

4. Arrays em TypeScript

Arrays em TypeScript são utilizados para armazenar múltiplos valores em uma única variável. Com a adição de tipos, o TypeScript permite que você defina o tipo dos elementos dentro do array, garantindo maior controle e segurança sobre os dados que você manipula.

garantindo maior controle e segurança sobre os dados que você manipula.

Definindo Arrays

1. Array com Tipo Único

- Você pode declarar um array onde todos os elementos são do mesmo tipo usando a notação de colchetes após o tipo de dado.

```
let numeros: number[] = [1, 2, 3, 4, 5];  
let nomes: string[] = ["Ana", "Beatriz", "Carlos"];
```

2. Array com Generic Array Type

- Alternativamente, TypeScript oferece a sintaxe genérica **Array<tipo>** para declarar um array.

```
let frutas: Array<string> = ["maçã", "banana", "manga"];  
let pontos: Array<number> = [9.5, 7.3, 8.6];
```

Operações Comuns em Arrays

• Adicionando Elementos

- Use o método **.push()** para adicionar elementos ao final de um array.

```
numeros.push(6);
```

Removendo Elementos

Use **.pop()** para remover o último elemento de um array.

```
let ultimoNumero = numeros.pop(); // Remove e retorna o último elemento
```

- **Acessando Elementos**
 - Acesse um elemento específico pelo seu índice.

```
let primeiroNome = nomes[0]; // "Ana"
```

- **Iterando sobre Elementos**
 - Arrays em TypeScript podem ser iterados usando loops como **for** ou métodos como **.forEach()**.

```
numeros.forEach((numero) => {  
    console.log(numero);  
});
```

Métodos Avançados de Arrays

- **Filtrar Elementos**
 - Use **.filter()** para criar um novo array com elementos que satisfazem uma condição específica.

```
let numerosFiltrados = numeros.filter(n => n > 2);
```

- **Reduzindo um Array**
 - O método **.reduce()** aplica uma função que resulta em um único valor resumido de todo o array.

```
let soma = numeros.reduce((acumulador, atual) => acumulador + atual, 0);
```

Exercícios para Praticar

1. Criação e Acesso:

- Crie um array de strings com os nomes de cinco cidades. Em seguida, escreva um código que imprime o nome da terceira cidade na lista.

2. Adicionando e Removendo Elementos:

- Dado o array `[10, 20, 30, 40, 50]`, adicione o número `35` entre `30` e `40`, e depois remova o número `20` do array. Imprima o array final.

3. Método `.map()`:

- Crie um array de números `[1, 2, 3, 4, 5]`. Use o método `.map()` para criar um novo array onde cada número é multiplicado por 3. Imprima o novo array.

4. Filtrando Valores:

- Utilize o método `.filter()` para criar um novo array contendo apenas os números ímpares do array original `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`.

5. Usando `.reduce()` para Soma:

- Dado um array de números `[5, 7, 10, 12, 15]`, utilize o método `.reduce()` para calcular e imprimir a soma total dos elementos do array.