Tipos Básicos em Typescript

Miguel Costa
miguelangcosta@gmail.com

Agenda

- O que são tipos
- Tipos básicos do TypeScript
- Inferência de tipos
- Tipar funções
- Usar TypeScript com HTML

O que é um tipo?

Ciências ULisboa

- Um tipo define:
  - Que tipo de valor uma variável pode guardar
- Ajuda a:
  - Evitar erros
  - Tornar o código mais claro

➡️ TypeScript verifica os tipos **antes de executar**

Tipos básicos do TypeScript

Ciências ULisboa

Os principais tipos básicos são:

- string → texto
- number → números
- boolean → verdadeiro / falso

Tipo string

Ciências ULisboa

✔️ Apenas texto é permitido

❌ Números ou boolean dão erro

```
let username: string = "Ana";
```

Tipo number

Ciências ULisboa

- Não existe int ou float
- Tudo é number

```
let age: number = 25;
let price: number = 9.99;
```

Tipo boolean

- Valores possíveis:
  - true
  - false

```
let isAdmin: boolean = true;
```

Erro de tipo (exemplo)

Ciências ULisboa

- Erro no editor
- TypeScript avisa antes de executar

```
let age: number = 20;
age = "vinte";
```

Inferência de tipos

Ciências ULisboa

TypeScript muitas vezes **descobre o tipo sozinho**

```
let name = "Ana";         // inferência
let age: number = 30;    // explícito
```

TypeScript sabe que name é string

Não é obrigatório tipar sempre

Quando tipar explicitamente?

Ciências ULisboa

Tipar é importante quando:

- Criamos funções
- Trabalhamos com objetos
- Trabalhamos em equipa

➡️ Código mais claro e seguro

Tipos em funções (parâmetros)

Ciências ULisboa

```
function greet(name: string) {
  console.log("Olá " + name);
}
```

❌ greet(10) → erro

Tipo de retorno da função

Ciências ULisboa

➡️ A função **tem de devolver um number**

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

Função sem retorno (void)

Ciências ULisboa

➡️ A função não devolve nada

```
function showMessage(msg: string): void {
  console.log(msg);
}
```

Ligação com HTML

Ciências ULisboa

TypeScript pode:

- Ler valores do HTML
- Alterar conteúdo da página
- Reagir a eventos

➡️ Como JavaScript, mas com tipos

```
const title = document.querySelector("h1");
```

❗ Pode ser null

➡️ TypeScript avisa

Type Assertion

Ciências ULisboa

```
const input = document.querySelector("#task") as HTMLInputElement;
```

➡️ Estamos a dizer ao TypeScript o tipo correto

Tipo any (o perigo)

Ciências ULisboa

- any aceita qualquer coisa
- Desliga a proteção do TypeScript

❌ Evitar usar sempre que possível

```
let value: any = 10;
value = "texto";
value = true;
```

Tipo any (o perigo)

Ciências ULisboa

- any aceita qualquer coisa = não verificar nada
- Desliga a proteção do TypeScript

❌ Evitar usar sempre que possível

```
let value: any = 10;
value = "texto";
value = true;
```

Por que evitar any?

```
let price: any = "10";
console.log(price * 2);
```

❌ Erro em execução

TypeScript não avisa

Tipo unknown

Ciências ULisboa

- Mais seguro que any
- Obriga a verificar antes de usar

```
let data: unknown;
data = "Olá";
```

unknown com verificação

```
if (typeof data === "string") {
  console.log(data.toUpperCase());
}
```

➡️ TypeScript só deixa usar depois da verificação

Tipo unknown

Ciências ULisboa

- Mais seguro que any
- Obriga a verificar antes de usar

```
let data: unknown;
data = "Olá";
```

unknown com verificação

```
if (typeof data === "string") {
  console.log(data.toUpperCase());
}
```

➡️ TypeScript só deixa usar depois da verificação

null e undefined

Ciências ULisboa

- null: valor vazio intencional

```
let result: string | null = null;
```

- undefined: valor não definido

```
let name: string;
console.log(name); // undefined
```

null e undefined

Ciências ULisboa

```
const title = document.querySelector("h1");
title.innerText = "Olá";
```

❌ Erro possível: title pode ser null

Correção com verificação

```
const title = document.querySelector("h1");

if (title) {
  title.innerText = "Olá";
}
```

Tipos de Elementos HTML no TypeScript

Ciências ULisboa

- Regra simples: **HTML + nome da tag + Element**
- Cada elemento HTML tem um tipo específico
- Isso permite acesso seguro às propriedades corretas

```
document.querySelector("div")      // HTMLDivElement
document.querySelector("input")    // HTMLInputElement
document.querySelector("button")   // HTMLButtonElement
```

Questões