

Atividade 1:

A função `exibirDobro` recebe um parâmetro chamado `num` e calcula o dobro desse número multiplicando-o por 2, retornando o resultado. Utilizo um loop `for` que itera três vezes, com a variável `i` controlando as repetições. Dentro do loop, crio uma variável chamada `numero` que recebe um valor digitado pelo usuário através de um prompt, onde o usuário é solicitado a digitar um número correspondente à sua iteração (1, 2 ou 3). Após obter o número, chamo a função `exibirDobro`, passando o número digitado como argumento e armazeno o resultado na variável `dobro`. Por fim, exibo um alerta informando ao usuário qual é o dobro do número que ele digitou.

Atividade 2:

A função `exibirAoQuadrado` recebe dois parâmetros, `num` e `num2`, e calcula o resultado de `num` elevado a `num2` usando o operador de exponenciação (**), retornando o valor. Em seguida, crio duas variáveis, `numero1` e `numero2`, que armazenam os valores inseridos pelo usuário por meio de prompts. Esses prompts solicitam ao usuário que digite dois números. Após obter os dois números, chamo a função `exibirAoQuadrado`, passando `numero1` e `numero2` como argumentos, e armazeno o resultado na variável `elevado`. Por fim, exibo um alerta que mostra o cálculo realizado, informando o usuário sobre o resultado da operação de exponenciação.

Atividade 3:

A função `criarNumero` recebe dois números, `num1` e `num2`, e extrai o primeiro dígito de `num1` e o segundo dígito de `num2`, convertendo-os em strings para facilitar o acesso aos dígitos. A função então combina esses dígitos e os converte de volta para um número inteiro usando `parseInt`, retornando o resultado. Após definir a função, duas variáveis, `numero1` e `numero2`, armazenam os valores que o usuário insere por meio de prompts, solicitando que digite números entre 10 e 99. Em seguida, há uma verificação para garantir que ambos os números estão dentro do intervalo permitido. Se a condição for verdadeira, chamo a função `criarNumero`, passando `numero1` e `numero2` como argumentos, e armazeno o resultado em `numero3`. Por fim, exibo um alerta informando o novo número gerado a partir dos dígitos escolhidos. Se os números não forem válidos, um alerta informa o usuário a digitar um número correto.

Atividade 4:

A função `escolhaDeNotas` recebe três notas e um tipo de média como parâmetros. Quando o tipo é "A", ela calcula a média aritmética, que é feita somando as três notas e dividindo o resultado por 3. Se o tipo for "P", a função calcula a média ponderada, onde a primeira nota tem peso 5, a segunda peso 3 e a terceira peso 2, dividindo o total por 10. Em seguida, eu pedi ao usuário para digitar as três notas usando prompts e para escolher o tipo de média, digitando "A" para a média aritmética ou "P" para a média ponderada. Depois, verifiquei qual tipo foi informado. Se for "A", eu chamei a função `escolhaDeNotas` e exibi o resultado da média em um alerta, formatado para mostrar duas casas decimais. O mesmo processo acontece se o tipo for "P". Se o usuário digitar um tipo inválido, eu mostro um alerta dizendo que o tipo de média é inválido, pedindo que use "A" ou "P" para escolher corretamente.

Atividade 5:

A função `calcularOperacao` recebe dois valores e um símbolo que representa a operação desejada. Se o símbolo for "+", a função soma os dois valores e retorna o resultado. Se for "-", subtrai o segundo valor do primeiro. Se for "/", realiza a divisão do primeiro valor pelo segundo. Se for "*", multiplica os dois valores. Caso o símbolo não corresponda a nenhuma operação válida, a função retorna "Operação

inválida". Depois, eu pedi ao usuário para digitar os dois valores e o símbolo da operação por meio de prompts. Em seguida, chamei a função `calcularOperacao` com esses valores e armazenei o resultado. No entanto, a condição que verifica se o resultado é uma operação válida está incorreta, pois está comparando o resultado com os símbolos em vez de verificar se o resultado é um número. Se o resultado for válido, um alerta exibe o resultado da operação. Se não for, um alerta mostra a mensagem de operação inválida.

Atividade 6:

A função `verDiasRestantes` recebe o número de dias restantes e verifica quantos dias faltam para o fim do ano. Se o número de dias for maior que 30, um alerta informa que ainda falta muito para o Natal. Se for menor que 30, outro alerta diz que está quase na hora das férias. Eu pedi ao usuário para digitar quantos dias restam para o fim do ano através de um prompt e armazenei esse valor em uma variável chamada `diasRestantes`. Em seguida, chamei a função `verDiasRestantes` passando essa variável como argumento. Dessa forma, o usuário recebe um feedback baseado na quantidade de dias restantes, permitindo uma expectativa sobre as próximas festas.

Atividade 7:

Acredito que nem precisa explicar essa função, apenas criei uma função com o alert, e depois chamei a função.

Atividade 8:

É a mesma coisa da atividade anterior, apenas criei uma função com o alert, e depois chamei a função.

Atividade 9:

A função `qtdDeDigitos` recebe um número como entrada e calcula a quantidade de dígitos que ele possui, utilizando a propriedade `length` da string que representa o número. Eu pedi ao usuário para digitar um número inteiro através de um prompt e armazenei essa entrada na variável `numero`. Em seguida, chamei a função `qtdDeDigitos`, passando o número como argumento, e armazenei o resultado na variável `resultado`. Por fim, exibi um alerta mostrando a quantidade de dígitos do número informado, permitindo que o usuário saiba rapidamente quantos dígitos ele digitou. Essa abordagem simples facilita a interação e fornece uma resposta imediata.

Atividade 10:

A função `reversoNumero` recebe um número inteiro como entrada e calcula o seu reverso. Primeiro, converto o número em uma string usando `toString`, em seguida, uso `split` para dividir a string em um array de caracteres, aplico `reverse` para inverter a ordem dos caracteres e, por fim, uso `join` para juntar os caracteres de volta em uma string. O resultado é então convertido de volta para um número inteiro com `parseInt`. Eu pedi ao usuário para digitar um número inteiro através de um prompt e armazenei essa entrada na variável `numero`. Depois, chamei a função `reversoNumero`, passando o número como argumento, e armazenei o resultado na variável `resultado`. Por último, exibi um alerta mostrando o número original e seu reverso, permitindo que o usuário veja rapidamente o resultado da operação. Essa implementação é uma maneira eficaz de manipular strings e números em JavaScript.

Atividade 11:

Atividade 12:

A função `dataMesAno` recebe uma data no formato DD/MM/AAAA e a transforma em uma string mais legível. Primeiro, eu divido a string usando o método `split`, separando os elementos por barra (/). Se o

número de elementos obtidos não for igual a 3, retorno null para indicar que a data é inválida. Em seguida, eu converto cada parte da data (dia, mês e ano) em números inteiros. Caso algum valor seja NaN ou se o dia for menor ou igual a 0 ou se o mês estiver fora do intervalo de 1 a 12, retorno null novamente. Depois, criei um array chamado meses que contém os nomes dos meses em português. Por fim, retorno uma string formatada que exibe a data por extenso. Eu pedi ao usuário para inserir uma data por meio de um prompt e armazenei essa entrada na variável data. Depois, chamei a função dataMesAno com a data digitada e armazenei o resultado. Se o resultado não for null, exibo um alerta mostrando a data por extenso; caso contrário, informo que a data é inválida, garantindo que o usuário receba um feedback claro sobre a entrada.

Atividade 13:

A função embaralhaPalavra recebe uma palavra e a transforma em uma string, se necessário. Primeiro, uso o método split para dividir a palavra em um array de caracteres. Em seguida, aplico o método sort com uma função que gera um número aleatório para reordenar os caracteres de forma aleatória. Depois, uso o método join para unir os caracteres novamente em uma única string. No código, a variável resultado deveria armazenar o retorno da função chamada com a palavra, mas a variável palavra não foi definida antes de ser usada. Para que o código funcione, deveria ter adicionado um prompt para solicitar ao usuário que digitasse uma palavra e armazenado essa entrada na variável palavra. Por fim, o resultado é exibido em um alerta mostrando a palavra embaralhada.

Atividade 14:

A função formulaBhaskara recebe três parâmetros: a, b e c, que representam os coeficientes de uma equação quadrática. No início, verifico se o valor de a é igual a zero; se for, exibo um alerta informando que esse valor não pode ser zero e retorno null. Em seguida, calculo o discriminante (delta) utilizando a fórmula $b^2 - 4ac$. Se delta for menor que zero, retorno uma mensagem indicando que não existem raízes reais. Caso contrário, calculo as duas raízes da equação usando as fórmulas para a raiz positiva e negativa, que são $(-b \pm \sqrt{\text{delta}}) / 2a$. O resultado das raízes é retornado como um array. Depois, chamo a função com os valores 2, -5 e -7, armazenando o resultado na variável resultado. Por fim, exibo um alerta mostrando as raízes calculadas a partir do array resultado, acessando as posições correspondentes.

Atividade 15:

A função pitagoras recebe dois parâmetros, cateto1 e cateto2, que representam os comprimentos dos catetos de um triângulo retângulo. Dentro da função, calculo a hipotenusa usando o teorema de Pitágoras, que é a raiz quadrada da soma dos quadrados dos catetos. Para isso, aplico Math.sqrt para obter a raiz quadrada do resultado da expressão $(\text{cateto1}^2 + \text{cateto2}^2)$ e retorno esse valor. Em seguida, crio duas variáveis, cateto1 e cateto2, que recebem os valores digitados pelo usuário através de prompts, convertidos para números inteiros com parseInt. Chamo a função pitagoras passando os valores dos catetos e armazeno o resultado na variável resultado. Por fim, exibo um alerta com o valor da hipotenusa calculada.

Atividade 16:

Basicamente, é igual às atividades 7 e 8, apenas joguei um enorme dentro da função, depois chamei ela, assim finalizando.

Eu utilizei fontes como: Chat-GPT, colegas de classe (Ygor, Pedro, Heitor Kenzo, Heitor Prado), o professor (Igor Cacerez), Sites online (Stack Over Flow, MDN Web Docs, Alura e GitHub), porém utilizei como um recurso de aprendizado, e algumas questões consegui fazer sozinho.

Muito Obrigado!