

# Saiba sobre os conceitos básicos do TypeScript





## o que é TypeScript?

É uma linguagem de programação e **superconjunto** do js desenvolvida pela Microsoft, surgida em 2012, atualmente muito usado em diversos frameworks JS como o Angular, React e Vue.

o “**TypeScript ser superconjunto de JavaScript**” significa que todo **código JS funciona em TS**, mas no TypeScript dá pra usar recursos a mais, como a tipagem de variáveis.

Na prática, o código TypeScript é transpilado(convertido) para JavaScript, já que os navegadores só entendem JS.

# Tipagem estática

o TypeScript adiciona tipagem estática, isto é, o programador precisa **especificar o tipo de dado**



```
let nome: string = "João";  
let idade: number = 25; /*integer*/  
let altura: number = 1.75; /*float*/  
let casado: boolean = false;
```

# Type Inference e Union Type

Type Inference: TypeScript deduz automaticamente o tipo de uma variável com base no valor atribuído.

Union Types: o TypeScript permite que uma variável tenha mais de um tipo, como `string | number`.

OBS: muito cuidado ao utilizar isso, pois usando vários tipos numa variável quebra a lógica do TypeScript de ser fortemente tipado, não trazendo a segurança e robustez do código



```
/*TYPE INFERENCE*/
```

```
let nome = "Carlos"; // TypeScript infere  
que nome é string
```

```
nome = 10; // ✗ Erro: O tipo 'número' não  
pode ser atribuído ao tipo 'string'
```

```
/*UNION TYPE*/
```

```
let objetos: string | number;
```

```
objetos = "Bola";
```

```
/* OU */
```

```
objetos = 100;
```

# Arrays e Tuplas

Arrays: no TS, você pode definir o tipo dos dados que a array vai armazenar, sendo rigoroso com o tipo de dado que você vai especificar .

Tuplas: é um tipo de array, mas com uma diferença: você define quantos elementos ele vai ter, por exemplo 2 elementos, e qual o tipo de cada um, por exemplo um é caractere, e outro é inteiro.



```
/*PRIMEIRA FORMA DE ARRAY*/
```

```
let nomes: Array<string> = ["Ana", "Lucas",  
"Carlos"];
```

```
/*SEGUNDA FORMA DE ARRAY*/
```

```
let nomes: string[] = ["Ana", "Lucas",  
"Carlos"];
```

```
/*TUPLAS*/
```

```
let pessoa: [string, number] = ['João',  
30];
```

# Enum

**Enum** (de enumeração) serve para criar especificamente **valores fixos e nomeados**. É ótimo pra usar quando você tem opções conhecidas, como dias da semana, status ou categorias.



```
enum dias {  
    domingo = 0,  
    segunda = 1,  
    terca = 2,  
    quarta = 3,  
    quinta = 4,  
    sexta = 5,  
    sabado = 6  
}  
  
console.log(dias.domingo); //0  
console.log(dias["domingo"]); //0  
console.log(dias[0]) /*obtendo a chave do  
valor 0 que é: domingo*/
```



# Funções

As funções do TypeScript São como no JavaScript, mas com tipagem estática, o que torna o código mais seguro e fácil de entender.



Obs: No TypeScript, é obrigatório declarar o tipo dos parâmetros, não é possível o Type Inference nesse caso



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

```
console.log(somar(5, 3)); // Saída: 8
```

 **Curtiu o conteúdo? me conte aqui nos comentários o que achou dessas características do TypeScript**

