

Lista de exercícios

LISTA

Exercício 1

Primeiramente, foi criada uma função contendo um parâmetro `num`. Dentro dela, realizou-se uma variável com o `let` com a fórmula para a soma, no caso “`num + num`” que seria o dobro do valor posteriormente mencionado. Em seguida, utilizou-se o `return` para devolver a variável do dobro. Logo, fora do `function`, foram criadas novas variáveis para os 3 números, a fim de invocar a função. Por fim, o `alert` dos resultados para cada número específico.

Exercício 2

Primeiramente, foi criada uma função contendo dois parâmetros: `X` e `Y`. Em seguida, realizou-se dentro dela uma variável com a fórmula para elevar, utilizando o comando `**` para isso. Em seguida, utilizou-se o `return` para devolver a variável do cálculo. Logo, fora do `function`, foi criada uma nova variável, contendo os números 2 e 3 para elevá-los. Por fim, o `alert` do resultado.

Exercício 3

Primeiramente, foi criada uma função contendo o `n1` e o `n2`. Dentro dela, criou-se duas variáveis, uma para o primeiro dígito e outra para o segundo. Nesta variável, utilizou-se o comando `to.string()` para transformar os valores em strings, facilitando o processo. Logo, em frente o comando, foi colocado os valores 0 e 1, respectivamente, mostrando a posição a qual o número deveria ser pegado (no caso, o primeiro elemento que é a posição 0 e o segundo que é a posição 1). Logo, retornou-se a soma desses elementos como número novamente, utilizando o comando `parseInt`. Em seguida, foi invocada a função, mencionando os valores 23 e 89 e por fim, o `alert` exibindo o resultado.

Exercício 4

Primeiramente, foram criadas quatro variáveis, sendo três para as notas e a outra para a letra correspondente. Logo, foi realizada a função do cálculo 1, contendo a fórmula para somar as notas e dividir por 3 (média aritmética, sendo 3 a quantidade de notas). Este número foi devolvido através do `return`. Este mesmo processo ocorreu com a média ponderada, sendo uma função para o cálculo 2, contendo a fórmula para multiplicar as notas com os respectivos pesos e dividir pela soma deles (pesos, que seria 10). Este número também foi devolvido através do `return`. Em seguida, foram estabelecidas as

condições (*if* e *else if*), a primeira para se caso for “A” (invocando a primeira função e exibindo através do *alert*) e a segunda se caso for “P” (invocando a segunda função e exibindo através do *alert*). Por fim, se caso o valor informado for diferindo dos estipulados, o programa exibe “Valor inválido” através do *else*.

Exercício 5

Primeiramente, foram criadas três variáveis, duas para informar os valores com *parseFloat(prompt)* (porque o número pode ser quebrado) e outra para informar o símbolo, apenas com o *prompt* pois é uma string. Logo, criou-se uma única *function* contendo as condições dentro dela. O primeiro para soma, o segundo para a subtração, o terceiro para a divisão e o último para a multiplicação. Todas essas variáveis possuem o mesmo nome (“resultado”). Em seguida, o *return* para devolver o resultado. Por fim, um *alert*, já invocando os valores e exibindo eles.

Exercício 6

Primeiramente, foi criada uma *function* para calcular qual seria a resposta do usuário. Dessa forma, dentro dela foram criadas as condições, se caso for maior que 30, exibir que falta muito tempo, se caso for o contrário, exibir que as férias estão chegando, criando uma variável para o resultado com a string. Fora do *if*, exibir o resultado com o *alert*. Por fim, fora do *function*, foi necessário criar a variável para pedir ao usuário a quantidade de dias e chamar a função com o comando *calcular(num1)* que é o nome dela.

Exercício 7

Primeiramente, criou-se uma *function* vazia chamada “calcular”. Dentro dela, foi criada uma variável chamada “frase”, contendo uma string (“Ufa! Este foi o último exercício! Acabei =D”). Ainda dentro do *function*, foi inserido o *alert* com a finalidade de exibir a frase. Por fim, fora da função, para substituir o *return*, foi chamada a função com o seu respectivo “nome” e valor (no caso vazia).

Exercício 8

Primeiramente, criou-se uma *function* vazia chamada “calcular”. Dentro dela, foi criada uma variável chamada “frase”, contendo uma string (“Fui Trolado :("). Ainda dentro do *function*, foi inserido o *alert* com a finalidade de exibir a frase. Por fim, fora da função, para substituir o *return*, foi chamada a função com o seu respectivo “nome” e valor (no caso vazia).

Exercício 9

Primeiramente foi criada a *function* a fim de analisar o valor informado. Dentro dela, foi criada uma variável chamada "string" para transformar o valor para uma string. Para isso, foi utilizado o comando `String(num1)` (este comando foi ensinado pelo professor). Logo, ainda dentro da *function*, foi exibido o resultado com o *alert* e, para separar a string utilizou-se o comando *length*. Fora da *function*, criou-se uma nova variável para o usuário informar o valor com *parseFloat* para números quebrados. Por fim, a função foi chamada com seu respectivo nome.

Exercício 10

Primeiramente criou-se uma *function* contendo variáveis dentro dela. A primeira, foi para transformar o número posteriormente mencionado em string. A segunda, para separar os números da variável anterior, utilizando o comando *split("")*. A terceira, para inverter os valores, utilizando o comando *reverse()*. Logo, criou-se uma variável vazia para posteriormente ser utilizada para contar. Em seguida, ainda dentro do *function*, foi criado o *for* para percorrer o valor e concatenar eles de acordo com a posição (i). Para devolver o valor, utilizou-se o *return*. Por fim, fora do *function*, foi criado a variável para o usuário inserir o valor e o *alert* para invocar a a função e exibir o resultado.

Exercício 11

função dados, são gerados e retornados resultados de dois lançamentos de dados, somando um número aleatório de 1 a 6 para cada. A função jogo faz o primeiro lançamento e exibe o resultado. Se o total for 7 ou 11, o jogador ganha, e se for 2, 3 ou 12, ele perde. Para outros resultados, um "ponto" é definido, e o jogador deve lançar os dados novamente. Em um loop, ele continua jogando até que consiga tirar o mesmo "ponto" e ganhar ou tirar um 7 e perder. Por fim, o resultado final do jogo é mostrado em um alerta.

Exercício 12

A função *dataMesAno* divide a data em dia, mês e ano, verificando se a entrada está correta. Se a data não tiver três partes ou se os valores forem inválidos (como mês fora do intervalo ou não numéricos), retorna null. Caso contrário, converte o mês para o nome correspondente em português e retorna a data no formato "dia de mês de ano". O usuário é solicitado a inserir a data através de um prompt, e o resultado é exibido em um alerta; se a data for inválida, uma mensagem de erro é mostrada.

Exercício 13

A função `embaralhaPalavra` recebe uma palavra, converte-a em uma lista de caracteres usando `split`, embaralha essa lista com `sort` e, em seguida, junta os caracteres novamente com `join` para formar a palavra embaralhada. Em seguida, o `alert` para exibir.

Exercício 14

Primeiramente, criei uma *function* com uma variável para calcular o valor dentro da raiz quadrada. Se caso o número for negativo, ou seja, menor que 0, o código acaba. Em seguida, a fórmula para calcular a raiz quadrada, com `** 0.5`. Em seguida para realizar a divisão foi criada outra variável. O mesmo processo foi realizado com a conta negativa. Em seguida, foi disposto para o usuário inserir os valores dos coeficientes e o `alert` e a variável para invocar e exibir.

Exercício 15

Primeiramente, criei uma *function* com uma variável para calcular o valor dentro da raiz quadrada. Se caso o número for negativo, ou seja, menor que 0, o código acaba, pois não existe raiz quadrada negativa. Em seguida, a fórmula para calcular a raiz quadrada, com `** 0.5`. Em seguida para realizar a divisão foi criada outra variável. Em seguida, foi disposto para o usuário inserir os valores e o `alert` e a variável para invocar e exibir.

Exercício 16

Primeiramente, criou-se uma *function* vazia chamada “calcular”. Dentro dela, foi criada uma variável chamada “frase”, contendo uma string verdadeira. Ainda dentro do *function*, foi inserido o `alert` com a finalidade de exibir a frase. Por fim, fora da função, para substituir o `return`, foi chamada a função com o seu respectivo “nome” e valor (no caso vazia).