

# Introdução ao Typescript

Miguel Costa  
[miguelangcosta@gmail.com](mailto:miguelangcosta@gmail.com)

# O que é JavaScript?

JavaScript é uma linguagem:

- Interpretada
- Dinâmica
- Sem tipos obrigatórios
- Muito usada no Front-End e Back-End

```
let age = 10;  
age = "dez";
```

Isto não dá erro em JavaScript!

# Problema do Javascript

- Erros só aparecem **em tempo de execução**
- Código difícil de manter em projetos grandes
- Bugs comuns:
  - a. Tipos errados
  - b. Funções mal usadas
  - c. Objetos incompletos

✗ Erros descobertos tarde  
✗ Código menos seguro

# Bugs Comuns do Javascript

Tipos errados:

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

- Resultado inesperado
- JavaScript permite misturar tipos
- Erro só aparece em execução

# Bugs Comuns do Javascript

Funções mal usadas:

```
function calculateTotal(price, quantity) {  
    return price * quantity;  
}  
  
calculateTotal(10, "5");
```

- "5" é uma string, não um número
- Pode gerar resultado errado
- Nenhum erro antes de executar
- JavaScript **não valida tipos**

# Bugs Comuns do Javascript

Objetos Incompletos:

```
function printUser(user) {
    console.log(user.name.toUpperCase());
}

const user = {
    age: 25
};

printUser(user);
```

- `user.name` é `undefined`
- Erro em tempo de execução

# O que é o Typescript

- TypeScript é:
  - Um **superset** de JavaScript
  - Criado pela **Microsoft**
- Adiciona:
  - Tipos (`string, number, boolean, etc...`)
  - Interfaces
  - Classes melhoradas
  - Validações, menos erros
  - Melhor organização, código mais seguro e profissional

📌 Todo JavaScript é TypeScript

📌 Nem todo TypeScript é JavaScript

# Exemplo JavaScript vs Typescript

JavaScript:

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

TypeScript:

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

- TypeScript NÃO impede erros de tipo em execução.
- Ele deteta erros de tipo ANTES de executar o código.

# O que são tipos?

Tipos dizem:

- Que tipo de valor uma variável pode guardar

Exemplos:

- `string` → texto
- `number` → números
- `boolean` → true / false

📌 Evita usar valores errados

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

- Erro no editor
- Erro encontrado antes de executar

# Onde o TypeScript é usado?

TypeScript é usado em:

- React
- Angular
- Vue
- Node.js
- Grandes empresas:
  - Google
  - Microsoft
  - Netflix

→ Muito usado no mercado de trabalho

# TypeScript no Front-End

TypeScript:

- Não roda diretamente no browser
- Precisa ser convertido para JavaScript

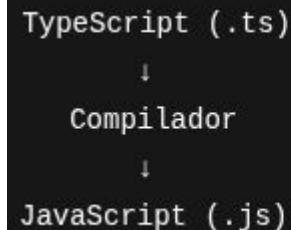
 **Browsers não entendem TypeScript**

# Como TypeScript funciona?

Fluxo:

1. Escrevemos código em `.ts`
2. TypeScript compila para `.js`
3. O browser executa o `.js`

→ TypeScript é uma **ferramenta de desenvolvimento**



# O que é o TypeScript Compiler (tsc)?

- Ferramenta oficial do TypeScript
- Converte `.ts` → `.js`
- Verifica erros de tipo
- Comando bash: `tsc`

# O que precisamos para usar TypeScript?

- Node.js instalado
- TypeScript instalado
- Editor de código (VS Code)

📌 VS Code é o mais usado com TypeScript

# Instalar o TypeScript

## 1. Verificar se o Node.js está instalado:

```
node -v
```

- Permite correr JavaScript fora do browser
- Inclui o **npm** (gestor de pacotes)

```
npm -v
```

## 2. Instalar o Typescript (global):

```
npm install -g typescript
```

- O comando **tsc** ficará disponível em qualquer pasta

```
tsc -v
```

# Desvantagens do TypeScript

- Mais código para escrever
- Precisa de compilação
- Curva de aprendizagem

# TypeScript + HTML + CSS

Neste curso vamos:

- Usar TypeScript com:
  - HTML
  - CSS
- Criar aplicações reais
- Evoluir um projeto ao longo do mês

# Projeto do curso

## Gestor de Tarefas

- Criar tarefas
- Marcar como concluidas
- Filtrar
- Guardar dados



Projeto cresce aula a aula

# Questões