

Explicação de atividades envolvendo funções em JavaScript

Nesse documento, as atividades 1 à 16 serão explicadas, resolvidas e referenciadas, segundo a necessidade de pesquisa sobre temas diversos. Em ordem crescente, seguem aqui os exercícios explicados e resolvidos:

1.

```
<script>
  let num1 = 1
  let num2 = 2
  let num3 = 3
  let resultados = []
  resultados.push(dobrarNumero(num1))
  resultados.push(dobrarNumero(num2))
  resultados.push(dobrarNumero(num3))
  console.log(resultados)
</script>

//1
function dobrarNumero(num) {
  let dobro = num*2
  return dobro
}
```

Cada exercício possui um script, englobado em um arquivo html, mas também uma função, em um outro arquivo nomeado scripts.js, onde todas as funções estarão armazenadas.

Para o primeiro exercício, foi necessário apenas pré-definir 3 variáveis com números distintos, colocar todas elas na função, e imprimir os resultados no console. Na função, tudo o que ocorre é a multiplicação do número por 2, e depois seu retorno para o html.

2.

```
<script>
  let x = parseInt(prompt('Insira a base da potenciação: '))
  let z = parseInt(prompt('Insira a potência: '))
  let resultado = elevarNumero(x, z)
  console.log(resultado)
</script>

//2
function elevarNumero(num, potencia) {
  let calculo = num**potencia
  return calculo
}
```

O segundo exercício também veio com uma proposta simples, típico desses exercícios do Igor, que começam tranquilos e vão se dificultando ao decorrer do tempo. No html, duas linhas de código pedem ao usuário a base, e em seguida a potência de uma mesma potenciação. Os números são levados a função, que por sua vez só são potenciados, utilizando um conceito JS que faz uma potenciação com somente 2 símbolos (**). Por fim, os valores são retornados e impressos no console.

3.

```
<script>
  let num1 = 0
  let num2 = 0
  while(true){
    num1 = parseInt(prompt('Insira o primeiro número de 10 à 99: '))
    if ((num1 < 10) || (num1 > 99)){
      alert('O número deve estar entre 10 e 99. Comece novamente.')
    } else {
      num2 = parseInt(prompt('Insira o segundo número de 10 à 99: '))
      if ((num2 < 10) || (num2 > 99)){
        alert('O número deve estar entre 10 e 99. Comece novamente.')
      } else {
        break
      }
    }
  }
  resultado = dezenaDeUmUnidadeDeOutro(num1, num2)
  alert('Número novo: ' + resultado)
</script>

//3
function dezenaDeUmUnidadeDeOutro(n1, n2) {
  let dezena = parseInt(n1 / 10)
  let unidade = n2 % 10
  let resultado = dezena.toString() + unidade.toString()
  return resultado
}
```

No terceiro exercício, 2 números inteiros foram requisitados ao usuário com um prompt, esses somente entre 10 e 99, visto com os if e else adicionados ao script. A função divide o número inicial por 10, e transforma-o em número inteiro, ficando somente com um número único, e resultando na dezena. Depois, pega a sobra do segundo número dividido por 10, resultando na unidade. Como o número deve ser até 99, isso funciona perfeitamente. Por fim, os 2 números novos são transformados em string, concatenados, e retornados.

4.

```

<script>
  let count = 1
  let lista = []
  while(count != 4){
    var nota = parseFloat(prompt('Digite a '+count+'ª nota: '))
    lista.push(nota)
    count +=1
  }
  while(true){
    var decisao = prompt('Insira A para média Aritmética e P para Ponderada: ')
    if ((decisao == 'P') || (decisao == 'A')){
      break
    } else{
      alert('Favor digitar A ou P, em maiúsculo')
    }
  }
  if (decisao == 'P'){
    let resultadoP = mediaPonderada(lista[0], lista[1], lista[2])
    alert('Notas: '+lista+'\nMétodo de média escolhido: '+decisao+'\nMédia: '+resultadoP)
  } else if (decisao == 'A'){
    let resultadoA = mediaAritmetica(lista[0], lista[1], lista[2])
    alert('Notas: '+lista+'\nMétodo de média escolhido: '+decisao+'\nMédia: '+resultadoA)
  }
}
</script>

```

```

//4
function mediaAritmetica(n1, n2, n3) {
  let sum = n1+n2+n3
  let div = sum/3
  return div
}
function mediaPonderada(n1, n2, n3) {
  let mult1 = n1*5
  let mult2 = n2*3
  let mult3 = n3*2
  let sum = mult1 + mult2 + mult3
  let div = sum/10
  return div
}

```

Tudo o que foi requisitado nesse exercício foi que eu fizesse uma função que fizesse a média aritmética de 3 notas, e outra a média ponderada. Seguindo os conceitos matemáticos sobre os dois cálculos de média, as funções foram criadas. O script no arquivo html somente pede as 3 notas por meio de um loop while, e depois, outro loop para decidir se será calculada a média aritmética ou ponderada. Os resultados são exibidos conforme a escolha anterior.

```

<script>
  var resultado = 0
  let num1 = parseFloat(prompt('Inserir número: '))
  while(true){
    var simbolo = prompt('Inserir símbolo (+, -, * ou /): ')
    if ((simbolo == '+') || (simbolo == '-') || (simbolo == '*') || (simbolo == '/')){
      break
    } else{
      alert('Inserir símbolo válido.')
    }
  }
  let num2 = parseFloat(prompt('Inserir número: '))

  if (simbolo == '+'){
    resultado = adicao(num1, num2)
    alert(num1 + simbolo + num2 +'='+resultado)
  } else if (simbolo == '-'){
    resultado = subtracao(num1, num2)
    alert(num1 + simbolo + num2 +'='+resultado)
  } else if (simbolo == '*'){
    resultado = multiplicacao(num1, num2)
    alert(num1 + simbolo + num2 +'='+resultado)
  } else if (simbolo == '/'){
    resultado = divisao(num1, num2)
    alert(num1 + simbolo + num2 +'='+resultado)
  }
}
</script>

```

```

//5
function adicao(n1, n2) {
  let adic = n1 + n2
  return adic
}
function subtracao(n1, n2){
  let sub = n1 - n2
  return sub
}
function multiplicacao(n1, n2) {
  let mult = n1 * n2
  return mult
}
function divisao(n1, n2) {
  let div = n1/n2
  return div
}

```

No quinto exercício, o desafio foi criar uma função que realizasse operações matemáticas básicas entre dois números com base em um símbolo fornecido pelo usuário. O script HTML solicita ao usuário que insira o primeiro número e, em seguida, um símbolo representando a operação desejada (+, -, *, /). Um loop é utilizado para garantir que o símbolo inserido seja válido. Após isso, o segundo número é solicitado. As funções responsáveis pelas

operações (adição, subtração, multiplicação e divisão) são definidas em um arquivo separado, scripts.js. Dependendo do símbolo escolhido, a função correspondente é chamada e o resultado é exibido em um alerta. O código é estruturado de forma que, ao inserir números e um símbolo válidos, a operação correta seja executada e o resultado seja mostrado ao usuário.

6.

```
<script>
    let dias = parseInt(prompt('Quantos dias ainda restam no ano? '))
    tempoParaFerias(dias)
</script>

//6
function tempoParaFerias(n1) {
    if (n1 > 30){
        alert('Ainda falta muito para o natal.')
    } else if (n1 < 30){
        alert('Oba, quase férias!')
    } else {
        alert('Sei lá')
    }
}
```

No sexto exercício, o objetivo foi construir uma função que não retorna valor, mas que imprime mensagens baseadas na quantidade de dias restantes no ano. O script HTML solicita ao usuário que insira o número de dias restantes. Essa entrada é então passada para a função tempoParaFerias, que avalia o valor. Dentro da função, são utilizadas estruturas condicionais para verificar se a quantidade de dias é maior ou menor que 30. Se restarem mais de 30 dias, a função imprime a mensagem "Ainda falta muito para o natal." Caso contrário, se restarem menos de 30 dias, a mensagem exibida é "Oba, quase férias!" Além disso, uma condição adicional foi implementada para lidar com o caso em que exatamente 30 dias restam, que foi só um sei lá. A impressão das mensagens é responsabilidade exclusiva da função. Essa questão foi bem mais simples que a anterior, mas ok.

7.

```
<script>
    acabarExercicios()
</script>

//7
function acabarExercicios() {
    alert('Ufa! Este foi o ultimo exercicio! Acabei =D')
}
```

E finalmente chegamos ao último exercício. Tudo o que foi necessário aqui foi criar uma função que dissesse o que eu finalmente sinto, ufa, acabei! Utilizei também do fato de imprimir a função diretamente no arquivo html, sem retornar nada.

8.

```
<script>
    fuiTrolado()
</script>
```

```
//8
function fuiTrolado(){
    alert('Fui Trolado :(')
}
```

☹️. Não era o último exercício. O exercício atual é exatamente igual ao anterior, então não tem necessidade de explicar nada.

9.

```
<script>
    let num = parseInt(prompt('Insira um número inteiro: '))
    let resultado = contarDigitosEmNumero(num)
    alert('O número inserido possui '+resultado+' dígitos.')
</script>
```

```
//9
function contarDigitosEmNumero(n1) {
    if (isNaN(n1)) {
        return 0
    }
    let str = n1.toString()
    let tamanho = str.length
    return tamanho
}
```

No nono exercício, a tarefa foi desenvolver uma função que determina a quantidade de dígitos de um número inteiro informado pelo usuário. O script HTML solicita a entrada de um número inteiro por meio de um prompt. Essa entrada é convertida em um número inteiro e passada para a função contarDigitosEmNumero. Dentro da função, é realizada uma verificação para garantir que o valor fornecido não seja NaN (Not a Number). Se for, a função retorna 0. Caso contrário, o número é convertido em uma string e o comprimento dessa string é calculado, representando a quantidade de dígitos. O resultado é então retornado e exibido em um alerta no script principal, informando ao usuário quantos dígitos o número possui.

10.

```
<script>
    let num = parseInt(prompt('Insira um número inteiro: '))
    let resultado = inverterNumero(num)
    alert('O número invertido é '+resultado)
</script>
//10
function inverterNumero(num){
    let str = num.toString()
    let reverseStr = str.split('').reverse().join('');
    return reverseStr
}
```

No décimo exercício, a tarefa foi criar uma função que retorne o reverso de um número inteiro informado pelo usuário. O script HTML solicita ao usuário que insira um número inteiro através de um prompt. Esse número é convertido para um inteiro e passado para a função `inverterNumero`. Dentro da função, o número é convertido em uma string, permitindo que suas características sejam manipuladas. A string é então dividida em um array de caracteres usando o método `split`, invertida com o método `reverse` e, por fim, unida novamente em uma única string usando `join`. O resultado, que é o número invertido, é retornado e exibido em um alerta no script principal, mostrando ao usuário o número na sua forma reversa.

11.

```
<script>
    alert('Bem-vindo ao Jogo de Craps! Ao prosseguir, os 2 dados serão girados.')
    let dado = girarDoisDadosDe6Lados()
    if ((dado == 7) || (dado == 11)){
        alert('O resultado do dado foi '+dado+'. Você ganhou sendo um natural! Clique em F5 ou recarregue a página para jogar novamente!')
    } else if ((dado == 2) || (dado == 3) || (dado == 12)){
        alert('O resultado do dado foi '+dado+'. Você perdeu com um craps... Clique em F5 ou recarregue a página para jogar novamente!')
    } else {
        alert('O resultado do dado foi '+dado+'. Esse é seu ponto, e seu objetivo é tirar esse número no dado, antes de tirar 7. Está pronto?')
        var ponto = dado
        while(true){
            dado = girarDoisDadosDe6Lados()
            alert('O resultado do dado foi '+dado+'.')
            if (ponto == dado){
                alert('Parabéns, você ganhou! Clique em F5 ou recarregue a página para jogar novamente!')
                break
            }
            if (dado == 7){
                alert('Infelizmente, você tirou 7 antes de tirar '+ponto+'... Clique em F5 ou recarregue a página para jogar novamente!')
                break
            }
        }
    }
</script>
```

```
//11
function girarDoisDadosDe6Lados() {
    let dado = Math.floor(Math.random() * (12 - 2 + 1)) + 2
    return dado
}
```

No décimo primeiro exercício, o objetivo foi implementar um jogo de Craps, onde o jogador lança um par de dados e obtém um resultado entre 2 e 12. O script HTML começa com uma mensagem de boas-vindas e, em seguida, chama a função `girarDoisDadosDe6Lados` para simular o lançamento dos dados. O resultado da primeira jogada é avaliado: se for 7 ou 11, o jogador ganha imediatamente, sendo considerado um "natural". Se o resultado for 2, 3 ou 12, o jogador perde, conhecido como "craps". Para valores entre 4 e 10, esse número é estabelecido como o "Ponto", e o jogador deve continuar lançando os dados até obter esse número novamente antes de tirar um 7. Caso o jogador tenha um ponto, um loop é iniciado, onde o dado é lançado repetidamente. As mensagens informam o resultado de cada jogada. Se o jogador conseguir tirar o Ponto, uma mensagem de vitória é exibida. Se um 7 for tirado antes do Ponto, uma mensagem de derrota é apresentada. O jogo termina após uma vitória ou derrota, e o usuário é incentivado a recarregar a página para jogar novamente.

12.

```
<script>
  let data = prompt('Insira uma data no seguinte formato: DD/MM/AAAA ');
  let lista = transformarModeloDeDataMesAno(data)
  if (lista === null){
    alert('NULL')
  } else {
    alert(lista[0]+' de '+lista[1]+' de '+lista[2])
  }
</script>
```

```
//12
function transformarModeloDeDataMesAno(data) {
  let str = data.toString();
  let partes = str.split('/');
  if (partes.length !== 3) {
    return null;
  } else {
    let dia = parseInt(partes[0], 10)
    let mes = parseInt(partes[1], 10)
    let ano = parseInt(partes[2], 10)
    if ((ano < 0) || (mes < 1) || (mes > 12) || (dia < 1) || (dia > 31)) {
      return null
    }
    if (mes === 2 && dia == 29 && (ano % 4 !== 0 || (ano % 100 == 0 && ano % 400 !== 0))) {
      return null
    }
    if ((mes == 4 || mes == 6 || mes == 9 || mes == 11) && dia == 31) {
      return null
    }
    if (mes == 2 && dia > 29) {
      return null
    }
    let meses = ['janeiro', 'fevereiro', 'março', 'abril', 'maio', 'junho', 'julho', 'agosto', 'setembro', 'outubro', 'novembro', 'dezembro']
    let lista = []
    lista.push(dia)
    lista.push(meses[mes - 1])
    lista.push(ano)
    return lista
  }
}
```

No décimo segundo exercício, a proposta foi construir uma função que recebe uma data no formato DD/MM/AAAA e a converte para uma string no formato "D de mêsPorExtenso de AAAA". O script HTML solicita ao usuário que insira uma data, que é então passada para a função `transformarModeloDeDataMesAno`. Dentro da função, a string da data é dividida em partes usando o método `split`, que separa o dia, o mês e o ano. A função valida a estrutura da data, retornando null caso não siga o formato adequado ou se os valores não forem válidos (por exemplo, mês fora do intervalo 1-12 ou dia fora do intervalo adequado para o mês). A validação inclui checagens específicas, como se o mês de fevereiro possui 29 dias em anos

bissextos e se os meses de abril, junho, setembro e novembro têm apenas 30 dias. Caso a data seja válida, um array é criado para armazenar o dia, o nome do mês por extenso (utilizando um array de strings com os meses) e o ano. Por fim, o array é retornado. No script principal, caso a função retorne null, um alerta é exibido informando que a data é inválida. Se a data for válida, uma mensagem formatada é exibida, apresentando a data na nova formatação.

13.

```
<script>
  let palavra = prompt('Insira uma palavra para embaralhá-la: ')
  let resultado = embaralhaPalavra(palavra)
  alert(resultado)
</script>
//13
function embaralhaPalavra(str) {
  let letras = str.split('')
  let lista = []
  let tamanho = letras.length
  let indicesUsados = []
  while (lista.length < tamanho) {
    let randomIndex = Math.floor(Math.random() * tamanho)
    if (!indicesUsados.includes(randomIndex)) {
      indicesUsados.push(randomIndex)
      lista.push(letras[randomIndex])
    }
  }
  return lista.join('')
}
```

No décimo terceiro exercício, a tarefa foi criar uma função que embaralha os caracteres de uma string recebida como parâmetro. O script HTML solicita ao usuário que insira uma palavra, que é então passada para a função `embaralhaPalavra`. Dentro da função, a string é dividida em um array de letras utilizando o método `split`. Um loop é utilizado para criar uma nova lista que irá conter os caracteres embaralhados. A cada iteração, um índice aleatório é gerado, e se esse índice ainda não foi utilizado, a letra correspondente é adicionada à lista de resultados. Para garantir que não haja repetições, um array de índices já usados é mantido. Depois que todos os caracteres foram coletados na nova ordem, o array é convertido novamente em uma string utilizando o método `join`. A função retorna essa nova string, que representa a palavra embaralhada. No script principal, o resultado é exibido em um alerta, mostrando ao usuário a palavra com os caracteres em ordem aleatória.

14.

```
<script>
  let n1 = parseFloat(prompt('Insira o valor de a: '))
  let n2 = parseFloat(prompt('Insira o valor de b: '))
  let n3 = parseFloat(prompt('Insira o valor de c: '))
  let resultado = bhaskara(n1, n2, n3)
  alert(resultado)
</script>
```

```
//14
function bhaskara(a, b, c) {
  if (a == 0) {
    return "O valor de 'a' não pode ser zero.";
  }
  let discriminante = b ** 2 - 4 * a * c;
  if (discriminante < 0) {
    return "Não existem raízes reais.";
  }
  let raiz1 = (-b + Math.sqrt(discriminante)) / (2 * a)
  let raiz2 = (-b - Math.sqrt(discriminante)) / (2 * a)
  return [raiz1, raiz2]
}
```

No décimo quarto exercício, tudo o que ocorre é uma pura representação da fórmula de Bhaskara. O script requisita os 3 números necessários, e a função calcula isso, com base nos conceitos matemáticos formulados por Bhaskara.

15.

```
<script>
  let c1 = parseFloat(prompt('Insira o valor do cateto 1: '))
  let c2 = parseFloat(prompt('Insira o valor do cateto 2: '))
  let resultado = pitagoras(c1, c2)
  alert('Hipotenusa = '+resultado)
</script>
```

```
//15
function pitagoras(cateto1, cateto2) {
  cateto1 *= cateto1
  cateto2 *= cateto2
  let soma = cateto1 + cateto2
  result = Math.sqrt(soma)
  return result
}
```

Da mesma forma que no exercício anterior, tudo o que fora feito aqui foi um cálculo matemático sobre Pitágoras. Catetos foram requisitados ao usuário, e multiplicados e raizados da forma correta.

16.

```
<script>
  alegriaEmpolgacao()
</script>
```

```
//16
function alegriaEmpolgacao() {
|   alert('Acabou.')
}
```

Acabou.

Referências

- MDN Web Docs. **JavaScript**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 27 set. 2024.
- W3Schools. **JavaScript**. Disponível em: <https://www.w3schools.com/js/>. Acesso em: 27 set. 2024.
- JavaScript.info. **JavaScript**. Disponível em: <https://javascript.info/>. Acesso em: 27 set. 2024.
- FreeCodeCamp. **JavaScript**. Disponível em: <https://www.freecodecamp.org/news/tag/javascript/>. Acesso em: 27 set. 2024.
- Stack Overflow. **Stack Overflow**. Disponível em: <https://stackoverflow.com/>. Acesso em: 27 set. 2024.
- CALCULADORA ONLINE. **Fórmula de Bhaskara**. Disponível em: <https://www.calculadora-online.com.br/funcoes/bhaskara>. Acesso em: 27 set. 2024.
- WIKIPEDIA. **Teorema de Pitágoras**. Disponível em: https://pt.wikipedia.org/wiki/Teorema_de_Pit%C3%A1goras. Acesso em: 27 set. 2024.
- MDN Web Docs. **Array.prototype.join()**. Disponível em: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/join. Acesso em: 27 set. 2024.
- ALURA. **JavaScript: Aprenda a Programar do Zero**. Disponível em: <https://www.alura.com.br/artigos/javascript-aprenda-a-programar-do-zero>. Acesso em: 27 set. 2024.
- COTIL. **Validação de Formulários com JavaScript**. Disponível em: <https://www.cotil.unicamp.br/validacao-formularios-javascript>. Acesso em: 27 set. 2024.
- BOLLY. **O que é a função Math.random() no JavaScript?**. Disponível em: <https://bolly.com.br/o-que-e-a-funcao-math-random-no-javascript/>. Acesso em: 27 set. 2024.

