SENAI AVAK BEDOUIAN

Curso: Técnico em Desenvolvimento de Sistemas

Estudante: Heitor Kenzo Mitsuuti

LISTA DE QUESTÕES SOBRE JAVASCRIPT (JS)

06 de setembro de 2024
Birigui – SP

Sumário:

Atividade 1:

1º passo - Abri o “script”, para inserir o código em JavaScript dentro dele.

2º passo - Primeiro, eu criei um array vazio chamado "numeros", pois esse array vai armazenar os 5 números que o usuário irá inserir e também inicializei uma string chamada "resultado" com "Números digitados:\n", que servirá para armazenar a lista de números em uma formatação de texto. O “\n” utilizado no código é uma sequência que adiciona uma quebra de linha, ajudando a formatar a saída final para que cada número apareça em uma linha separada.

3º passo - Eu utilizo um loop usando o comando “for” que vai se repetir 5 vezes. A variável "valor" começa em 0 e vai até 4, permitindo que o loop execute 5 iterações. Com isso, dentro do loop que criei, eu uso prompt para pedir ao usuário que digite um número. O texto da caixa de diálogo inclui o número da iteração atual (de 1 a 5), que eu calculo como (valor + 1). A entrada do usuário é uma string, então eu a converto para um número inteiro usando parseInt e armazeno esse número na variável "numero".

4º passo - Depois, eu adiciono o número digitado ao array "numeros" usando o método push. Isso armazena todos os números fornecidos pelo usuário no array e atualizo a string "resultado" adicionando o número digitado e uma quebra de linha (\n). Assim, cada número digitado é adicionado à string e ficará em uma linha separada quando exibido.

5º passo - Após o loop terminar e eu ter adicionado todos os números à string "resultado", eu uso alert para exibir a string em uma caixa de diálogo. A caixa de diálogo mostra todos os números digitados pelo usuário, cada um em uma nova linha, de acordo com a formatação que eu defini.

1-)

```
<script>
        var numeros = []
        var resultado = "Números digitados:\n"

        for (var valor = 0; valor < 5; valor++) {
            var numero = parseInt(prompt("Digite o número " + (valor + 1) + ":"))
            numeros.push(numero)
            resultado += numero + "\n"
        }

        alert(resultado);
    </script>
```

Atividade 18:

1º passo - Abri o “script”, para inserir o código em JavaScript dentro dele.

2º passo - Criei uma variável (array) com a lista chamada “Votos” vazia, para armazenar os votos posteriormente.

3º passo - Criei uma variável totalVotos para manter o controle da quantidade total de votos computados.

4º passo - Defini três variáveis para armazenar informações sobre o jogador mais votado: melhorJogador (o número da camisa), melhorJogadorVotos (quantidade de votos recebidos) e melhorJogadorPercentual (percentual de votos).

5º passo - Aqui, criei um loop infinito com while (true) que continuará solicitando ao usuário o número do jogador votado. Depois, adicionei o código prompt() que captura a entrada do usuário e parseInt() converte essa entrada de string para um número inteiro.

6º passo - Se o usuário digitar "0", o loop será encerrado, pois break força a saída do while utilizando a condição “if”.

7º passo - Verifico se o número votado é inválido (fora do intervalo de 1 a 23). Se for inválido, imprimo um alerta avisando que o número informado não é válido e utilizo o comando “continue” para ignorar o restante do loop e reiniciar o ciclo, pedindo um novo voto.

8º passo - Se o número do voto for válido, adiciono esse voto ao array “votos” usando push(). Em seguida, incremento a variável “totalVotos” em 1, para registrar mais um voto.

9º passo - Usei um loop for para percorrer todos os votos registrados. Para cada iteração, a variável “jogador” recebe o número da camisa do jogador correspondente, e a variável “votosJogador” é definida como 0, para contar os votos desse jogador específico.

10º passo - Dentro deste segundo for, percorro novamente o array de “votos”. Se o voto encontrado for igual ao jogador atual, incremento “votosJogador” em 1, contando assim o número de votos para aquele determinado jogador.

11º passo - Após contar os votos do jogador atual, calculo o percentual de votos que ele recebeu em relação ao total. Divido a var “votosJogador” pelo “totalVotos” e multiplico por 100 para obter o percentual.

12º passo - Por meio do “if” verifico se o jogador atual recebeu mais votos do que o jogador mais votado até o momento. Se sim, atualizo as variáveis: “melhorJogador”, “melhorJogadorVotos” e “melhorJogadorPercentual” com os dados do jogador atual.

13º passo - Exibo um alerta mostrando o total de votos registrados e também a lista cada um dos votos.

14º passo - Na 57º linha do meu código, utilizei um comando “Set”, que é uma estrutura de dados que não permite elementos duplicados. O operador “...” (spread operator) cria um novo array com todos os elementos únicos. Ou seja, a variável “votosUnicos” contém apenas um registro de cada número de jogador votado.

- Qual a função do “...” (operador spread)?

R: A função do spread operator (...) é pegar todos os valores únicos que estão no Set e espalhá-los em um novo array. O Set, por si só, armazena valores únicos, mas não é um array. Para transformar o Set de volta em um array, o spread operator entra em ação, expandindo os elementos do Set para dentro do novo array.

- Qual a função do “new”?

R: Em JavaScript, a palavra-chave new é utilizada para instanciar objetos a partir de classes ou funções construtoras, ou seja, no caso do meu código, quando utilizo o new eu crio uma nova lista vazia para o Set com os valores dela, removendo automaticamente os valores duplicados (sem o “new” o “Set” não funcionaria, porque ele é uma classe, então ele precisa ser “criado” antes de ser usado ou chamado).

- Qual a função do “Set”?

R:Um Set é uma estrutura de dados, que armazena valores únicos, ou seja, ele automaticamente remove quaisquer elementos duplicados. Neste caso, ele está pegando todos os votos registrados no array “votos”, mas só mantendo um único exemplar de cada número de jogador, sem repetição.

- Como o comando: `[...new Set(votos)]` se aplica ao contexto do meu código JS?

R: Sem o “Set”, o código teria que lidar com jogadores repetidos várias vezes. Ao invés de verificar um jogador várias vezes (cada vez que ele apareceu em votos), eu crio o array votosUnicos e depois faço o cálculo de votos para cada jogador único apenas uma vez.

15º passo - Agora, para cada jogador que recebeu votos, percorro a lista “votosUnicos”. Defino jogador como o número do jogador e inicializo “votosJogador” com 0 para contar quantos votos ele recebeu.

16º passo - Abro um novo “for” para repetir o processo de contagem dos votos para o jogador atual, comparando cada voto no array “votos” com o número do jogador e incrementando “votosJogador” quando houver correspondência.

17º passo - Depois de contar os votos, calculo novamente o percentual de votos que o jogador recebeu.

18º passo - Exibo um alerta para mostrar quantos votos o jogador recebeu e qual foi o percentual em relação ao total de votos, mas para gerenciar melhor os valores na hora em que forem impressos, eu utilizei o comando “.toFixed()” porque posso definir quantas casas decimais poderão aparecer e neste caso defini apenas uma casa.

19º passo - Por fim, exibo o resultado da votação, informando qual foi o melhor jogador, quantos votos ele recebeu e qual foi o percentual de votos em relação ao total.

Fontes de pesquisa (referências bibliográficas):

- https://www.devmedia.com.br/o-que-sao-arrays-e-como-utiliza-los-no-javascript/43563

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide

- https://vidafullstack.com.br/javascript/new-set-com-javascript/#:~:text=O%20new%20Set()%20serve,para%20ordenar%20de%20forma%20crescente.

- Ajuda do prof. Igor

- https://chatgpt.com/

ChatGPT

e qual é a função do "..." que é o spread operator?

O operador `...`, conhecido como **spread operator**, tem a função de "espalhar" ou "expandir" os elementos de uma estrutura iterável, como arrays, objetos ou sets, em JavaScript.

No contexto do código:

```javascript
[...new Set(votos)]
```

A função do **spread operator** (`...`) é pegar todos os valores únicos que estão no `Set` e **espalhá-los** em um novo array. O `Set`, por si só, armazena valores únicos, mas não é um array. Para transformar o `Set` de volta em um array, o spread operator entra em ação, expandindo os elementos do `Set` para dentro do novo array.

Resumo da função do spread operator:

- **Expande** os elementos de uma estrutura iterável (como um array, set ou objeto).
- **Converte** estruturas como o `Set` (que só aceita valores únicos) de volta para arrays, que podem ser manipulados mais facilmente.

Exemplo aplicado no código:

```javascript
var votosUnicos = [...new Set(votos)];
```

Mensagem ChatGPT

O ChatGPT pode cometer erros. Considere verificar informações importantes.

18-)

```
<script>
    var votos = []
    var totalVotos = 0
    var melhorJogador = 0
    var melhorJogadorVotos = 0
    var melhorJogadorPercentual = 0

    while (true) {
        var voto = parseInt(prompt("Insira o número do jogador e digite 0 para encerrar): "))

        if (voto == 0) {
            break
        }

        if (voto < 1 || voto > 23) {
            alert("Número de camisa inválido, informe um valor entre 1 e 23 ou 0 para sair!")
            continue
        }
        votos.push(voto)
        totalVotos++
    }

    for (var num = 0; num < votos.length; num++) {
        var jogador = votos[num]
        var votosJogador = 0

        for (var quantidade = 0; quantidade < votos.length; quantidade++) {
            if (votos[quantidade] == jogador) {
                votosJogador++
            }
        }

        var percentual = (votosJogador / totalVotos) * 100

        if (votosJogador > melhorJogadorVotos) {
            melhorJogador = jogador
            melhorJogadorVotos = votosJogador
            melhorJogadorPercentual = percentual
        }
    }

    alert("Foram computados " + totalVotos + " votos.")

    alert(votos)

    // Cria uma lista removendo os numeros repitos na lista votos
    // ... o que significa?
    // o que é new ?
    // o que é Set ?
    var votosUnicos = [...new Set(votos)];

    for (var num = 0; num < votosUnicos.length; num++) {
        var jogador = votos[num]
        var votosJogador = 0

        for (var quantidade = 0; quantidade < votos.length; quantidade++) {
            if (votos[quantidade] == jogador) {
                votosJogador++
            }
        }

        var percentual = (votosJogador / totalVotos) * 100

        alert("Jogador " + jogador + ": " + votosJogador + " votos (" + percentual.toFixed(1) + "%)")
    }

    alert("Resultado da votação: O melhor jogador foi o número " + melhorJogador + ", com " +
     melhorJogadorVotos + " votos, correspondendo a " + melhorJogadorPercentual.toFixed(1) + "% do total de votos.")
</script>
```

Atividade 20:

1º passo - Abri o “script”, para inserir o código em JavaScript dentro dele.

2º passo - Logo de começo, eu criei dois arrays vazios (listas), um chamado "salarios" para armazenar os salários dos funcionários e o outro chamado "abonos" (o abono é um benefício providenciado a alguma pessoa ou entidade que tenha direitos adquiridos) para armazenar os abonos calculados para cada funcionário.

3º passo - Inicializo a variável "totalGasto" com zero, pois é ela que irá acumular o total gasto com abonos.

4º passo - Defino o valor mínimo de abono como 100 reais e armazeno na variável "abonoMin".

5º passo - Inicio a variável "funcionariosMin" com zero para contar quantos funcionários receberam o abono mínimo.

6º passo - Também inicializo a variável "maiorAbono" com zero para armazenar o maior valor de abono pago.

7º passo - Início um loop infinito por meio do “while (true)” que irá continuar até que eu decida parar com uma condição interna e logo em seguida, peço ao usuário para inserir o salário do funcionário através de um “prompt” e converto a entrada para um número decimal com parseFloat.

8º passo - Por meio do “if (salario == 0):” eu verifico se o salário inserido é 0, assim essa condição indica que o usuário quer encerrar o loop, ou seja, se o número inserido for 0 o loop é finalizado com o comando “break”.

9º passo - Adiciono o salário inserido ao array "salarios", utilizando o comando “.push”.

10º passo - Calculo o abono como 20% do salário e o armazeno na variável "abono".

11º passo - Verifico se o abono é menor que o valor mínimo e se caso seja menor, ajusto o abono para o mínimo e aumento o contador de funcionários que receberam o mínimo. Depois adiciono o abono ao array "abonos" e atualizo "totalGasto" com o valor do abono e assim, se o abono é maior que o "maiorAbono" registrado, atualizo "maiorAbono".

12º passo - Inicializo a string "resultado" com o cabeçalho "Salário - Abono". Após isso, eu utilizei um loop para percorrer todos os salários e abonos, formatando cada par com duas casas decimais usando o comando “.toFixed()”, além de cada linha do relatório ser adicionada a "resultado" com uma quebra de linha “\n” no final de cada entrada. Dessa forma, exibo o resultado final com um alert.

13º passo - Por fim, coloqueis os alertas, onde:

"salarios.length" exibe o número total de colaboradores que tiveram seus salários processados.

"totalGasto.toFixed(2)" formata o total gasto com abonos com duas casas decimais. Exibo o total gasto com abonos.

"funcionariosMin" mostra quantos funcionários receberam o valor mínimo de abono.

"maiorAbono.toFixed(2)" formata o maior valor de abono com duas casas decimais e exibe essa informação.

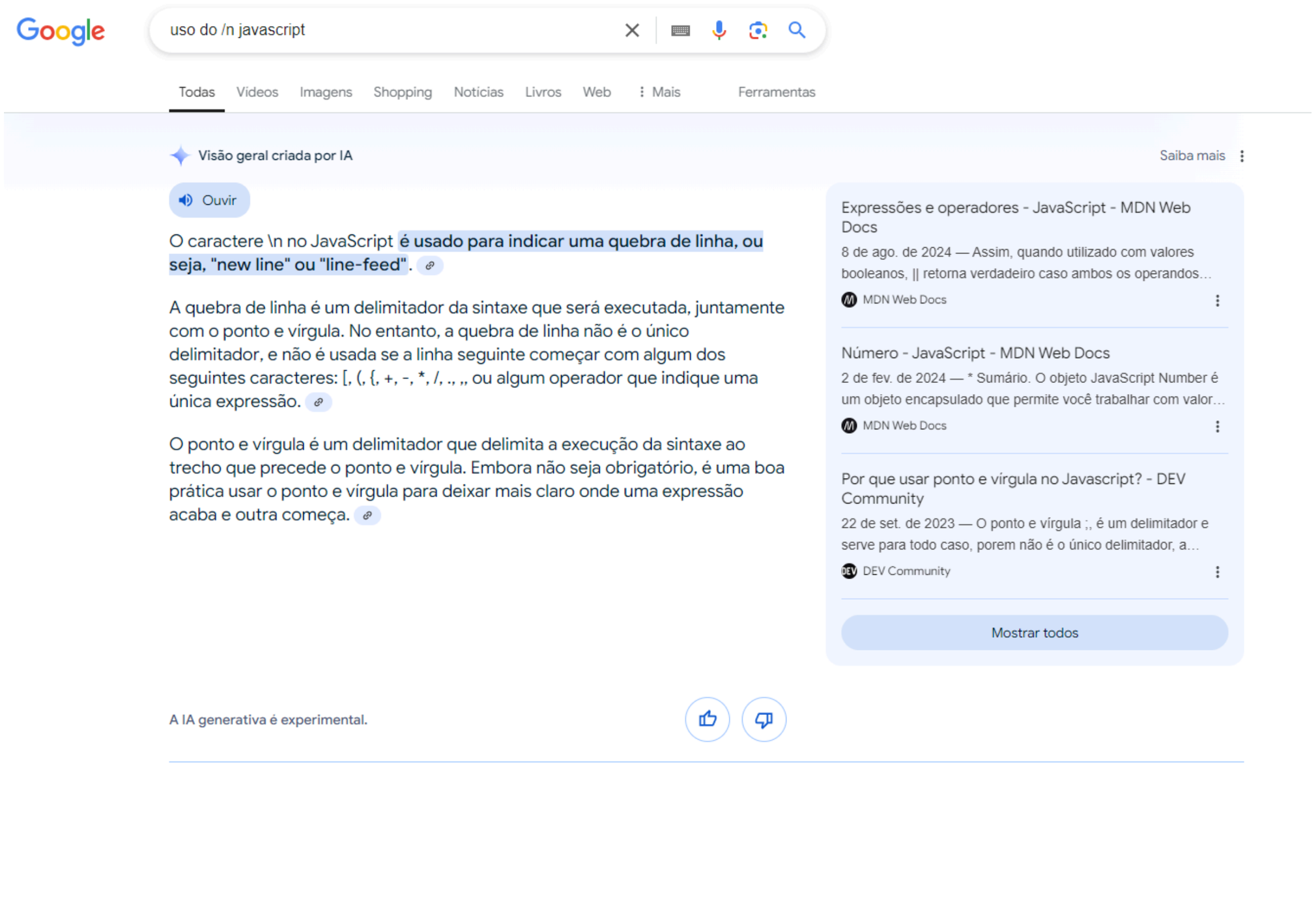
Fontes de pesquisa (referências bibliográficas):

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide

- https://www.oitchau.com.br/blog/abono-salarial-o-que-e-e-quem-tem-direito/#:~:text=Existem%20dois%20tipos%20de%20abono,%C3%A9%20destinado%20aos%20servidores%20p%C3%BAblicos.

- Ajuda do prof. Igor

- Gemini (IA do Google)



20-)

```
<script>
    var salarios = []
    var abonos = []
    var totalGasto = 0
    var abonoMin = 100
    var funcionariosMin = 0
    var maiorAbono = 0

    while (true) {
        var salario = parseFloat(prompt("Insira o salário do funcionário ou digite 0 para encerrar: "))

        if (salario == 0) {
            break
        }

        salarios.push(salario)

        var abono = salario * 0.2
        if (abono < abonoMin) {
            abono = abonoMin
            funcionariosMin++
        }

        abonos.push(abono)
        totalGasto += abono

        if (abono > maiorAbono) {
            maiorAbono = abono
        }
    }

    var resultado = "Salário - Abono\n"
    for (var valor = 0; valor < salarios.length; valor++) {
        resultado += "R$ " + salarios[valor].toFixed(2) + " - R$ " + abonos[valor].toFixed(2) + "\n"
    }
    alert(resultado)

    alert("Foram processados " + salarios.length + " colaboradores.")

    alert("Total gasto com abonos: R$ " + totalGasto.toFixed(2))

    alert("Valor mínimo pago a " + funcionariosMin + " colaboradores.")

    alert("Maior valor de abono pago: R$ " + maiorAbono.toFixed(2))
</script>
```

Atividade 21:

1º - Abri o “script”, para inserir o código em JavaScript dentro dele.

2º - Eu criei uma variável array chamado modelos e coloquei os nomes dos carros que quero analisar: "fusca", "gol", "uno", "vectra" e "peugeot".

3º - Em seguida, eu criei outro array (lista), chamado “consumos” para armazenar o consumo de cada carro em quilômetros por litro. Cada valor corresponde ao consumo dos carros na mesma ordem que a variável array “modelos”.

4º - Defini o preço da gasolina por litro como R$ 2,25 e armazenei esse valor na variável “precoGasolina”.

5º - Iniciei um for que vai percorrer todos os carros. O “valores_lista” é o índice que começa em 0 e vai até o comprimento do array modelos (percorre cada string ou número).

6º - Dentro do for, calculei a quantidade de litros que o carro atual precisa para percorrer 1000 km. Usei o método toFixed(1) para arredondar o resultado para apenas uma casa decimal.

7º -

CONCLUSÃO

Com base nos 24 exercícios realizados, sob as instruções do professor Igor Cacerez, pode-se concluir que JavaScript é uma ferramenta poderosa e versátil para desenvolvimento web. A prática contínua e a compreensão dos conceitos fundamentais são essenciais para dominar essa linguagem e aplicar suas capacidades para criar soluções interativas e eficientes. Estou ansioso para continuar explorando e aprendendo mais sobre JavaScript e suas diversas aplicações.