Interactive Quiz Guide

Creating an **Interactive Quiz Application** is a great way to practice JavaScript, especially **DOM manipulation, event handling, and basic logic**. I'll walk you through the steps and provide the full code at the end.

Step 1: Plan the Quiz

Features:

1. Display a question with multiple choices.
2. Let users select an answer.
3. Highlight correct and incorrect answers.
4. Move to the next question after selecting an answer.
5. Show the final score when the quiz ends.

Step 2: Set Up the Project

Create three files:

- `` `index.html` `` → The structure of the quiz.
- `` `style.css` `` → Styling for a clean UI.
- `` `script.js` `` → JavaScript logic.

Step 3: Create the HTML Structure (`` `index.html` ``)

We need:

- A **container** to hold the quiz.
- A **question** area.
- **Buttons** for multiple-choice options.
- A **next** button to go to the next question.

```
html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Interactive Quiz</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="quiz-container">
        <h1>Quiz Application</h1>
        <div id="question-container">
            <p id="question">Question text here...</p>
            <div id="answer-buttons" class="btn-container"></div>
        </div>
        <button id="next-btn">Next</button>
        <p id="score-display"></p>
    </div>

    <script src="script.js"></script>
</body>
</html>
```

Step 4: Add CSS for Styling (`` `style.css` ``)

This will make the quiz visually appealing.

```
css

body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f5f5f5;
}

.quiz-container {
    background: white;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: center;
    width: 400px;
}

.btn-container {
    display: flex;
    flex-direction: column;
}

button {
    margin: 5px;
    padding: 10px;
    width: 100%;
    font-size: 16px;
    cursor: pointer;
    border: none;
    border-radius: 5px;
    transition: 0.3s;
}

button.correct {
    background-color: #4CAF50;
    color: white;
}

button.wrong {
    background-color: #f44336;
    color: white;
}

#next-btn {
    display: none;
    background-color: #007BFF;
    color: white;
}
```

Step 5: Write JavaScript Logic (`` `script.js` ``)

This file will:

1. Store quiz questions.
2. Display a question and its choices.
3. Handle user selection and check if it's correct.
4. Move to the next question.
5. Show the final score at the end.

```js
// Select necessary elements
const questionElement = document.getElementById("question");
const answerButtons = document.getElementById("answer-buttons");
const nextButton = document.getElementById("next-btn");
const scoreDisplay = document.getElementById("score-display");

let currentQuestionIndex = 0;
let score = 0;

// Quiz questions array
const questions = [
    {
        question: "What is the capital of France?",
        answers: [
            { text: "Berlin", correct: false },
            { text: "Madrid", correct: false },
            { text: "Paris", correct: true },
            { text: "Rome", correct: false }
        ]
    },
    {
        question: "Which language runs in a web browser?",
        answers: [
            { text: "Java", correct: false },
            { text: "C", correct: false },
            { text: "Python", correct: false },
            { text: "JavaScript", correct: true }
        ]
    },
    {
        question: "What does CSS stand for?",
        answers: [
            { text: "Computer Style Sheets", correct: false },
            { text: "Cascading Style Sheets", correct: true },
            { text: "Colorful Style Sheets", correct: false },
            { text: "Creative Style Sheets", correct: false }
        ]
    }
];

// Function to start quiz
function startQuiz() {
    currentQuestionIndex = 0;
    score = 0;
    nextButton.innerText = "Next";
    nextButton.style.display = "none";
    scoreDisplay.innerText = "";
    showQuestion();
}

// Function to show a question
function showQuestion() {
    resetState();
    let currentQuestion = questions[currentQuestionIndex];
    questionElement.innerText = currentQuestion.question;

    currentQuestion.answers.forEach(answer => {
        const button = document.createElement("button");
        button.innerText = answer.text;
        button.classList.add("btn");
        button.addEventListener("click", () => selectAnswer(button, answer.correct));
        answerButtons.appendChild(button);
    });
}

// Function to reset question state
function resetState() {
    nextButton.style.display = "none";
    answerButtons.innerHTML = "";
}

// Function to handle answer selection
function selectAnswer(button, isCorrect) {
    if (isCorrect) {
        button.classList.add("correct");
        score++;
    } else {
        button.classList.add("wrong");
    }

    // Disable all buttons after selecting an answer
    Array.from(answerButtons.children).forEach(btn => btn.disabled = true);

    nextButton.style.display = "block"; // Show next button
}

// Function to go to the next question or finish quiz
function handleNextButton() {
    currentQuestionIndex++;

    if (currentQuestionIndex < questions.length) {
        showQuestion();
    } else {
        showScore();
    }
}

// Function to display the final score
function showScore() {
    resetState();
    questionElement.innerText = `Quiz Completed! Your score: ${score} / ${questions.length}`;
    nextButton.innerText = "Restart Quiz";
    nextButton.style.display = "block";
}

// Event listener for next button
nextButton.addEventListener("click", () => {
    if (currentQuestionIndex < questions.length) {
        handleNextButton();
    } else {
        startQuiz();
    }
});

// Start the quiz initially
startQuiz();
```

Step 6: Explanation

1. **HTML**
   - Contains a `div` to hold the quiz.
   - `p` tag for displaying questions.
   - Buttons for answer choices are dynamically created via JavaScript.
   - A "Next" button to proceed to the next question.
2. **CSS**
   - Adds styling for the quiz container and buttons.
   - Highlights correct answers in green and incorrect answers in red.

3. **JavaScript**
    - Stores questions and answers in an array.
    - Displays one question at a time.
    - Highlights selected answers (green for correct, red for incorrect).
    - Moves to the next question or shows the final score.

Step 7: How to Run the Project

1. Save **`index.html`**, **`style.css`**, and **`script.js`** in the same folder.
2. Open **`index.html`** in a browser.

✅ **Done! You have an interactive quiz application.** 🚀

What You Need to Do Every Time for Projects

1. **Break down the requirements.**
2. **Plan the HTML structure.**
3. **Style with CSS.**
4. **Write JavaScript for interactivity.**
5. **Test and Debug.**
6. **Enhance (like adding a timer or animations).**

Let me know if you want any modifications! 😊

ChatGPT can make mistakes. Check important info.