

Aqui vai uma explicação detalhada do código HTML e JavaScript que você pode usar para o seu canal no YouTube:

### HTML:

```
` `` html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="style.css">

  <title>Saiba mais que nos | Stv</title>

</head>

<body>

  <div class="quiz-container" id="quiz">

    <div class="quiz-header">

      <h2 id="question">Question text</h2>

      <ul>

        <li>

          <input type="radio" name="answer" id="a" class="answer">

          <label for="a" id="a_text">Question</label>

        </li>

        <li>

          <input type="radio" name="answer" id="b" class="answer">

          <label for="b" id="b_text">Question</label>

        </li>

        <li>
```

```

        <input type="radio" name="answer" id="c" class="answer">
        <label for="c" id="c_text">Question</label>
    </li>
    <li>
        <input type="radio" name="answer" id="d" class="answer">
        <label for="d" id="d_text">Question</label>
    </li>
</ul>
</div>

<button id="submit">Submit</button>
</div>

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

```

- **DOCTYPE html**: Declara que o documento está usando HTML5.
- **html lang="en"**: Define que o idioma principal do documento é o inglês.
- **meta charset="UTF-8"**: Configura o conjunto de caracteres do documento para UTF-8, que suporta a maioria dos caracteres e símbolos.
- **meta name="viewport" content="width=device-width, initial-scale=1.0"**: Faz com que a página seja responsiva, ajustando-se corretamente em dispositivos móveis.
- **link rel="stylesheet" href="style.css"**: Importa o arquivo `style.css` para adicionar estilos à página.
- **title**: Define o título da aba ou janela no navegador, que aqui é "Saiba mais que nos | Stv".
- **body**: Contém o conteúdo visível da página, como a estrutura do quiz.

- **`div class="quiz-container"`**: Cria um container para o quiz, com a classe `quiz-container` e o ID `quiz`.
- **`h2 id="question"`**: Onde o texto da pergunta do quiz será exibido.
- **`input type="radio" name="answer"`**: Botões de rádio para selecionar as respostas.
- **`label for="a" id="a_text"`**: As opções de resposta que serão exibidas como texto ao lado dos botões de rádio.
- **`button id="submit"`**: Um botão para submeter a resposta escolhida.
- **`script src="script.js"`**: Importa o arquivo `script.js`, que contém a lógica do quiz.

### JavaScript:

```
` `` javascript
```

```
const quizData= [  
  {  
    question: "What is the capital of INDIA?",  
    a: "Delhi",  
    b: "Mumbai",  
    c: "Kolkata",  
    d: "Chennai",  
    correct: "a"  
  },  
  // Outras perguntas seguem o mesmo formato  
];
```

```
const quiz = document.getElementById('quiz');  
const answersEls = document.querySelectorAll('.answer');  
const questionEl = document.getElementById('question');
```

```
const a_text = document.getElementById('a_text');
const b_text = document.getElementById('b_text');
const c_text = document.getElementById('c_text');
const d_text = document.getElementById('d_text');
const submitBtn = document.getElementById('submit');
```

```
let currentQuiz = 0;
```

```
let score = 0;
```

```
loadQuiz();
```

```
function loadQuiz(){
  deslectAnswers();
```

```
  const currentData = quizData[currentQuiz];
  questionEl.innerText = currentData.question;
  a_text.innerText = currentData.a;
  b_text.innerText = currentData.b;
  c_text.innerText = currentData.c;
  d_text.innerText = currentData.d;
}
```

```
function deslectAnswers(){
  answersEls.forEach(answerEl => answerEl.checked = false);
}
```

```
submitBtn.addEventListener('click', () => {
  const answer = getSelected();
```

```

if(answer){
  if(answer === quizData[currentQuiz].correct){
    score++;
  }
  currentQuiz++;
  if(currentQuiz < quizData.length){
    loadQuiz();
  }else{
    quiz.innerHTML = `
      <h2>You answered correctly at ${score}/${quizData.length} questions.</h2>
      <button onclick="location.reload()">Reload</button>
    `;
  }
}
});

```

```

function getSelected(){
  let answer;
  answersEls.forEach(answerEl => {
    if(answerEl.checked){
      answer = answerEl.id;
    }
  });
  return answer;
}
` ``

```

#### Explicação:

- **quizData**: Um array de objetos, onde cada objeto contém uma pergunta (`question``), quatro respostas (`a``, `b``, `c``, `d``), e a resposta correta (`correct``).
- **getElementById()** e **querySelectorAll()**: Métodos que selecionam elementos HTML pelo ID ou classe para manipulação no JavaScript.
- **currentQuiz**: Controla qual pergunta do quiz está sendo exibida no momento.
- **score**: Armazena a pontuação do usuário.
- **loadQuiz()**: Função que carrega as perguntas e respostas na tela.
- **deslectAnswers()**: Reseta os botões de rádio, para garantir que nenhuma resposta anterior permaneça selecionada.
- **addEventListener('click')**: A função que é executada quando o botão de "Submit" é clicado, verificando se a resposta está correta e carregando a próxima pergunta.
- **getSelected()**: Função que retorna a resposta selecionada pelo usuário.
- **quiz.innerHTML**: Substitui o conteúdo do quiz quando todas as perguntas foram respondidas, mostrando a pontuação final e oferecendo um botão para recarregar a página.

Esse código básico cria um sistema de quiz interativo e dinâmico.

## Funcoes empecificas

Aqui está uma explicação passo a passo do que acontece nas funções `loadQuiz()` e `deslectAnswers()`:

### **1. Chamada da Função:**

```
```javascript
```

```
loadQuiz();
```

```
```
```

A função `loadQuiz()` é chamada para carregar os dados da pergunta atual no quiz e atualizar a interface com as opções de respostas.

### \*\*2. Definição da Função `loadQuiz()`:\*\*

```
```javascript
```

```
function loadQuiz(){
```

```
```
```

Aqui, a função `loadQuiz()` é definida. Ela será usada para atualizar o conteúdo da página com os dados da próxima pergunta e suas respectivas opções de resposta.

### \*\*3. Chamada da Função `deslectAnswers()`:\*\*

```
```javascript
```

```
deslectAnswers();
```

```
```
```

A primeira coisa que a função `loadQuiz()` faz é chamar `deslectAnswers()`. Essa função serve para garantir que, ao carregar uma nova pergunta, nenhuma resposta anterior continue selecionada.

#### Explicação de `deslectAnswers()`:

```
```javascript
```

```
function deslectAnswers(){
```

```
    answersEls.forEach(answerEl => answerEl.checked = false);
```

```
}
```

```
```
```

- **answersEls**: Esta variável contém uma lista de todos os elementos de resposta (botões de rádio) com a classe `.answer`. O método `querySelectorAll('.answer')` seleciona todos esses elementos.
- **forEach**: Este método itera por cada elemento dentro de `answersEls`.
- **answerEl.checked = false**: Dentro do loop `forEach`, cada botão de rádio é "desmarcado" ao definir a propriedade `checked` como `false`. Isso desseleciona todas as respostas para garantir que o usuário comece com todos os botões de rádio desmarcados a cada nova pergunta.

#### ### **4. Atribuição da Pergunta Atual:**

```
````javascript
```

```
const currentData = quizData[currentQuiz];
```

```
````
```

- **quizData[currentQuiz]**: A variável `currentQuiz` guarda o índice da pergunta atual dentro do array `quizData`. `quizData` é uma lista de objetos, onde cada objeto contém uma pergunta e suas opções.
- **currentData**: Esta variável armazena o objeto atual do quiz (pergunta, opções de resposta e a resposta correta) baseado no índice atual (`currentQuiz`).

#### ### **5. Atualização da Pergunta na Interface:**

```
````javascript
```

```
questionEl.innerText = currentData.question;
```

```
````
```

- **questionEl**: Este elemento representa a área do HTML onde a pergunta do quiz será exibida (o elemento `<h2>` com o ID `question`).
- **currentData.question**: Esta parte acessa o texto da pergunta atual no objeto `currentData`.
- **innerText**: O conteúdo textual do elemento `questionEl` é atualizado com o texto da pergunta.



### ### **\*\*6. Atualização das Opções de Resposta:\*\***

As próximas linhas atualizam os textos das quatro opções de resposta (a, b, c, d) na interface:

```
```javascript
```

```
a_text.innerText = currentData.a;
```

```
b_text.innerText = currentData.b;
```

```
c_text.innerText = currentData.c;
```

```
d_text.innerText = currentData.d;
```

```
```
```

- **\*\*a\_text, b\_text, c\_text, d\_text\*\***: Esses elementos HTML representam os rótulos (as opções de resposta) associados aos botões de rádio. Cada um tem um ID específico ('a\_text', 'b\_text', etc.).

- **\*\*currentData.a, currentData.b, currentData.c, currentData.d\*\***: Essas partes acessam os textos das opções de resposta no objeto 'currentData'.

- **\*\*innerText\*\***: O conteúdo de cada rótulo é atualizado com o texto da opção correspondente.

### ### **\*\*Resumo Geral:\*\***

A função 'loadQuiz()' realiza as seguintes tarefas:

1. **\*\*Desmarca\*\*** todas as respostas previamente selecionadas.
2. **\*\*Carrega\*\*** os dados da pergunta atual com base no índice 'currentQuiz'.
3. **\*\*Atualiza\*\*** o texto da pergunta e as opções de resposta na interface.

A função 'deslectAnswers()' garante que todas as respostas sejam desmarcadas antes de exibir a nova pergunta, para que o usuário comece sempre com opções não selecionadas.

## Funcoes especificas

Aqui está uma explicação detalhada do que acontece dentro desse trecho de código:

### **\*\*1. Adicionando um Event Listener ao Botão de Submissão:\*\***

```
```javascript
```

```
submitBtn.addEventListener('click', () => {
```

```
...
```

- **\*\*submitBtn\*\***: Este é o botão no HTML com o ID `submit`. O código está adicionando um **\*\*event listener\*\*** para o evento `click`, ou seja, ele escuta quando o usuário clica no botão.

- Quando o botão é clicado, a função dentro de `addEventListener` é executada.

### **\*\*2. Obtendo a Resposta Seleccionada:\*\***

```
```javascript
```

```
const answer = getSelected();
```

```
...
```

- **\*\*getSelected()\*\***: Esta função retorna a resposta que o usuário escolheu, verificando qual botão de rádio foi marcado. Se nenhuma resposta for selecionada, `getSelected()` retorna `undefined`.

- **\*\*answer\*\***: Aqui, armazenamos o valor retornado de `getSelected()` na variável `answer`.

### **\*\*3. Verificando se o Usuário Seleccionou uma Resposta:\*\***

```
```javascript
```

```
if(answer){
```

```
...
```

- Este `if` verifica se o usuário realmente selecionou uma resposta. Se `answer` for `undefined` (ou seja, o usuário não escolheu nenhuma opção), o código dentro do `if` não será executado.
- Se o usuário tiver selecionado uma resposta válida, o código continua.

### \*\*4. Verificando se a Resposta Está Correta:\*\*

```
```javascript
```

```
if(answer === quizData[currentQuiz].correct){
```

```
    score++;
```

```
}
```

```
...
```

- **`answer === quizData[currentQuiz].correct`**: Esta linha compara a resposta escolhida (`answer`) com a resposta correta armazenada no objeto `quizData[currentQuiz].correct`. Se forem iguais, o usuário escolheu a resposta correta.
- **`score++`**: Se a resposta estiver correta, o código incrementa a pontuação do usuário (`score`) em 1.

### \*\*5. Atualizando para a Próxima Pergunta:\*\*

```
```javascript
```

```
currentQuiz++;
```

```
...
```

- **`currentQuiz++`**: Incrementa o valor de `currentQuiz` para carregar a próxima pergunta do quiz. Isso move o índice para a próxima pergunta no array `quizData`.

### \*\*6. Verificando se Ainda Existem Mais Perguntas:\*\*

```

```javascript
if(currentQuiz < quizData.length){
    loadQuiz();
}else{
    quiz.innerHTML = `
        <h2>You answered correctly at ${score}/${quizData.length}
questions.</h2>

        <button onclick="location.reload()">Reload</button>
    `;
}
```

```

- **if(currentQuiz < quizData.length)**: Esta condição verifica se ainda há mais perguntas no quiz. O número total de perguntas é armazenado em `quizData.length`.

- **Se houver mais perguntas**: A função `loadQuiz()` é chamada novamente, o que irá carregar a próxima pergunta e suas respectivas respostas.

- **Se não houver mais perguntas**: Isso significa que o usuário completou o quiz. Em vez de carregar outra pergunta, o código:

- Atualiza o conteúdo do `quiz` com a pontuação final.
- Exibe uma mensagem ao usuário mostrando quantas perguntas ele respondeu corretamente.
- Adiciona um botão de "Reload" que, ao ser clicado, recarrega a página inteira, permitindo que o usuário reinicie o quiz.

### ### **7. Botão de Reiniciar o Quiz:**

```

```javascript
<button onclick="location.reload()">Reload</button>
```

```

- Este botão usa o método ``location.reload()``, que recarrega a página e reinicia o quiz desde o começo.

### \*\*Resumo Geral do Processo:\*\*

1. O botão "Submit" é clicado.
2. A função ``getSelected()`` é chamada para verificar qual resposta foi escolhida.
3. Se uma resposta foi selecionada, o código verifica se está correta.
4. A pontuação é atualizada se a resposta estiver correta.
5. O quiz avança para a próxima pergunta se ainda houver perguntas.
6. Se todas as perguntas forem respondidas, o quiz exibe a pontuação final e oferece a opção de reiniciar.

Este código faz com que o quiz seja interativo, permitindo ao usuário verificar sua resposta, avançar para a próxima pergunta, e ao final, visualizar sua pontuação total.

## Funções

A função ``getSelected()`` tem o propósito de identificar e retornar qual das opções de resposta o usuário selecionou no quiz. Vamos analisar o código passo a passo:

### \*\*1. Definição da Função ``getSelected()``

```
```javascript
```

```
function getSelected(){
```

```
```
```

- Aqui estamos declarando a função `getSelected()`, que será responsável por verificar qual botão de rádio (opção de resposta) está selecionado pelo usuário e retornar essa informação.

### \*\*2. Declaração da Variável `answer`\*\*

```
```javascript
```

```
let answer;
```

```
```
```

- **`let answer`**: Uma variável chamada `answer` é declarada, mas inicialmente não é atribuída nenhum valor. Ela será usada para armazenar o **ID** da opção de resposta que o usuário selecionou.

### \*\*3. Loop Através das Opções de Resposta

```
```javascript
```

```
answersEls.forEach(answerEl => {
```

```
```
```

- **answersEls**: Esta é uma coleção de todos os elementos de botão de rádio do quiz, que têm a classe `.answer`.

- **forEach**: Este método percorre cada item (ou seja, cada botão de rádio) dentro de `answersEls`.

- **answerEl**: Durante o loop, cada elemento individual de resposta (botão de rádio) é referenciado pela variável `answerEl`.

### \*\*4. Verificando se o Botão Está Selecionado

```
```javascript
```

```
if(answerEl.checked){
```

```
```
```

- **answerEl.checked**: Cada botão de rádio tem uma propriedade chamada `checked` que é `true` se o botão foi selecionado, ou `false` se não foi.

- Esta linha verifica se o botão de rádio está selecionado. Se sim, o bloco de código dentro do `if` será executado.

### **\*\*5. Armazenando o ID da Resposta Selecionada\*\***

```
```javascript
```

```
answer = answerEl.id;
```

```
```
```

- Se a opção de resposta estiver marcada (`checked`), o **\*\*ID\*\*** desse elemento é atribuído à variável `answer`.

- Cada botão de rádio (opção de resposta) no HTML tem um ID (`a`, `b`, `c`, ou `d`). Esse ID é o que será armazenado na variável `answer`.

### **\*\*6. Retornando a Resposta Selecionada\*\***

```
```javascript
```

```
return answer;
```

```
```
```

- Depois de percorrer todas as opções de resposta, a função retorna o valor armazenado na variável `answer`. Se nenhuma opção tiver sido selecionada, `answer` permanecerá indefinida.

### **\*\*O Que a Função Faz:\*\***

1. **\*\*Percorre todas as opções de resposta\*\*** (botões de rádio) dentro de `answersEls`.

2. **\*\*Verifica qual botão de rádio foi marcado\*\*** pelo usuário, verificando se a propriedade `checked` é `true`.

3. **\*\*Armazena o ID da opção selecionada\*\*** (que será "a", "b", "c", ou "d") na variável `answer`.

4. **\*\*Retorna o ID\*\*** da resposta selecionada para o lugar onde a função foi chamada.

### ### \*\*Exemplo de Funcionamento:\*\*

Se o usuário selecionar a primeira opção de resposta (ID "a"), a função irá:

- Encontrar que o botão com o ID "a" está marcado (`checked`).
- Atribuir `"a"` à variável `answer`.
- Retornar `"a"` para que outras partes do código possam usar essa informação.

### ### \*\*Por que é importante:\*\*

A função `getSelected()` é fundamental para que o sistema de quiz saiba qual opção de resposta o usuário escolheu. Essa informação é usada para verificar se a resposta está correta e para continuar o quiz.