

## Módulo 15

### Exercícios

Lembre-se de que os exercícios podem ser resolvidos com diferentes abordagens, e nosso gabarito é apenas uma forma de implementação.

**Exercício 1:** Você é responsável por atualizar os salários de todos os funcionários de uma empresa. A empresa decidiu aumentar o salário de todos em 10%. Crie um array com os salários atuais e utilize o método `.map()` para calcular o novo salário de cada funcionário. Qual será o array resultante?

**Exemplo de Entrada:** // [3000, 4500, 5000, 6000, 7500];

**Exemplo de Saída:** // [3300, 4950, 5500, 6600, 8250];

**Exercício 2:** Uma loja decidiu reajustar os preços de todos os produtos em 5% para cobrir os custos operacionais. Crie um array com os preços dos produtos e utilize o método `.map()` para calcular os novos preços. Quais serão os novos preços?

**Exemplo de Entrada:** // [50, 100, 150, 200, 250];

**Exemplo de Saída:** //// [52.5, 105, 157.5, 210, 262.5]

**Exercício 3:** Você precisa adicionar um identificador único (ID) a cada funcionário da empresa. Crie um array com os nomes dos funcionários e utilize o método `.map()` para adicionar um ID a cada um deles. O ID deve ser baseado na posição do funcionário no array. Como ficará o array resultante?

**Exemplo de Entrada:** // ['Ana', 'Bruno', 'Carlos', 'Diana', 'Eduardo'];

**Exemplo de Saída:** // [ { id: 1, nome: 'Ana' }, { id: 2, nome: 'Bruno' }, { id: 3, nome: 'Carlos' }, { id: 4, nome: 'Diana' }, { id: 5, nome: 'Eduardo' } ]

**Para os exercícios 4, 5 e 6 utilize a estrutura do objeto abaixo:**

```
const imoveis = [ { id: 1, tipo: 'Apartamento', valor: 400000, disponivel: true, bairro: 'Centro' }, { id: 2, tipo: 'Casa', valor: 550000, disponivel: false, bairro: 'Subúrbio' }, { id: 3, tipo: 'Cobertura', valor: 900000, disponivel: true, bairro: 'Centro' }, { id: 4, tipo: 'Apartamento', valor: 300000, disponivel: false, bairro: 'Zona Sul' }, { id: 5, tipo: 'Casa', valor: 700000, disponivel: true, bairro: 'Centro' } ];
```

**Exercício 4:** Você tem uma lista de imóveis, alguns disponíveis para venda e outros não. Utilize o método `.filter()` para criar uma nova lista contendo apenas os imóveis que estão disponíveis para venda. Qual será o array resultante?

**Exemplo de Saída:** // [ { id: 1, tipo: 'Apartamento', valor: 400000, disponivel: true, bairro: 'Centro' }, { id: 3, tipo: 'Cobertura', valor: 900000, disponivel: true, bairro: 'Centro' }, { id: 5, tipo: 'Casa', valor: 700000, disponivel: true, bairro: 'Centro' } ]

**Exercício 5:** Você tem uma lista de imóveis com seus respectivos valores. Utilize o método `.filter()` para criar uma nova lista contendo apenas os imóveis que têm valor acima de R\$ 500.000. Qual será o array resultante?

**Exemplo de Saída:** // [ { id: 2, tipo: 'Casa', valor: 550000, disponivel: false, bairro: 'Subúrbio' }, { id: 3, tipo: 'Cobertura', valor: 900000, disponivel: true, bairro: 'Centro' }, { id: 5, tipo: 'Casa', valor: 700000, disponivel: true, bairro: 'Centro' } ]

**Exercício 6:** Você tem uma lista de imóveis em diferentes bairros da cidade. Utilize o método `.filter()` para criar uma nova lista contendo apenas os imóveis localizados no bairro "Centro". Qual será o array resultante?

**Exemplo de Saída:** // [ { id: 1, tipo: 'Apartamento', valor: 400000, disponivel: true, bairro: 'Centro' }, { id: 3, tipo: 'Cobertura', valor: 900000, disponivel: true, bairro: 'Centro' }, { id: 5, tipo: 'Casa', valor: 700000, disponivel: true, bairro: 'Centro' } ]

**Exercício 7:** Você tem um array de objetos representando as vendas de uma empresa, onde cada objeto contém o valor da venda. Utilize o método `.reduce()` para calcular o faturamento total da empresa. Qual será o valor total?

**Exemplo de Entrada:** // [ { id: 1, valor: 1500 }, { id: 2, valor: 2300 }, { id: 3, valor: 3200 }, { id: 4, valor: 4100 }, { id: 5, valor: 1800 } ];

**Exemplo de Saída:** // 12900

**Exercício 8:** Você tem um array de objetos representando avaliações de clientes, onde cada objeto contém uma pontuação de satisfação (de 1 a 5). Utilize o método `.reduce()` para calcular a média de satisfação dos clientes. Qual será a média?

**Exemplo de Entrada:** // [ { cliente: 'João', pontuacao: 4 }, { cliente: 'Maria', pontuacao: 5 }, { cliente: 'Pedro', pontuacao: 3 }, { cliente: 'Ana', pontuacao: 4 }, { cliente: 'Lucas', pontuacao: 5 } ];

**Exemplo de Saída:** // 4.2

**Exercício 9:** Você tem um array de objetos representando produtos vendidos, onde cada objeto contém o nome do produto e sua categoria. Utilize o método `.reduce()` para criar um objeto que conte quantos produtos foram vendidos em cada categoria. Como ficará o objeto resultante?

**Exemplo de Entrada:** `// [ { nome: 'Camisa', categoria: 'Vestuário' }, { nome: 'Calça', categoria: 'Vestuário' }, { nome: 'Notebook', categoria: 'Eletrônicos' }, { nome: 'Celular', categoria: 'Eletrônicos' }, { nome: 'Tênis', categoria: 'Calçados' }, { nome: 'Meias', categoria: 'Vestuário' } ]`;  
**Exemplo de Saída:** `// { 'Vestuário': 3, 'Eletrônicos': 2, 'Calçados': 1 }`

**Exercício 10:** Crie uma função tradicional chamada `calcularDesconto` que receba o preço de um produto e a porcentagem de desconto, e retorne o valor do produto com o desconto aplicado. Após criar a função tradicional, transforme-a em uma Arrow Function.

**Exemplo de Entrada:** `//const precoProduto = 200; const percentualDesconto = 15;`  
**Exemplo de Saída:** `// 170;`

**Exercício 11:** Crie uma função tradicional chamada `filtrarPorCategoria` que receba uma lista de produtos e uma categoria, e retorne os produtos que pertencem àquela categoria. Após criar a função tradicional, transforme-a em uma Arrow Function.

**Exemplo de Entrada:** `// const produtos = [ { nome: 'Camisa', categoria: 'Vestuário' }, { nome: 'Notebook', categoria: 'Eletrônicos' }, { nome: 'Tênis', categoria: 'Calçados' }, { nome: 'Calça', categoria: 'Vestuário' } ]`; `const categoriaDesejada = 'Vestuário'`;  
**Exemplo de Saída:** `// [ { nome: 'Camisa', categoria: 'Vestuário' }, { nome: 'Calça', categoria: 'Vestuário' } ]`;

**Exercício 12:** Crie uma função tradicional chamada `calcularFaturamentoTotal` que receba uma lista de vendas e retorne o faturamento total somando o valor de todas as vendas. Após criar a função tradicional, transforme-a em uma Arrow Function.

**Exemplo de Entrada:** `// const vendas = [ { id: 1, valor: 1500 }, { id: 2, valor: 2300 }, { id: 3, valor: 3200 } ]`;  
**Exemplo de Saída:** `// 7000;`