



Curso de JavaScript para Iniciantes: Fundamentos e Prática

EXERCÍCIOS
MÓDULO 2

LEANDRO OLIVEIRA
LECONNECT
<https://leconnect.online/>

Exercícios do Módulo 2 - Funções

Conteúdo base

1. Introdução às funções em JavaScript
2. Parâmetros e argumentos de função
3. Escopo de variáveis
4. Funções anônimas e callbacks

Preparado para começar?

Nesta lista, você encontrará 20 exercícios com diversos tipos de abordagem. São elas: verdadeiro ou falso, única escolha, múltipla escolha e abertos (resposta livre). Ao final da lista, você encontrará as respostas.

Exercício 1:

Assunto: Introdução às funções em JavaScript

Tipo: Verdadeiro ou Falso

Pergunta: Uma função em JavaScript é um bloco de código que pode ser chamado pelo nome e reutilizado em diferentes partes do programa.

Exercício 2:

Assunto: Introdução às funções em JavaScript

Tipo: Única escolha

Pergunta: Qual é a palavra-chave utilizada para declarar uma função em JavaScript?

- a) function
- b) funcao
- c) def
- d) var

Exercício 3:

Assunto: Introdução às funções em JavaScript

Tipo: Preencha as lacunas

Pergunta: Complete a declaração de uma função em JavaScript para calcular a área de um retângulo:

```
function calcularArea(base, altura) {  
    return ____ * ____;  
}
```

Exercício 4:

Assunto: Introdução às funções em JavaScript

Tipo: Múltiplas escolhas

Pergunta: Uma função pode retornar apenas um valor em JavaScript.

- a) Verdadeiro
- b) Falso

Exercício 5:

Assunto: Parâmetros e argumentos de função

Tipo: Resposta livre

Pergunta: Explique a diferença entre parâmetros e argumentos de uma função em JavaScript.

Exercício 6:

Assunto: Parâmetros e argumentos de função

Tipo: Verdadeiro ou Falso

Pergunta: Os parâmetros de uma função são especificados na chamada da função, enquanto os argumentos são definidos na declaração da função.

Exercício 7:

Assunto: Parâmetros e argumentos de função

Tipo: Única escolha

Pergunta: Qual é a forma correta de chamar a função "saudacao" com o argumento "nome" em JavaScript?

- a) saudacao(nome);
- b) saudacao = nome;
- c) saudacao(nome = "Alice");
- d) saudacao:nome;

Exercício 8:

Assunto: Parâmetros e argumentos de função

Tipo: Preencha as lacunas

Pergunta: Complete a chamada da função para calcular o valor total de uma compra, considerando o valor unitário do produto e a quantidade comprada:

```
const valorTotal = calcularValorTotal(_____, _____);
```

Exercício 9:

Assunto: Parâmetros e argumentos de função

Tipo: Múltiplas escolhas

Pergunta: Em JavaScript, uma função pode ter quantos parâmetros forem necessários.

- a) Apenas 1 parâmetro
- b) No máximo 3 parâmetros
- c) Apenas 2 parâmetros
- d) Qualquer quantidade de parâmetros

Exercício 10:

Assunto: Escopo de variáveis

Tipo: Resposta livre

Pergunta: Explique o que é o escopo de uma variável em JavaScript e como o escopo global e o escopo local se diferenciam.

Exercício 11:

Assunto: Escopo de variáveis

Tipo: Verdadeiro ou Falso

Pergunta: Variáveis declaradas dentro de uma função têm escopo global em JavaScript.

- a) Verdadeiro
- b) Falso

Exercício 12:

Assunto: Escopo de variáveis

Tipo: Única escolha

Pergunta: Qual é o valor de "resultado" após a execução do código a seguir?

```
let resultado = 0;

function somar(numero) {
  resultado = resultado + numero;
}

somar(5);
somar(3);
```

- a) 5
- b) 3
- c) 8
- d) 0

Exercício 13:

Assunto: Escopo de variáveis

Tipo: Preencha as lacunas

Pergunta: Complete a função abaixo para calcular a área de um círculo (use 3.14 como valor de pi):

```
function calcularAreaCirculo(raio) {  
  const area = _____ * raio * raio;  
  return area;  
}
```

Exercício 14:

Assunto: Funções anônimas e callbacks

Tipo: Verdadeiro ou Falso

Pergunta: Uma função anônima é uma função sem nome e pode ser usada como um valor em JavaScript.

- a) Verdadeiro
- b) Falso

Exercício 15:

Assunto: Funções anônimas e callbacks

Tipo: Única escolha

Pergunta: Qual é o objetivo principal do uso de callbacks em JavaScript?

- a) Reduzir a complexidade do código
- b) Otimizar o desempenho do programa
- c) Controlar o fluxo de execução assíncrona
- d) Evitar o uso de funções anônimas

Exercício 16:

Assunto: Funções anônimas e callbacks

Tipo: Múltiplas escolhas

Pergunta: As funções de callback em JavaScript sempre precisam ser anônimas.

- a) Verdadeiro
- b) Falso

Exercício 17:

Assunto: Funções anônimas e callbacks

Tipo: Preencha as lacunas

Pergunta: Complete a chamada da função "setTimeout" para exibir a mensagem "Olá!" após 2 segundos:

```
setTimeout(_____, 2000);
```

Exercício 18:

Assunto: Funções anônimas e callbacks

Tipo: Resposta livre

Pergunta: Explique o conceito de "hoisting" em relação a funções em JavaScript.

Exercício 19:

Assunto: Introdução às funções em JavaScript

Tipo: Resposta livre

Pergunta: Crie uma função em JavaScript chamada "dobrarNumero" que recebe um número como parâmetro e retorna o dobro desse número.

Exercício 20:

Assunto: Escopo de variáveis

Tipo: Verdadeiro ou Falso

Pergunta: Variáveis declaradas com a palavra-chave "const" em JavaScript não podem ter seu valor reatribuído.

- a) Verdadeiro
- b) Falso

Respostas

Exercício 1:

Resposta: Verdadeiro

Exercício 2:

Resposta: a) function

Exercício 3:

Resposta:

```
function calcularArea(base, altura) {  
  return base * altura;  
}
```

Exercício 4:

Resposta: Falso

Exercício 5:

Resposta: Parâmetros são variáveis declaradas na definição da função, enquanto argumentos são valores passados para a função quando ela é chamada.

Exercício 6:

Resposta: Falso. Os parâmetros são definidos na declaração da função, e os argumentos são passados na chamada da função.

Exercício 7:

Resposta: a) saudacao(nome);

Exercício 8:

Resposta:

```
const valorTotal = calcularValorTotal(valorUnitario, quantidade);
```


Exercício 9:

Resposta: d) Qualquer quantidade de parâmetros

Exercício 10:

Resposta: O escopo de uma variável em JavaScript refere-se à área do código onde a variável é acessível. O escopo global torna a variável acessível em todo o código, enquanto o escopo local torna a variável acessível apenas dentro do bloco onde foi declarada.

Exercício 11:

Resposta: Falso. Variáveis declaradas dentro de uma função têm escopo local.

Exercício 12:

Resposta: c) 8

Exercício 13:

Resposta:

```
function calcularAreaCirculo(raio) {  
  const area = 3.14 * raio * raio;  
  return area;  
}
```

Exercício 14:

Resposta: Verdadeiro

Exercício 15:

Resposta: c) Controlar o fluxo de execução assíncrona

Exercício 16:

Resposta: b) Falso

Exercício 17:

Resposta:

```
setTimeout(function() {  
  console.log("Olá!");  
}, 2000);
```

Exercício 18:

Resposta: Hoisting é um comportamento em JavaScript em que declarações de variáveis e funções são movidas para o topo do escopo em tempo de execução.

Exercício 19:

Resposta:

```
function dobrarNumero(numero) {  
  return numero * 2;  
}
```

Exercício 20:

Resposta: Verdadeiro

Considerações finais

Para cada módulo do curso, serão realizados exercícios com o objetivo de consolidar o conteúdo aprendido. Dessa forma, disponibilizarei um PDF semelhante a este, para que você possa praticar.

É importante lembrar que os exercícios serão constantemente atualizados e novos serão adicionados. Portanto, sempre verifique os módulos que você já concluiu para encontrar novos exercícios.

Agradeço imensamente por dedicar seu tempo para realizar essas atividades no [Curso de JavaScript para Iniciantes: Fundamentos e Prática](#).

Recomendo que, para avançar para o próximo módulo, você responda aos exercícios propostos. Quando se sentir confortável, sinta-se à vontade para prosseguir com os próximos módulos do curso.

Não deixe de compartilhar com seus amigos e em suas redes sociais, isso nos ajudará a melhorar e criar novos cursos.

Bons estudos!

Próximo módulo

