

5. Funções em TypeScript

Funções são blocos fundamentais no desenvolvimento de software, permitindo reutilizar código e criar estruturas modulares. TypeScript adiciona uma camada de segurança adicional com a tipagem de parâmetros e valores de retorno, ajudando a evitar muitos erros comuns.

Definindo Funções

1. Função com Tipos nos Parâmetros e Retorno

- Em TypeScript, você pode especificar o tipo de cada parâmetro e o tipo do valor de retorno.

```
function soma(a: number, b: number): number {  
    return a + b;  
}
```

2. Funções Anônimas e Arrow Functions

- Funções anônimas e arrow functions também suportam tipagem nos parâmetros e no retorno.

```
const multiplica = (x: number, y: number): number => x * y;
```

Parâmetros Opcionais e Padrões

• Parâmetros Opcionais

- Parâmetros opcionais são marcados com um **?** e devem vir após os parâmetros obrigatórios.

```
function saudacao(nome: string, sobrenome?: string): string {  
    return `Olá, ${nome} ${sobrenome || ''}!`;  
}
```

• Parâmetros com Valores Padrão

- Você pode definir valores padrão para parâmetros, que serão usados caso nenhum valor seja fornecido.

```
function potencia(base: number, expoente: number = 2): number {  
    return base ** expoente;  
}
```

Tipos Avançados em Funções

- Tipos de Função
 - Você pode criar um tipo específico para uma função, que define a assinatura da função.

```
type OperacaoMatematica = (x: number, y: number) => number;  
const subtrai: OperacaoMatematica = (x, y) => x - y;
```

- Overloads de Função
 - TypeScript permite definir múltiplas assinaturas para uma função, conhecidas como overloads, permitindo que a função aceite diferentes tipos de parâmetros e retorne diferentes tipos de valores.

```
function ajustarValor(valor: number): number;  
function ajustarValor(valor: string): string;  
function ajustarValor(valor: any): any {  
    return typeof valor === 'number' ? valor.toFixed(2) : valor.trim();  
}
```

Exercícios para Praticar

1. Implementando Funções:

- Escreva uma função `concatenaNomes` que recebe dois parâmetros, `nome` e `sobrenome`, e retorna o nome completo.

2. Uso de Arrow Functions:

- Converta a função `soma` para uma arrow function e implemente a mesma lógica.

3. Explorando Parâmetros Opcionais:

- Modifique a função `saudacao` para incluir um parâmetro opcional `titulo`, que se usado, precederá o nome no cumprimento.

4. Trabalhando com Overloads:

- Crie uma função `ajustar` que pode receber tanto um `number` quanto uma `string`, retornando o valor ajustado conforme o tipo.

5. Função com Valor Padrão:

- Implemente uma função `incrementa` que recebe um número e um valor de incremento opcional, que, se não fornecido, será `1`.