

Programação Orientada a Objetos

Arrays e Strings

Ely – ely.miranda@ifpi.edu.br

Arrays

- Conjunto de tipos primitivos ou de objetos indexáveis da posição 0 a (tamanho-1);
- Podem ser declarados de forma simples:
let numeros: **number**[];
- Podem ser declarados e inicializados:
let numeros: **number**[]; = [1, 2, 3];

ely.miranda@ifpi.edu.br

2

Arrays

- Conjunto de tipos primitivos ou de objetos indexáveis da posição 0 a (tamanho-1);
- Podem ser declarados como “arrays tipados” usando generics:

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

ely.miranda@ifpi.edu.br

3

Arrays

- O acesso aos elementos dos arrays se dá através de índices/posições:

```
let fruits: string[] =  
    ['Apple', 'Orange', 'Banana'];  
console.log(fruits[1]); // Orange
```

ely.miranda@ifpi.edu.br

4

Arrays

- O acesso a posições não existentes retorna undefined:

```
let fruits: string[] =  
    ['Apple', 'Orange', 'Banana'];  
console.log(fruits[3]); // undefined
```

ely.miranda@ifpi.edu.br

5

Arrays

- Novos elementos podem ser adicionados dinamicamente por atribuição:

```
let fruits: string[] =  
    ['Apple', 'Orange', 'Banana'];  
fruits[3] = 'Strawberry';  
console.log(fruits[3]); // Strawberry
```

ely.miranda@ifpi.edu.br

6

Arrays

- Podem conter elementos de diferentes tipos de dados usando a sintaxe abaixo:

```
let values: (string | number)[] =  
    ['Apple', 2, 'Orange', 3, 4, 'Banana'];  
// ou  
let values: Array<string | number> =  
    ['Apple', 2, 'Orange', 3, 4, 'Banana'];
```

ely.miranda@ifpi.edu.br

7

Arrays

- O atributo length armazena o tamanho do array;
- Acessando elementos e percorrendo através de loops:

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];  
  
for (var i = 0; i < fruits.length; i++) {  
    console.log(fruits[i]);  
    // output: Apple Orange Banana  
}
```

ely.miranda@ifpi.edu.br

8

Arrays

- O atributo length armazena o tamanho do array;
- Acessando elementos e percorrendo através de loops:

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];

for (var i in fruits) {
    console.log(fruits[i]);
    // output: Apple Orange Banana
}
```

ely.miranda@ifpi.edu.br

9

Arrays

- Arrays são objetos e possuem alguns métodos:

```
let numeros: number[] = [1, 2, 3];
numeros.push(4);
console.log(numeros.reverse()); // [4,3,2,1]
```

<https://www.tutorialsteacher.com/typescript/typescript-array>

ely.miranda@ifpi.edu.br

10

Strings

- Strings também são objetos, mas não precisam ser instanciadas;
- São mantidas internamente como arrays de caracteres;
- O operador "+" é sobrecarregado:
 - Usado para concatenar Strings;

```
let s : String = "Bom";  
s = s + " dia";
```

ely.miranda@ifpi.edu.br

11

Strings

- Não há uma distinção explícita sobre o uso de aspas simples ou duplas;

```
let color: string = "azul";  
color = 'vermelho' + " escuro";
```
- Há ainda um terceiro tipo de “marcação” para strings: as template strings.

ely.miranda@ifpi.edu.br

12

Template Strings

- Podem ter múltiplas linhas sem necessidade de caracteres de escape;
- Também podem conter expressões;
- São delimitadas por **crases** ``:

```
let nome : string = "Ely Miranda";  
let idade: number = 39;  
let frase: string = `Meu nome é ${nome}.  
  
Completarei ${ idade + 1 } mês que vem.`;  
console.log(frase);
```

ely.miranda@ifpi.edu.br

13

Alguns métodos úteis

- `charAt()`: retorna o caractere na posição especificada pelo índice/parâmetro;

```
let str: string = 'Hello TypeScript';  
console.log(str.charAt(0)); // 'H'  
console.log(str.charAt(2)); // 'l'  
console.log("Hello World".charAt(2)); // 'l'
```

ely.miranda@ifpi.edu.br

14

Alguns métodos úteis

- `indexOf ()`: retorna o índice da primeira ocorrência da string procurada;
- Retorna -1 se a string não for encontrada;

```
let str: string = 'TypeScript';  
console.log(str.indexOf('T')); // 0  
console.log(str.indexOf('p')); // 2  
console.log(str.indexOf('x')); // -1
```

ely.miranda@ifpi.edu.br

15

Alguns métodos úteis

- `indexOf ()`: retorna o índice da primeira ocorrência da string procurada;
- Permite que se passe opcionalmente a partir de que posição deve ser feita a pesquisa:

```
let str: string = 'TypeScript';  
console.log(str.indexOf('T', 1)); // -1  
console.log(str.indexOf('p', 3)); // 8
```

ely.miranda@ifpi.edu.br

16

Alguns métodos úteis

- `replace()`: substitui uma ocorrência de uma string em outra;

```
let str1: string = 'Hello Javascript';  
console.log(str1.replace('Java', 'Type'));  
// 'Hello TypeScript'
```

-

ely.miranda@ifpi.edu.br

17

Alguns métodos úteis

- `split()`: retorna um array com “pedaços” de uma string de acordo com um separador;

```
let str1: string = 'Apple, Banana, Orange';  
  
console.log(str1.split(',')); //  
[ 'Apple', ' Banana', ' Orange' ]
```

ely.miranda@ifpi.edu.br

18

Programação Orientada a Objetos

Arrays e Strings

Ely – ely.miranda@ifpi.edu.br