

Async/await

As palavras-chave *async* e *await* são uma sintaxe que simplifica a programação assíncrona, facilitando o fluxo de escrita e leitura do código. Assim é possível escrever código que funciona de forma assíncrona, porém lido e estruturado de forma síncrona.

Definindo uma função como `async`, podemos utilizar a palavra-chave `await` antes de qualquer expressão que retorne uma *Promise*. Dessa forma, a execução da função externa (a função `async`) será pausada até que a *Promise* seja resolvida.

Uma função declarada como *async* significa que o valor de retorno da função será, "por baixo dos panos", uma *Promise*. Se a *Promise* se resolver normalmente, o objeto-*Promise* retornará o valor. Caso lance uma exceção, podemos usar o *try/catch* como estamos acostumados em programas síncronos.

O *async/await* surgiu como uma opção mais "legível" ao `then()`. É possível usar os dois métodos em um mesmo código e, a partir da versão 10 do Node.js, ambas as formas são equivalentes em termos de performance. O *async/await* simplifica a escrita e a interpretação do código, mas não é tão flexível e só funciona com uma *Promise* por vez.

Referência

AMOASEI, Juliana. **Async/await no JavaScript:** o que é e quando usar a programação assíncrona? Disponível em: <<https://www.alura.com.br/artigos/async-await-no-javascript-o-que-e-e-quando-usar>>. Acesso em: 10 mar. 2023. Com adaptações.