

JavaScript Impressionador - Módulo 9 - JavaScript Assíncrono

Exercícios

Callbacks:

Exercício 1: Callback Simples

Escreva uma função **imprimirMensagem** que aceita uma mensagem e uma função de retorno de chamada (callback). A função deve imprimir a mensagem no console e, em seguida, invocar a função de retorno de chamada.

Exercício 2: Leitura de Arquivo com Callback

Escreva uma função **lerArquivo** que aceita um nome de arquivo e uma função de retorno de chamada (callback). A função deve ler o conteúdo do arquivo e passá-lo para a função de retorno de chamada. Por fim, chame a função lerArquivo() usando um callback que faça o conteúdo do arquivo ser exibido no console.

Dica: disponibilizamos um arquivo de texto para esse exercício. Coloque-o na mesma pasta do seu código JavaScript pra facilitar a resolução.

Obs: Para esse exercício, aqui vai um trecho de código que mostra como disponibilizar no seu código as funcionalidades de uso de arquivos com JavaScript/NodeJS. O Trecho abaixo deve ser utilizado para a leitura do arquivo.

```
const fs = require('fs');

//o método readFile lê o conteúdo de um arquivo identificado
//por filename (coloque o nome do arquivo e o caminho dele,
// ex: './textoExercicio2.txt' ) e entrega esse conteúdo dentro //do objeto "data"

fs.readFile(filename, (err, data) => {
  if (err) {
    //Aqui escrevemos um tratamento para um erro na tentativa de leitura do arquivo
    return;
  }
  //Aqui escrevemos o que deve ser feito com o objeto "data"
});
```

Promises:

Exercício 3: Conceitos Básicos de Promises

Crie uma nova Promise que seja resolvida após um atraso de 1 segundo e retorne a string "Promise resolvida". Use .then() para lidar com o valor resolvido e imprimir no console.

Exercício 4: Encadeamento de Promises

Crie duas Promises: uma que seja resolvida para "Olá" após um atraso de 1 segundo e outra que seja

resolvida para "Mundo" após um atraso de 2 segundos. Encadeie essas Promises usando `.then()` para imprimir "Olá, Mundo" no console.

Exercício 5: Tratamento de Erros com Promises

Crie uma Promise que seja rejeitada com um erro após um atraso de 1 segundo. Use `.catch()` para lidar com o erro e imprimir no console.

Exercício 6: Promise All

Crie três Promises: uma que seja resolvida para "Um" após um atraso de 1 segundo, outra que seja resolvida para "Dois" após um atraso de 2 segundos e uma terceira que seja resolvida para "Três" após um atraso de 3 segundos. Use `Promise.all()` para aguardar a resolução de todas as Promises e, em seguida, imprima o array de valores resolvidos no console.

Async/Await:

Exercício 7: Async/Await Básico

Escreva uma função assíncrona `buscarDados` que usa `await` para simular a busca de dados de uma API após um atraso de 2 segundos. Imprima os dados buscados no console.

Exercício 8: Tratamento de Erros com Async/Await

Escreva uma função assíncrona `obterDados` que usa `await` para buscar dados de uma API após um atraso de 1 segundo. Se ocorrer um erro, capture-o e imprima uma mensagem de erro no console.

Exercício 9: Async/Await com Promise.all()

Escreva uma função assíncrona `buscarMultiplosDados` que usa `Promise.all()` e `await` para buscar dados de várias APIs simultaneamente. Imprima o array de dados buscados no console.

Exercício 10: Async/Await com Tratamento de Erros

Escreva uma função assíncrona `obterDadosComFallback` que usa `try...catch` e `await` para buscar dados de uma API. Se ocorrer um erro, utilize um valor de fallback e imprima-o no console.