

Vitor Amadeu Souza

Introdução ao

TypeScript

Parte V

© 2025 by Cerne Tecnologia e Treinamento Ltda.

© 2025 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Fevereiro de 2025

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

“Conhecimento é poder.”

Francis Bacon

Cerne Tecnologia

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos eletrônicos em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android, iOS, Blackberry, Windows Phone e no desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes.

Desenvolvemos esquemas elétricos e layout de PCI, tanto em tecnologia convencional como SMD.

Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço cerne-tec.com.br para obter mais informações.



cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento	6
1. Introdução	6
Capítulo II – Programação em TypeScript	7
1. Spread Operator (em Arrays e Objetos)	7
2. Rest Parameters	8
3. Função de Array (filter)	9
4. Função de Array (reduce)	11
5. Classes com Métodos Estáticos	12
6. Funções Assíncronas	13
7. Desestruturação de Objetos	14
8. Desestruturação de Arrays	16
9. Classes com Getter e Setter	17
10. Interface com Função	19

Capítulo I

Metodologia de desenvolvimento

1. Introdução

Este livro tem como objetivo ensinar o desenvolvimento web utilizando TypeScript, explorando seus recursos de forma prática e aplicada. Ao longo desta série, diversas aplicações serão desenvolvidas, permitindo uma compreensão aprofundada da linguagem e de suas principais funcionalidades. Cada volume abordará temas específicos, conforme detalhado no sumário, proporcionando um aprendizado progressivo e estruturado.

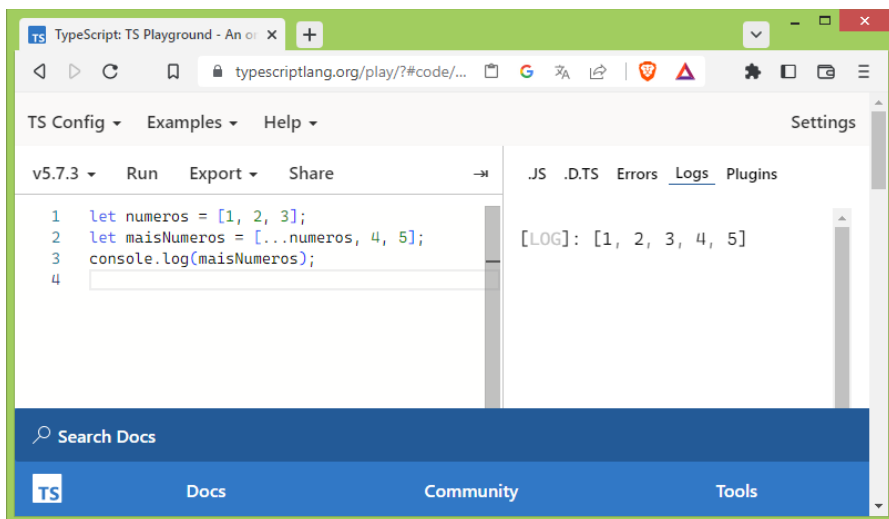
Capítulo II

Programação em TypeScript

1. Spread Operator (em Arrays e Objetos)

No código, o operador de espalhamento (...) é usado para criar um novo array. A variável `numeros` contém o array `[1, 2, 3]`. Em seguida, a variável `maisNumeros` é criada copiando todos os elementos de `numeros` para um novo array e adicionando os valores 4 e 5 ao final. Assim, o array `maisNumeros` fica com os elementos `[1, 2, 3, 4, 5]`. Quando o `console.log` é executado, esse array é impresso no console, mostrando a combinação dos elementos de `numeros` com os novos valores. O uso do operador de espalhamento permite combinar arrays de maneira simples e imutável.

```
let numeros = [1, 2, 3];  
  
let maisNumeros = [...numeros, 4, 5];  
  
console.log(maisNumeros);
```

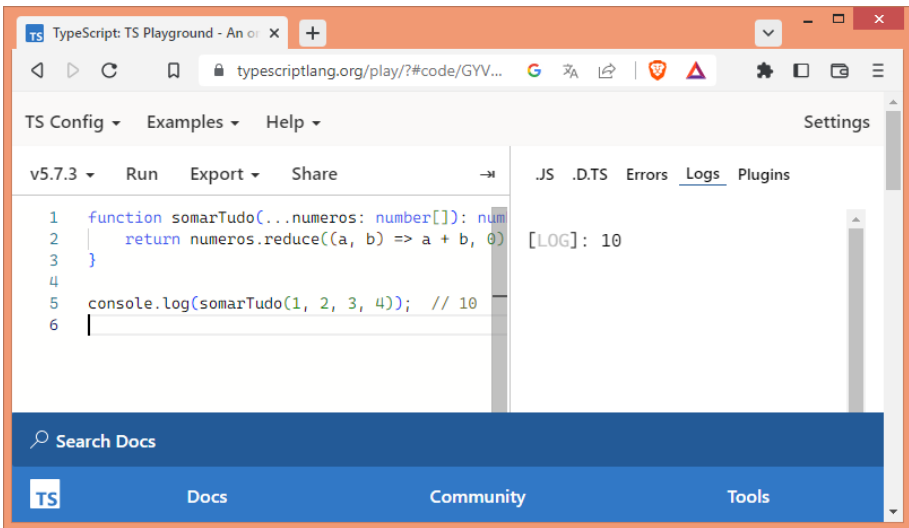


2. Rest Parameters

No código, a função `somarTudo` utiliza o operador rest (...) para permitir que um número variável de argumentos seja passado para ela, todos do tipo `number`. Esses argumentos são coletados em um array chamado `numeros`. Dentro da função, o método `reduce` é usado para somar todos os elementos do array. A função `reduce` começa com um valor inicial de 0 e soma cada número ao acumulado. Quando a função `somarTudo` é chamada com os valores 1, 2, 3, 4, a soma é calculada, resultando em 10. O `console.log` então imprime esse valor. Esse tipo de abordagem é útil para trabalhar com

um número indefinido de argumentos e realizar operações sobre eles de forma eficiente.

```
function somarTudo(...numeros: number[]): number {  
    return numeros.reduce((a, b) => a + b, 0);  
}  
  
console.log(somarTudo(1, 2, 3, 4)); // 10
```



3. Função de Array (filter)

No código, a variável `numeros` contém um array com os valores de 1 a 5. O método `filter` é utilizado para criar um novo array, `pares`, que inclui apenas os números do array original que são

divisíveis por 2, ou seja, os números pares. A função de callback passada para o filter verifica se cada número é par utilizando a condição `n % 2 === 0`. Como resultado, o array pares será composto pelos números 2 e 4. Portanto, o `console.log` imprime o array `[2, 4]`. O método `filter` é útil para selecionar elementos de um array que atendem a uma condição específica.

```
let numeros = [1, 2, 3, 4, 5];  
let pares = numeros.filter(n => n % 2 === 0);  
console.log(pares);
```

