestion = () => {
166         clearAnswers();
167
168         takeTriviaBtn.style.visibility = "hidden";
169         takeTriviaBtn.style.opacity = "0";
170
171         if (questions && questions.questions && questions.questions.length > currentQuesti
172             const questionData = questions.questions[currentQuestionIndex];
173
174             if (questionData) {
175                 currentQuestion = questionData;
176
177                 // Update the question number
178                 questionNumber++;
179                 const questionNumberElement = document.querySelector(".question-section h3
180                 if (questionNumberElement) {
181                     questionNumberElement.textContent = `${questionNumber}/${questions.que
182                 }
183
184                 const questionText = document.querySelector(".question-section .question-t
185
186                 if (questionText) {
187                     questionText.textContent = currentQuestion.question;
188                 }
189
190                 const choices = document.querySelectorAll(".multiple-choice .choice");
191
192                 // Reset the display property to 'block' for each choice
193                 choices.forEach((choice, index) => {
194                     choice.style.display = "block";
195                     choice.textContent = currentQuestion.options[index];
196                 });
197             } else {
198                 console.error("Error loading current question.");
199             }
200         } else {
201             console.log("Quiz concluded!");
202         }
203         }};
204
205 });
206
```

(use 'esversion: 6').

72 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

73 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

74 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

79 Missing semicolon.

82 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

86 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

92 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

94 'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

97 'template literal syntax' is only available in ES6 (use 'esversion: 6').

98 'template literal syntax' is only available in ES6 (use 'esversion: 6').

101 'template literal syntax' is only available in ES6 (use 'esversion: 6').

102 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

103 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

107 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

111 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

112 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

114 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

123 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

125 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

131 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

140 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

149 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

160 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

165 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

165 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

172 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

179 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

181 'template literal syntax' is only available in ES6 (use 'esversion: 6').

184 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

190 'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

193 'arrow function syntax (=>)' is only available in ES6 (use 'esversion: 6').

203 Missing semicolon.

203 Expected ')' and instead saw ';'.

203 Missing semicolon.

203 Unrecoverable syntax error. (98% scanned).

version 2.13.6
(https://github.com/jshint/j

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)