



Curso de JavaScript para Iniciantes: Fundamentos e Prática

EXERCÍCIOS

MÓDULO 1

LEANDRO OLIVEIRA

LECONNECT

<https://leconnect.online/>

Exercícios do Módulo 1 - Estruturas de Controle

Conteúdo base

1. Estruturas condicionais: if, else if, else
2. Estruturas de repetição: while, do-while, for
3. O uso de operadores lógicos

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: Estruturas condicionais: if, else if, else

Tipo: Verdadeiro ou Falso

Pergunta: A estrutura "if" permite executar um bloco de código somente se uma condição especificada for verdadeira.

Exercício 2:

Assunto: Estruturas condicionais: if, else if, else

Tipo: Única escolha

Pergunta: Qual é a sintaxe correta para uma estrutura "else if" em JavaScript?

- a) else-if
- b) elseif
- c) else if
- d) elif

Exercício 3:

Assunto: Estruturas condicionais: if, else if, else

Tipo: Preencha as lacunas

Pergunta: Complete a estrutura condicional abaixo para verificar se um número é positivo, negativo ou zero:

```
if (_____ > 0) {  
    console.log("O número é positivo");  
} else if (_____ < 0) {  
    console.log("O número é negativo");  
} else {  
    console.log("O número é zero");  
}
```

Exercício 4:

Assunto: Estruturas condicionais: if, else if, else

Tipo: Múltiplas escolhas

Pergunta: Qual das seguintes opções é uma estrutura condicional aninhada em JavaScript?

- a) if else
- b) else if
- c) if else if
- d) if if

Exercício 5:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Resposta livre

Pergunta: Explique a diferença entre as estruturas "while" e "do-while" em JavaScript, e forneça um exemplo de uso para cada uma delas.

Exercício 6:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Verdadeiro ou Falso

Pergunta: A estrutura "for" é mais adequada quando o número de iterações é desconhecido ou depende de uma condição.

Exercício 7:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Única escolha

Pergunta: Qual é a sintaxe correta para um loop "while" em JavaScript?

- a) while (condição) { ... }
- b) (condição) while { ... }
- c) loop while (condição) { ... }
- d) while { ... } (condição)

Exercício 8:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Preencha as lacunas

Pergunta: Complete o loop "for" abaixo para exibir os números de 1 a 5 no console:

```
for (let i = __; i <= __; __) {  
  console.log(i);  
}
```

Exercício 9:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Múltiplas escolhas

Pergunta: Qual das seguintes opções é uma estrutura de repetição que garante que o bloco de código seja executado pelo menos uma vez?

- a) while
- b) for
- c) do-while
- d) repeat-while

Exercício 10:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Resposta livre

Pergunta: Crie um código JavaScript usando a estrutura de repetição adequada para exibir a soma dos números de 1 a 10 no console.

Exercício 11:

Assunto: O uso de operadores lógicos

Tipo: Verdadeiro ou Falso

Pergunta: O operador lógico "&&" retorna verdadeiro apenas se ambos os operandos forem verdadeiros.

Exercício 12:

Assunto: O uso de operadores lógicos

Tipo: Única escolha

Pergunta: Qual é o valor resultante da expressão: true && false ?

- a) true
- b) false
- c) undefined
- d) null

Exercício 13:

Assunto: O uso de operadores lógicos

Tipo: Múltiplas escolhas

Pergunta: Qual é o operador lógico usado para inverter o valor de uma expressão booleana?

- a) ||
- b) !
- c) &&
- d) ==!

Exercício 14:

Assunto: O uso de operadores lógicos

Tipo: Preencha as lacunas

Pergunta: Complete a seguinte expressão lógica em JavaScript para verificar se um número está no intervalo entre 10 e 20 (inclusive):

```
if ( _____ > 10 && _____ <= 20) {  
    console.log("O número está no intervalo de 10 a 20");  
} else {  
    console.log("O número está fora do intervalo");  
}
```

Exercício 15:

Assunto: O uso de operadores lógicos

Tipo: Resposta livre

Pergunta: Explique o conceito de curto-circuito em operadores lógicos e como ele pode ser útil em condições complexas.

Exercício 16:

Assunto: O uso de operadores lógicos

Tipo: Verdadeiro ou Falso

Pergunta: O operador lógico "||" retorna verdadeiro apenas se ambos os operandos forem falsos.

Exercício 17:

Assunto: O uso de operadores lógicos

Tipo: Única escolha

Pergunta: Qual é o valor resultante da expressão: true || false ?

- a) true
- b) false
- c) undefined
- d) null

Exercício 18:

Assunto: Estruturas de repetição: while, do-while, for

Tipo: Preencha as lacunas

Pergunta: Complete o loop "do-while" abaixo para exibir os números de 1 a 5 no console:

```
let i = 1;

do {
  console.log(i);
  ____;
} while (i <= ____);
```

Exercício 19:

Assunto: Estruturas condicionais: if, else if, else

Tipo: Resposta livre

Pergunta: Crie um código JavaScript que verifique se um número é par ou ímpar e exiba uma mensagem no console com o resultado.

Exercício 20:

Assunto: Estruturas condicionais: if, else if, else

Tipo: Verdadeiro ou Falso

Pergunta: A estrutura "else" pode ser usada sozinha, sem a presença de um "if" correspondente.

Respostas

Exercício 1:

Resposta: Verdadeiro

Exercício 2:

Resposta: c) else if

Exercício 3:

Resposta:

```
if (numero > 0) {  
    console.log("O número é positivo");  
} else if (numero < 0) {  
    console.log("O número é negativo");  
} else {  
    console.log("O número é zero");  
}
```

Exercício 4:

Resposta: c) if else if

Exercício 5:

Resposta: A estrutura "while" executa um bloco de código enquanto a condição é verdadeira, enquanto a estrutura "do-while" executa o bloco uma vez e, em seguida, verifica a condição.

Exercício 6:

Resposta: Falso. A estrutura "for" é mais adequada quando o número de iterações é conhecido.

Exercício 7:

Resposta: a) while (condição) { ... }

Exercício 8:

Resposta:

```
for (let i = 1; i <= 5; i++) {  
  console.log(i);  
}
```

Exercício 9:

Resposta: c) do-while

Exercício 10:

Resposta:

```
let soma = 0;  
  
for (let i = 1; i <= 10; i++) {  
  soma += i;  
}  
  
console.log(soma);
```

Exercício 11:

Resposta: Verdadeiro

Exercício 12:

Resposta: b) false

Exercício 13:

Resposta: b) !

Exercício 14:

Resposta:

```
if (numero > 10 && numero <= 20) {  
    console.log("O número está no intervalo de 10 a 20");  
} else {  
    console.log("O número está fora do intervalo");  
}
```

Exercício 15:

Resposta: Curto-circuito é um conceito em que, quando um operador lógico encontra o resultado da expressão mesmo antes de avaliar todos os seus operandos, ele interrompe a avaliação e retorna o resultado atual.

Exercício 16:

Resposta: Falso. O operador "||" retorna verdadeiro se pelo menos um dos operandos for verdadeiro.

Exercício 17:

Resposta: a) true

Exercício 18:

Resposta:

```
let i = 1;  
  
do {  
    console.log(i);  
    i++;  
} while (i <= 5);
```

Exercício 19:

Resposta:

```
const numero = 10;
if (numero % 2 === 0) {
  console.log("O número é par");
} else {
  console.log("O número é ímpar");
}
```

Exercício 20:

Resposta: Falso. A estrutura "else" sempre precisa estar associada a um "if" correspondente.

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

