Atividade 03 - JS - Functions

Adriano Gomes

•

13:10 (Editado às 13:12)

Data de fim: 3/06

1. \*\*Função de Saudação\*\*:  
   - Crie uma função chamada `saudacao` que receba um nome como argumento e exiba uma mensagem de saudação no console.  
  
2. \*\*Função de Soma\*\*:  
   - Crie uma função chamada `soma` que receba dois números como argumentos e retorne a soma deles.  
  
3. \*\*Função de Subtração\*\*:  
   - Crie uma função chamada `subtracao` que receba dois números como argumentos e retorne a subtração do segundo número pelo primeiro.  
  
4. \*\*Função de Multiplicação\*\*:  
   - Crie uma função chamada `multiplicacao` que receba dois números como argumentos e retorne a multiplicação deles.  
  
5. \*\*Função de Divisão\*\*:  
   - Crie uma função chamada `divisao` que receba dois números como argumentos e retorne a divisão do primeiro pelo segundo.  
  
6. \*\*Função de Fatorial\*\*:  
   - Crie uma função chamada `fatorial` que receba um número como argumento e retorne o fatorial desse número.  
  
7. \*\*Função para Verificar Número Primo\*\*:  
   - Crie uma função chamada `ePrimo` que receba um número como argumento e retorne `true` se o número for primo e `false` caso contrário.  
  
8. \*\*Função para Calcular Média\*\*:  
   - Crie uma função chamada `media` que receba um array de números como argumento e retorne a média dos números.  
  
9. \*\*Função para Inverter String\*\*:  
   - Crie uma função chamada `inverterString` que receba uma string como argumento e retorne a string invertida.  
  
10. \*\*Função para Contar Caracteres\*\*:  
    - Crie uma função chamada `contarCaracteres` que receba uma string e um caractere como argumentos e retorne o número de vezes que o caractere aparece na string.  
  
11. \*\*Função Anônima\*\*:  
    - Crie uma função anônima e atribua-a a uma variável chamada `multiplicar`. A função deve receber dois números como argumentos e retornar a multiplicação deles.  
  
12. \*\*Função de Ordem Superior\*\*:  
    - Crie uma função chamada `operacao` que receba dois números e uma função como argumentos. A função deve aplicar a função fornecida aos dois números e retornar o resultado.  
  
13. \*\*Função de Retorno\*\*:  
    - Crie uma função chamada `criarSaudacao` que receba uma saudação como argumento e retorne uma nova função. A função retornada deve receber um nome como argumento e exibir a saudação seguida pelo nome.  
  
14. \*\*Função Recursiva\*\*:  
    - Crie uma função chamada `contagemRegressiva` que receba um número como argumento e exiba uma contagem regressiva a partir desse número até 0, usando recursão.  
  
15. \*\*Função para Calcular Fibonacci\*\*:  
    - Crie uma função chamada `fibonacci` que receba um número `n` como argumento e retorne o `n`-ésimo número da sequência de Fibonacci.  
  
16. \*\*Função para Converter Celsius para Fahrenheit\*\*:  
    - Crie uma função chamada `celsiusParaFahrenheit` que receba uma temperatura em Celsius e retorne a temperatura convertida para Fahrenheit.  
  
17. \*\*Função para Converter Fahrenheit para Celsius\*\*:  
    - Crie uma função chamada `fahrenheitParaCelsius` que receba uma temperatura em Fahrenheit e retorne a temperatura convertida para Celsius.  
  
18. \*\*Função para Verificar Paridade\*\*:  
    - Crie uma função chamada `ePar` que receba um número como argumento e retorne `true` se o número for par e `false` caso contrário.  
  
19. \*\*Função para Calcular Quadrado\*\*:  
    - Crie uma função chamada `quadrado` que receba um número como argumento e retorne o quadrado desse número.  
  
20. \*\*Função para Calcular Raiz Quadrada\*\*:  
    - Crie uma função chamada `raizQuadrada` que receba um número como argumento e retorne a raiz quadrada desse número.  
  
21. \*\*Função para Contar Vogais\*\*:  
    - Crie uma função chamada `contarVogais` que receba uma string como argumento e retorne o número de vogais na string.  
  
22. \*\*Função para Verificar Palíndromo\*\*:  
    - Crie uma função chamada `ePalindromo` que receba uma string como argumento e retorne `true` se a string for um palíndromo e `false` caso contrário.  
  
23. \*\*Função para Encontrar o Maior Número em um Array\*\*:  
    - Crie uma função chamada `maiorNumero` que receba um array de números como argumento e retorne o maior número do array.  
  
24. \*\*Função para Ordenar um Array em Ordem Crescente\*\*:  
    - Crie uma função chamada `ordenarArray` que receba um array de números como argumento e retorne o array ordenado em ordem crescente.  
  
25. \*\*Função para Filtrar Números Pares\*\*:  
    - Crie uma função chamada `filtrarPares` que receba um array de números como argumento e retorne um novo array contendo apenas os números pares.  
  
26. \*\*Função para Concatenar Arrays\*\*:  
    - Crie uma função chamada `concatenarArrays` que receba dois arrays como argumentos e retorne um novo array que seja a concatenação dos dois arrays.  
  
27. \*\*Função de Memoização para Fatorial\*\*:  
    - Crie uma função chamada `memoFatorial` que usa memoização para otimizar o cálculo do fatorial.  
  
28. \*\*Função para Verificar Substring\*\*:  
    - Crie uma função chamada `contemSubstring` que receba duas strings como argumentos e retorne `true` se a primeira string contiver a segunda string, e `false` caso contrário.  
  
29. \*\*Função de Curry\*\*:  
    - Crie uma função chamada `currySoma` que usa currying para somar três números. A função deve ser chamada assim: `currySoma(a)(b)(c)`.  
  
30. \*\*Função para Reduzir um Array\*\*:  
    - Crie uma função chamada `reduzirArray` que receba um array de números e uma função de redução (callback) como argumentos e retorne o valor reduzido.  
  
31. \*\*Função para Remover Duplicatas de um Array\*\*:  
    - Crie uma função chamada `removerDuplicatas` que receba um array como argumento e retorne um novo array sem duplicatas.  
  
32. \*\*Função de Composição\*\*:  
    - Crie uma função chamada `compor` que receba duas funções como argumentos e retorne uma nova função que é a composição das duas.  
  
33. \*\*Função para Mapear um Array\*\*:  
    - Crie uma função chamada `mapearArray` que receba um array e uma função de mapeamento (callback) como argumentos e retorne um novo array com os resultados da função de mapeamento aplicada a cada elemento.  
  
34. \*\*Função de Ordenação Personalizada\*\*:  
    - Crie uma função chamada `ordenarPersonalizado` que receba um array de objetos e uma função de comparação (callback) como argumentos e retorne o array ordenado.  
  
35. \*\*Função para Encontrar a Intersecção de Dois Arrays\*\*:  
    - Crie uma função chamada `intersecaoArrays` que receba dois arrays como argumentos e retorne um novo array contendo os elementos que são comuns aos dois arrays.