

Roteiro V - Javascript ES6

- 01)** Desenvolva uma função assíncrona que utilize `Promise` e `setTimeout` para simular uma requisição a um servidor. Use apenas *arrow functions*.
- 02)** Desenvolva uma função que receba um número indefinido de parâmetros e retorne um valor booleano informando se existem parâmetros do mesmo tipo e valor. Utilize o operador `rest`, apenas *arrow functions* e *callback functions*.
- 03)** Use `spread` para combinar dois objetos e exibir o resultado. Use apenas *arrow functions* e *callback functions*.
- 04)** Crie uma função que aceite múltiplos arrays como argumentos e os combine em um único array ordenado usando `spread`. Use apenas *arrow functions* e *callback functions*.
- 05)** Escreva duas funções arrow usando a técnica de promises (Javascript ES6). A primeira função, `colocarTodasLetrasEmMaiusculoEm500ms`, receberá um único parâmetro e o colocará em maiúsculo depois de 500 milissegundos. Verifique, dentro da função, se o parâmetro se trata de um tipo `String`. Se o parâmetro não for do tipo `String`, a promise deve rejeitar o parâmetro. A segunda função, `inverteTodasLetras()`, realizará uma inversão das letras da `String`. Atenção: você deverá utilizar encadeamento de funções.
- 06)** Transforme as funções desenvolvidas na questão 04 para usarem a técnica `async/await` ao invés de só usar promises.
- 07)** Você está desenvolvendo o módulo de gerenciamento de produtos de uma aplicação web e precisa integrar-se com uma API REST fictícia criada no mockable.io.

Sua tarefa será criar dois endpoints, no mockable.io:

- `GET /produtos` → retorna uma lista de produtos no formato JSON:

```
[
  { "id": 1, "nome": "Notebook X", "preco": 3500.00 },
  { "id": 2, "nome": "Mouse Gamer", "preco": 150.00 },
  { "id": 3, "nome": "Teclado Mecânico", "preco": 500.00 }
]
```

- POST /produtos → recebe um produto em formato JSON e retorna o mesmo objeto com um campo extra "status": "criado".

Em JavaScript, construa um código que:

Faça uma requisição GET para /produtos e exiba no console todos os produtos com formatação:

```
Produto: Notebook X | Preço: R$ 3500.00  
Produto: Mouse Gamer | Preço: R$ 150.00  
Produto: Teclado Mecânico | Preço: R$ 500.00
```

Adicione o produto à lista retornada e envie este produto via POST para o endpoint /produtos criado, recebendo a resposta da API.

Exiba no console uma mensagem confirmando a criação do produto, no formato:

```
Produto "Headset" criado com sucesso! Status: criado
```

Diretrizes para a solução:

- Utilize async/await e tratamento de erros com try...catch.
 - Crie uma função separada para buscar produtos e outra para cadastrar produtos.
1. Comente cada etapa do código explicando sua função.