

JavaScript Impressionador - Módulo 5 - Estruturas Condicionais e de Repetição

Exercícios

Estrutura de Repetição - FOR:

Exercício 1: Cálculo do Faturamento Semanal

Você é o gerente de uma lanchonete e deseja calcular o valor total das vendas de hambúrgueres em uma semana. Cada hambúrguer custa R\$10, e você registra o número de hambúrgueres vendidos a cada dia da semana. Crie um programa que calcule o valor total das vendas em uma semana.

```
function calcularFaturamentoSemanal(vendasPorDia, precoPorHamburger) {
    let faturamentoSemanal = 0;
    for (let i = 0; i < vendasPorDia.length; i++) {
        faturamentoSemanal += vendasPorDia[i] * precoPorHamburger;
    }
    return faturamentoSemanal;
}

const vendasPorDia = [20, 15, 25, 18, 22, 30, 28];
const precoPorHamburger = 10;
const faturamentoTotal = calcularFaturamentoSemanal(
    vendasPorDia,
    precoPorHamburger
);
console.log(`O faturamento total da semana é R${faturamentoTotal}.`);
```

Exercício 2: Listagem do Cardápio Digital

Você deseja criar um cardápio digital para sua lanchonete. Crie um programa que liste os itens do cardápio juntamente com seus preços. Utilize um loop for para percorrer o cardápio e exibi-lo.

```
const cardapio = [
    { nome: "Hambúrguer", preco: 10 },
    { nome: "Batata Frita", preco: 5 },
    { nome: "Refrigerante", preco: 3 },
    { nome: "Milkshake", preco: 7 },
];

function exibirCardapio(cardapio) {
    console.log("Cardápio da Lanchonete:");
    for (let i = 0; i < cardapio.length; i++) {
        console.log(`${cardapio[i].nome}: R${cardapio[i].preco}`);
    }
}
```

```
exibirCardapio(cardapio);
```

Estrutura de Repetição - WHILE:

Exercício 3: Contagem de Dinheiro no Caixa

Você é o caixa de uma lanchonete e precisa contar o dinheiro recebido até que a última venda do dia seja feita. Crie um programa que simule a contagem do dinheiro recebido a cada venda até o fechamento da lanchonete.

```
function realizarVendas(totalVendasDesejado) {  
  let dinheiroRecebido = 0;  
  let vendaConcluida = false;  
  let vendasRealizadas = 0;  
  
  while (!vendaConcluida) {  
    const valorDaVenda = 20; // Valor da venda (exemplo)  
  
    // Simulando uma venda  
    dinheiroRecebido += valorDaVenda;  
    vendasRealizadas++;  
  
    // Verificando se atingiu o número desejado de vendas  
    if (vendasRealizadas >= totalVendasDesejado) {  
      vendaConcluida = true;  
    }  
  }  
  
  console.log(`Total de dinheiro recebido: R${dinheiroRecebido.toFixed(2)}`);  
}  
  
const totalVendasDesejado = 5; // Defina o número desejado de vendas  
realizarVendas(totalVendasDesejado);
```

Exercício 4: Entrega de Pedidos Pendentes

Você é o entregador de uma lanchonete e precisa entregar pedidos até que não haja mais pedidos pendentes. Crie um programa que simule a entrega de pedidos até que não haja mais pedidos na lista.

```
const pedidosPendentes = [  
  "Hambúrguer",  
  "Batata Frita",  
  "Refrigerante",  
  "Milkshake",  
];  
  
function realizarEntregas(pedidosPendentes) {
```

```
let pedidoAtual = "";

console.log("Iniciando entregas:");
while (pedidosPendentes.length > 0) {
    pedidoAtual = pedidosPendentes.shift(); // Remove o primeiro elemento de
    um array e retorna o elemento que foi removido.
    console.log(`Entregando: ${pedidoAtual}`);
}

console.log("Todos os pedidos foram entregues.");
}

realizarEntregas(pedidosPendentes);
```

Estrutura de Repetição - DO...WHILE:

Exercício 5: Adivinhe o Número Secreto para Desconto

Você deseja criar um programa que permite aos clientes da sua lanchonete adivinharem um número secreto para ganhar um desconto. Eles podem tentar quantas vezes quiserem, mas só ganharão o desconto quando adivinharem o número secreto. Crie um programa que use a estrutura do...while para permitir que os clientes tentem adivinhar o número secreto.

```
function adivinharNumeroSecreto() {
    const numeroSecreto = Math.floor(Math.random() * 100); // Número secreto entre
    0 e 99
    let tentativa;
    let tentativas = 0;

    do {
        tentativa = Math.floor(Math.random() * 100); // Gere uma nova tentativa
        aleatória
        tentativas++;

        if (tentativa === numeroSecreto) {
            console.log(
                `Parabéns! Você adivinhou o número secreto (${numeroSecreto}) em
                ${tentativas} tentativas.`
            );
        }
    } while (tentativa !== numeroSecreto);
}

adivinharNumeroSecreto();
```

Exercício 6: Gerenciamento de Estoque de Itens em Falta

Você é o gerente de uma lanchonete e está verificando o estoque de itens em falta. Crie um programa que permita que você insira os itens que estão em falta e os adicione ao estoque até que você decida encerrar o

processo.

```
const estoqueLanchonete = {
  hamburguer: 20,
  batataFrita: 15,
  refrigerante: 30,
  milkshake: 10,
};

function atualizarEstoque(estoque, itemEmFalta, quantidadeAdicionar) {
  if (estoque.hasOwnProperty(itemEmFalta)) {
    estoque[itemEmFalta] += quantidadeAdicionar;
    console.log(`Estoque atualizado: ${itemEmFalta}:
    ${estoque[itemEmFalta]}`);
  } else {
    console.log("Item não encontrado no estoque.");
  }
}

console.log("Estoque da Lanchonete:");
console.log(estoqueLanchonete);

let continuarAdicionando = true;
let itemNaoEncontrado = false; // Variável para rastrear se o item não foi
encontrado

do {
  const itemEmFalta = "hamburguer"; // Substitua pelo item em falta (ou obtenha
do usuário)
  const quantidadeAdicionar = 5; // Quantidade a ser adicionada

  if (!estoqueLanchonete.hasOwnProperty(itemEmFalta)) {
    // Se o item não existe no estoque, exiba a mensagem apenas uma vez
    if (!itemNaoEncontrado) {
      console.log("Item não encontrado no estoque.");
      itemNaoEncontrado = true;
    }
  } else if (estoqueLanchonete[itemEmFalta] + quantidadeAdicionar > 50) {
    continuarAdicionando = false;
    console.log(
      `Limite de estoque (${itemEmFalta}: ${estoqueLanchonete[itemEmFalta]})
atingido.`
    );
  } else {
    atualizarEstoque(estoqueLanchonete, itemEmFalta, quantidadeAdicionar);
  }
} while (continuarAdicionando);

console.log("Estoque final:");
console.log(estoqueLanchonete);
```