

**Curso Web Moderno - Exercícios Javascript**

1. Crie uma função que dado dois valores (passados como parâmetros) mostre no console a soma, subtração,​ multiplicação e divisão desses valores.

function Calculadora(num1, num2){

    const soma = num1 + num2

    console.log("A soma do",num1," + ",num2," = ", soma)

    const sub = num1 - num2

    console.log(`A subtração do ${num1} - ${num2} = ${sub}`)

    const multi = num1 \* num2

    console.log(`A multiplicação do ${num1} X ${num2} = ${multi}`)

}

Calculadora(1,2)

Calculadora(10,10)

1. Os triângulos podem ser classificados em 3 tipos quanto ao tamanho de seus lados:​

Equilátero: Os três lados são iguais. Isósceles: Dois lados iguais. Escaleno: Todos os lados são diferentes. Crie uma função que recebe os comprimentos dos três lados de um triângulo e retorne sua classificação quanto ao tamanho de seus lados. (Neste exemplo deve-se abstrair as condições matemáticas de existência de um triângulo).

function Triangulo(lado1, lado2,lado3){

    if(lado1 === lado2 && lado2 ===lado3 ){

        return "Esse triangulo é Equilatero"

    }

    else if(lado1 === lado2 || lado2 === lado3 || lado1=== lado3)

    {

        return "Esse triangulo é Isósceles"

    }

    else{

        return "Esse triangulo é Escaleno"

    }

}

console.log(Triangulo(1,2,3))

console.log(Triangulo(1,2,2))

console.log(Triangulo(2,2,2))

1. Crie uma função que recebe dois parâmetros, base e expoente, e retorne a base elevada ao expoente.​

function elevarBase(base, expoente){

   return Math.pow(base, expoente )

}

console.log(elevarBase(2,3))

function elevarBase2(base, expoente){

    return base \*\* expoente

 }

 console.log(elevarBase2(2,3))

1. Crie uma função que irá receber dois valores, o dividendo e o divisor. A função deverá imprimir o resultado​ e o resto da divisão destes dois valores.

function Divisor(dividendo, divisor){

    if(dividendo >= divisor)

    {

        return dividendo/divisor

    }

    else{

        return "Imposivel Operar"

    }

}

console.log(Divisor(1,2))

console.log(Divisor(2,2))

1. Lidar com números em JavaScript pode dar muita dor de cabeça. Você já viu o que acontece quando faz o​ seguinte comando no console: console.log(0.1 + 0.2); O resultado será: 0.30000000000000004. Outra coisa importante de observar, é o fato que o ponto é utilizado no lugar da vírgula e vice versa. Com isso, vamos fazer um exercício simples para mostrar dinheiro sempre da forma correta. Desenvolva uma função JavaScript para que ela receba um valor como 0.30000000000000004 e retorne R$0,30 (observe a vírgula e o ponto).

function ArrumaVirgula(num1 = 2, num2 = 4)

{

const resposta =  num1 + num2

const respostaformat =  resposta.toFixed(2).replace(".", ",")

return respostaformat

}

console.log(ArrumaVirgula())

1. Elabore duas funções que recebem três parâmetros: capital inicial, taxa de juros e tempo de aplicação. A​ primeira função retornará o montante da aplicação financeira sob o regime de juros simples e a segunda retornará o valor da aplicação sob o regime de juros compostos.

function jurosSimples(capital, juros, tempo){

    let Total = capital \* juros \* tempo

    return Total

}

function jurosCompostos(capital, juros,tempo){

    let Total = capital \* ((1 + juros)\*\*tempo)

    return Total

}

console.log(jurosSimples(1000,1,2))

console.log(jurosCompostos(1000,1,2))

1. Uma das vantagens da programação é a automatização de tarefas que não gostamos de realizar. Dito isto,​ elabore uma função cujo objetivo é resolver a fórmula de Bhaskara. Para isso, sua função deve receber três parâmetros, “ax2”, “bx” e “c”, de tal modo que na equação: 3x² - 5x + 12 os valores seriam respectivamente: 3, -5, 12. Como retorno deve ser passado um vetor que tem 2 valores um para cada possível resultado, mesmo que os resultados sejam iguais. Caso o delta seja negativo, retorne, ao invés do vetor, um string com a frase:

“Delta é negativo”.

1. Pedro joga N jogos de basquete por temporada. Para saber como está ele está progredindo, ele mantém​ registro de todos os as pontuações feitas por jogo. Após cada jogo ele anota no novo valor e confere se o mesmo é maior ou menor que seu melhor e pior desempenho. Dada uma lista string = “pontuação1 pontuação2 pontuação3 etc..”, escreva uma função que ao recebê-la irá comparar os valores um a um e irá retornar um vetor com o número de vezes que ele bateu seu recorde de maior número de pontos e quando fez seu pior jogo. (Número do pior jogo).

Obs.: O primeiro jogo não conta como novo recorde do melhor.

Exemplo:

String: “10 20 20 8 25 3 0 30 1”

Retorno: [3, 7] (Significa que ele bateu três vezes seu recorde de melhor pontuação e a pior pontuação aconteceu no sétimo jogo.)

1. Construa uma função para um sistema de notas de uma instituição que possui a seguinte política de​ classificação: Todo aluno recebe uma nota de 0 a 100. Alunos com nota abaixo de 40 são reprovados. As notas possuem a seguinte regra de arredondamento: Se a diferença entre a nota e o próximo múltiplo de 5 for menor que 3, arredondar a nota para esse próximo múltiplo de 5. Se a nota for abaixo de 38, não é feito nenhum arredondamento pois esta nota resulta na reprovação do aluno. Por exemplo, a nota 84 será arredondada para 85, mas a nota 29 não será arredondada por ser abaixo de 40 e não ser possível arredondamento eficiente, ou seja, que evite a reprovação do aluno. No caso de a nota ser 38, o arredondamento é possível pois atingirá 40 e o aluno será aprovado.

1. Crie uma função que verifica se um número inteiro passado como parêmetro é divisível por 3 e retorne true​ ou false.

1. As regras para o cálculo dos anos bissextos são as seguintes:​

De 4 em 4 anos é ano bissexto;

De 100 em 100 anos não é ano bissexto;

De 400 em 400 anos é ano bissexto;

Prevalecem as últimas regras sobre as primeiras.

Partindo daí elabore uma função que recebe um ano e calcula se ele é ano bissexto, imprimindo no console a mensagem e retornando true ou false.

1. Faça um algoritmo que calcule o fatorial de um número.​

1. Crie um programa que exibe se um dia é dia útil, fim de semana ou dia inválido dado o número referente ao​ dia. Considere que domingo é o dia 1 e sábado é o dia 7. Utilize a estrutura Switch.

1. Crie uma estrutura condicional switch que receba uma string com o nome de uma fruta e que possua três​ casos: Caso maçã, retorne no console: “Não vendemos esta fruta aqui”. Caso kiwi, retorne: “Estamos com escassez de kiwis”. Caso melancia, retorne: “Aqui está, são 3 reais o quilo”. Teste com estas três opções .Crie também um default, que retornará uma mensagem de erro no console.

1. Um homem decidiu ir à uma revenda comprar um carro. Ele deseja comprar um carro hatch, e a revenda​ possui, além de carros hatch, sedans, motocicletas e caminhonetes. Utilizando uma estrutura switch, caso o comprador queira o hatch, retorne: “Compra efetuada com sucesso”. Nas outras opções, retorne: “Tem certeza que não prefere este modelo?”. Caso seja especificado um modelo que não está disponível, retorne no console:

“Não trabalhamos com este tipo de automóvel aqui”.

1. Utilizando a estrutura do Switch faça um programa que simule uma calculadora básicaO programa recebe​ como parâmetros dois valores numéricos e uma string referente à operação e a realize com os valores numéricos na ordem que foram inseridos. Por exemplo: calculadora (2, ‘+’, 3). A função efetuará a soma de 2 e

3. Dica: Os sinais das operações são: ‘+’. ‘-’, ‘\*’ e ‘/’. Crie um caso default para operações inválidas.

1. Um funcionário irá receber um aumento de acordo com o seu plano de​ trabalho, de acordo com a tabela abaixo:

|  |  |
| --- | --- |
| Plano | Aumento |
| A | 10% |
| B | 15% |
| C | 20% |

Faça uma função que leia o plano de trabalho e o salário atual de um funcionário e calcula e imprime o seu novo salário. Use a estrutura switch e faça um caso default que indique que o plano é inválido.

1. Faça um programa que leia um número entre 0 e 10, e escreva este número por extenso. Use o comando​ switch. Crie um case default que escreva ‘Número fora do intervalo.’

**19)**​ O cardápio de uma lanchonete é o seguinte:​

|  |  |  |
| --- | --- | --- |
| Código | Descrição do Produto | Preço |
| 100 | Cachorro Quente | R$ 3,00 |
| 200 | Hambúrguer Simples | R$ 4,00 |
| 300 | Cheeseburguer | R$ 5,50 |
| 400 | Bauru | R$ 7,50 |
| 500 | Refrigerante | R$ 3,50 |
| 600 | Suco | R$ 2,80 |

Implemente uma função que receba como parâmetros o código do item pedido, a quantidade e calcule o valor a ser pago por aquele lanche. Considere que a cada execução somente será calculado um item. Use o comando switch. Crie um caso default para produto não existente.

1. Crie um programa para informar quais e quantas notas são necessárias para entregar o mínimo de cédulas​ para um determinado valor informado pelo usuário considerando notas de R$ 100, R$ 50, R$ 10 e R$ 5 e R$ 1. Seu programa deve mostrar apenas as notas utilizadas. Por exemplo, ao solicitar R$18, o programa deve informar apenas a seguinte informação (note que não foram exibidas informações sobre as demais cédulas): 1 nota(s) de R$ 10. 1 nota(s) de R$ 5. 3 nota(s) de R$ 1.

1. Criar um programa para identificar o valor a ser pago por um plano de saúde dada a idade do conveniado​ considerando que todos pagam R$ 100 mais um adicional conforme a seguinte tabela: 1) crianças com menos de 10 anos pagam R$80; 2) conveniados com idade entre 10 e 30 anos pagam R$50; 3) conveniados com idade acima de 30 e até 60 anos pagam R$ 95; e 4) conveniados acima de 60 anos pagam R$130

1. Criar uma função para calcular o valor a ser pago de anuidade de uma associação. A função recebe como​ parâmetro um inteiro que representa o mês (1 - janeiro, 2 - fevereiro…) que foi paga e o valor da anuidade. A anuidade deve ser paga no mês de janeiro. Por mês, é cobrado 5% de juros (sob o regime de juros compostos). O retorno deve ser o valor a ser pago para o respectivo mês escolhido.

1. Escreva um algoritmo que leia o código de um aluno e suas três notas. Calcule a média ponderada do​ aluno, considerando que o peso para a maior nota seja 4 e para as duas restantes, 3. Mostre o código do aluno, suas três notas, a média calculada e uma mensagem "APROVADO" se a média for maior ou igual a 5 e "REPROVADO" se a média for menor que 5. Repita a operação até que o código lido seja negativo.

1. Crie um programa que imprima 11 vezes a frase " Hello World!" utilizando uma estrutura de repetição while.​

1. Escrever um programa para exibir os números de 1 até 50 na tela.​

1. Fazer um programa para encontrar todos os pares entre 1 e 100.​

1. Construa uma função que receba como parâmetros as alturas e as taxas de crescimento anuais de duas​ crianças e calcule se existe uma criança menor, caso exista se a criança menor ultrapassará a maior e em quantos anos isso acontecerá. Utilize centímetros para as unidades de medida.

1. Ler um vetor de números inteiros e imprimir quantos são pares e quantos são ímpares.​

1. Utilizando a estrutura de repetição for, faça uma função que percorra um vetor e conte quantos números​ deste vetor estão no intervalo [10,20] (repare que o intervalo é fechado, ou seja, inclui o 10 e o 20) e quantos deles estão fora do intervalo, escrevendo estas informações.

1. Escreva um algoritmo que percorre um vetor de inteiros e defina o maior e menor valor dentro do vetor.​

1. Escrever um algoritmo que percorre um vetor de inteiros, conta quantos números negativos há nesse vetor​ e imprime a quantidade no console.

1. Construir um algoritmo que calcule a média aritmética dos valores de um vetor de inteiros.​

1. Crie três vetores, chamados vetorInteiro, vetorString e vetorDouble. Cada um destes vetores deverá conter​ quatro valores, sendo o primeiro com valores inteiros, o segundo com strings e o terceiro com valores decimais. Declarados os vetores, utilize a função de união concat() de duas maneiras diferentes para unir os vetores, e mostre o resultado no console. Todos os elementos do vetor resultado deverão aparecer no console.

1. Construa uma função que receberá duas Strings de tamanhos variados e que retornará True ou False caso​ todos os caracteres (independentemente de ser maiúsculo ou minúsculo) estejam contidos em ambas palavras. **35)** Crie dois vetores chamados vetorPilha e vetorAdiciona. Inicialmente, o vetorPilha conterá cinco elementos​ inteiros: [1, 2, 3, 4, 5]. Você deverá adicionar os valores contidos no vetorAdiciona [6, 7, 8, 9, 10] ao vetor pilha e mostrá-los no console. É importante lembrar que o método Push retorna somente o tamanho do Vetor. Ao final das operações imprima os vetores no console.

1. Crie duas funções que recebem dois parâmetros, um vetor com apenas valores numéricos e um número​ inteiro. Faça com que a primeira função retorne outro vetor que será resultado da multiplicação de cada elemento pelo número passado como parâmetro. A segunda função fará o mesmo da primeira se e somente se o valor do elemento for maior que 5.

1. Escreva duas funções, uma para progressão aritmética e uma para progressão geométrica que recebam​ como parâmetros um número n (número de termo), *a1*​ ​ (o primeiro termo) e *r*​​ (a razão) e escreva os *n*​ ​ termos , bem como a soma dos elementos.

1. Escreva uma função que receba dois parâmetros início e fim. Essa função deve imprimir todos os números​ ímpares que estão entre esses valores. Por padrão os valores devem ser 0 para início e 100 para fim. Atente para corrigir a ordem dos parâmetros caso a função receba o valor maior antes do menor.

1. Crie uma função que receba dois vetores de igual tamanho e troque seus elementos de modo que o​ primeiro elemento do vetorA passe a ser o primeiro elemento do vetorB e vice versa e assim sucessivamente. Faça a troca sem utilizar uma variável auxiliar.

1. Faça uma função que receba como parâmetro um vetor de notas e mostre os conceitos de cada uma de​ modo que de 0,0 a 4,9 seja atribuído o conceito D, de 5,0 a 6,9 seja atribuído o conceito C, de 7,0 a 8,9 o conceito B e de 9,0 a 10,0 o conceito A.