

types em arrays no typescript e como usar ?



> Em TypeScript, é possível definir tipos para os elementos de um array. Existem duas maneiras de fazer isso:

1. Usando a sintaxe de `tipo[]`

typescript

Copy code

```
let array: number[] = [1, 2, 3];
```

Nesse exemplo, o tipo do array é `number[]`, o que significa que ele contém apenas números.

2. Usando a sintaxe de `Array<tipo>`

typescript

Copy code

```
let array: Array<number> = [1, 2, 3];
```

Nesse exemplo, o tipo do array é `Array<number>`, o que também significa que ele contém apenas números.

Além disso, é possível usar tipos genéricos para definir o tipo dos elementos do array. Por exemplo, para criar um array de strings ou números, você pode usar a sintaxe genérica:

typescript

Copy code

```
let array: Array<string | number> = ["hello", 42];
```

Nesse exemplo, o tipo do array é `Array<string | number>`, o que significa que ele pode conter tanto strings quanto números.

Para acessar os elementos do array, você pode usar a notação de colchetes e o índice do elemento que deseja acessar:

typescript

 Copy code

```
let array: number[] = [1, 2, 3]; let firstElement: number = array[0]; //  
primeiro elemento (índice 0) let secondElement: number = array[1]; // segundo  
elemento (índice 1)
```

Também é possível usar métodos como `push`, `pop`, `shift` e `unshift` para adicionar ou remover elementos do array:

typescript

 Copy code

```
let array: number[] = [1, 2, 3]; array.push(4); // adiciona o número 4 no  
final do array array.pop(); // remove o último elemento do array  
array.unshift(0); // adiciona o número 0 no início do array array.shift(); //  
remove o primeiro elemento do array
```
