

### Arrays:

1. Crie uma função chamada **somaArray** que recebe um array de números e retorna a soma de todos os elementos.
2. Crie uma função chamada **maiorNumero** que recebe um array de números e retorna o maior número.
3. Crie uma função chamada **mediaArray** que recebe um array de números e retorna a média dos valores.

### Objetos:

4. Crie uma função chamada **contarPropriedades** que recebe um objeto e retorna o número de propriedades no objeto.
5. Crie uma função chamada **juntarObjetos** que recebe dois objetos e retorna um novo objeto que combina as propriedades de ambos.
6. Crie uma função chamada **encontrarPropriedade** que recebe um objeto e uma chave e retorna o valor da propriedade correspondente.

### Promises:

7. Crie uma função chamada **delayedPromise** que retorna uma promise que é resolvida após um atraso de 1 segundo.
8. Crie uma função chamada **fetchData** que faz uma solicitação HTTP usando fetch e retorna uma promise com os dados obtidos.
9. Crie uma função chamada **gerarNumeroAleatorio** que retorna uma promise com um número aleatório após um atraso de 2 segundos.

### Timeout:

10. Crie uma função chamada **executarAposTempo** que recebe uma função e um tempo em milissegundos e a executa após o tempo especificado.
11. Crie uma função chamada **cancelarExecucao** que recebe um identificador de timeout e o cancela.
12. Crie uma função chamada **intervaloPersonalizado** que executa uma função a cada intervalo de tempo especificado em milissegundos.

### Interval:

13. Crie uma função chamada **contadorInterval** que inicia um contador de 1 a 10 a cada segundo e exibe o valor no console.

14. Crie uma função chamada **pararContador** que interrompe o contador iniciado na função anterior.
15. Crie uma função chamada **gerarNumerosAleatorios** que gera números aleatórios a cada 500 milissegundos e os exibe no console.

#### **Desestruturação e Arrays:**

16. Crie uma função chamada **primeiroEUltimo** que recebe um array com pelo menos dois elementos e retorna um novo array contendo apenas o primeiro e o último elementos do array original.
17. Crie uma função chamada **extrairInfo** que recebe um array de objetos, onde cada objeto contém as propriedades nome e idade. A função deve retornar um novo array contendo apenas as idades dos objetos.
18. Crie uma função chamada **separarArrays** que recebe um array contendo números e o desestrutura em dois arrays: um contendo os números pares e outro contendo os números ímpares.

#### **Desestruturação e Objetos:**

19. Crie uma função chamada **infoPessoa** que recebe um objeto com as propriedades nome, idade e cidade. A função deve retornar uma string formatada, como "Nome: [nome], Idade: [idade], Cidade: [cidade]".
20. Crie uma função chamada **desestruturarUsuario** que recebe um objeto com as propriedades id, nome e email e retorna um novo objeto contendo apenas as propriedades id e email.
21. Crie uma função chamada **informacoesLivro** que recebe um objeto representando as informações de um livro com as propriedades titulo, autor e ano. A função deve desestruturar o objeto para extrair essas informações e retorná-las em uma string formatada como "O livro [titulo] foi escrito por [autor] em [ano]".

#### **Rest e Spread:**

22. Crie uma função chamada **concatenarArrays** que recebe dois ou mais arrays e os concatena em um único array usando o operador de spread.
23. Crie uma função chamada **somaNumeros** que aceita um número variável de argumentos. A função deve calcular a soma de todos os números passados como argumentos e retornar o resultado.
24. Crie uma função chamada **mesclarObjetos** que aceita dois ou mais objetos como argumentos. A função deve mesclar esses objetos em um único objeto usando o

operador spread e retornar o objeto resultante. Em caso de conflito de propriedades, a última propriedade encontrada deve prevalecer.