

Primeiro, a estrutura geral do código JavaScript organiza os dados e as interações do quiz de forma bem simples e direta. Eu uso uma variável chamada `perguntaAtual` para controlar qual pergunta o usuário está respondendo, e um array `pontuacao` para armazenar a pontuação de cada personagem. Isso ajuda a manter o estado do quiz e a acompanhar as escolhas do usuário enquanto ele avança.

Uma das primeiras funções que eu criei foi a `iniciarQuiz()`. Ela é acionada assim que o usuário clica no botão 'Iniciar'. Ela basicamente esconde a tela de boas-vindas e exibe a primeira pergunta. Para isso, manipulo o DOM utilizando `classList.add` e `classList.remove`, o que altera as classes dos elementos HTML e faz com que eles apareçam ou desapareçam conforme a interação.

Depois, a função `mostrarPergunta()` é responsável por exibir a pergunta e suas opções de resposta. Ela pega a pergunta atual do array de perguntas e coloca no lugar certo na página. Além disso, ela usa `querySelectorAll` para pegar todos os botões de resposta e, para cada um deles, eu defino o texto que será exibido e associo uma função de clique. Essa função de clique chama a função `responder()`, que é onde a lógica de pontuação acontece. Quando o usuário escolhe uma opção, a pontuação do personagem é atualizada, e o código decide qual pergunta mostrar em seguida.

Dentro da função `responder()`, eu faço uma verificação para ver se ainda há mais perguntas. Se houver, ela chama `mostrarPergunta()` para exibir a próxima pergunta. Se não, ela chama a função `mostrarResultado()`, que vai exibir o resultado final do quiz.

A função `mostrarResultado()` é bem simples: ela pega o personagem que o usuário mais se identificou e exibe sua imagem e descrição. Para isso, ela chama a função `obterPersonagem()`, que é responsável por determinar qual personagem tem a maior pontuação. Eu uso a função `Math.max()` para encontrar o maior valor de pontuação e, com isso, definir qual personagem o usuário seria. Depois, ele retorna a imagem e a descrição do personagem correspondente, e esses dados são exibidos na tela.

Além disso, o código é bem orientado a objetos em algumas partes. Eu uso objetos para organizar as perguntas, as opções e as pontuações, o que facilita o gerenciamento e a leitura do código.

Em resumo, o código utiliza interações com o DOM para exibir perguntas, opções e resultados. Ele faz uso de funções para organizar o comportamento do quiz, estruturas condicionais para controlar o fluxo do quiz e laços de repetição para iterar pelas opções de resposta. Tudo isso está organizado de uma forma que torna o quiz interativo e fácil de entender.