

Dado um número inteiro **millis**, escreva uma função assíncrona que durma por **millis** milissegundos. Ele pode resolver qualquer valor.

Resolução

```
async function sleep(millis) {  
  return new Promise((resolve) => {  
    setTimeout(() => {  
      resolve(1);  
    }, millis);  
  });  
}  
  
let t = Date.now()  
sleep(100).then(() => console.log(Date.now() - t)) // 100
```

1. Definição da função assíncrona `sleep`:

A função `sleep` é definida como uma função assíncrona usando a palavra-chave `async`. Isso indica que a função conterá operações assíncronas e pode retornar uma promessa. A função recebe um argumento `millis`, que representa o tempo em milissegundos que você deseja esperar.

2. Retorno de uma promessa dentro da função `sleep`:

Dentro da função `sleep`, é criada uma nova promessa usando o construtor `Promise`. Essa promessa envolve uma operação de tempo, especificamente, um `setTimeout`. O `setTimeout` é usado para agendar uma função para ser executada após um certo número de milissegundos (especificado pelo argumento `millis`).

3. Execução do `setTimeout`:

Dentro do `setTimeout`, é definida uma função de retorno que será executada quando o temporizador expirar (após o tempo especificado em `millis`). Nesse caso, a função de retorno simplesmente chama `resolve(1)`. O `resolve` é um argumento passado para a função interna do construtor de promessa que permite que a promessa seja resolvida com um valor. Neste caso, a promessa será resolvida com o valor `1`.

4. Chamada da função `sleep`:

Depois de definir a função `sleep`, a próxima linha de código chama a função `sleep` com um argumento de `100` milissegundos. Isso significa que a função vai esperar por 100 milissegundos antes de resolver a promessa com o valor `1`.

5. `.then()` e impressão do resultado:

Após chamar a função `sleep`, é encadeado um `.then()` para a promessa retornada. O `.then()` permite que você especifique uma função a ser executada quando a promessa for resolvida. Nesse caso, a função de impressão é usada para calcular o tempo decorrido entre quando a função `sleep` foi chamada e quando a promessa foi resolvida. Isso é feito subtraindo o valor retornado por `Date.now()` (representando o tempo atual em milissegundos) no início da função `sleep` do valor retornado por `Date.now()` após a promessa ser resolvida.