



UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ

# Fundamentos de TypeScript

QXD0020 - Desenvolvimento de Software para Web

Prof. Bruno Góis Mateus ([brunomateus@ufc.br](mailto:brunomateus@ufc.br))

# Agenda

- Introdução
- Configurando Ambiente
- Sistemas de Tipo e Inferência de Tipo
- Orientação a objetos

# Introdução



# Introdução

## TypeScript

- É uma linguagem **fortemente tipada** construída em cima do JavaScript
  - Criada e mantida pela **Microsoft**
- Superconjunto do **JavaScript (*typed*)**
  - Qualquer programa em JS é um programa válido em TS
- Projetada para ser usada no Front-end e no Back-end
  - **Compilada** / Traduzida para JS

# Introdução

## Características

- TypeScript é um superset do ECMAScript 2015, logo ela herda as seguintes características:
  - Classes
  - Módulos
  - Funções Arrow
  - Parâmetros opcionais e com valores padrão

# Introdução

## Características

- Tem como principal característica o **sistema de tipos**
  - É possível determinar os tipos de variáveis e parâmetros
  - Por meio de checagem estática erros de tipo são capturados mais cedo
  - Em várias ocasiões, a declaração do tipo é desnecessária devido ao mecanismo de inferência de tipos

# Introdução

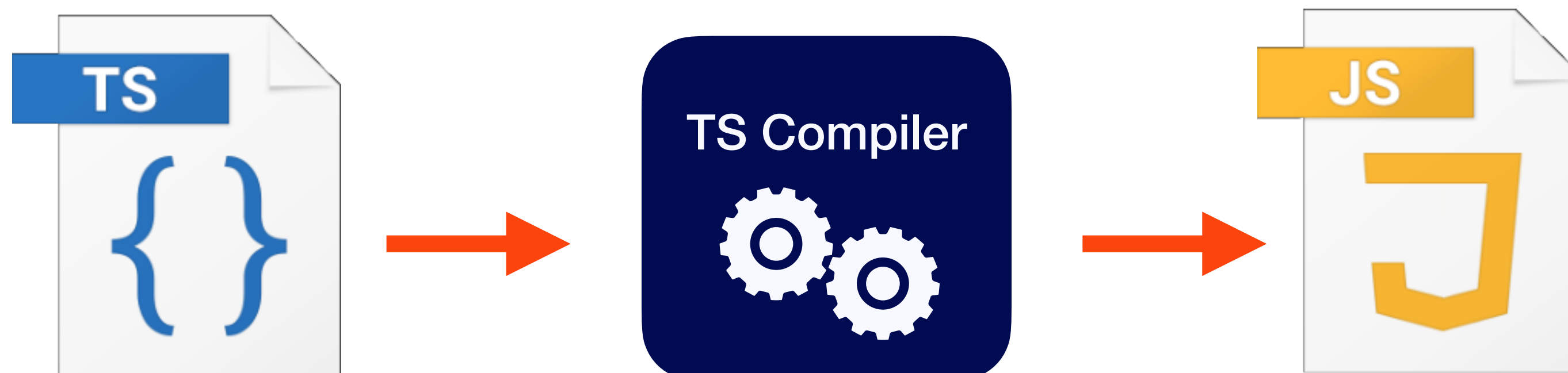
## Características

- Outras features de código introduzidas pelo TS são:
  - Interfaces
  - Classes Abstratas
  - Modificadores de visibilidade
  - Generics
  - Optionals
  - Function Overloading
  - Decorators e mais

# Introdução

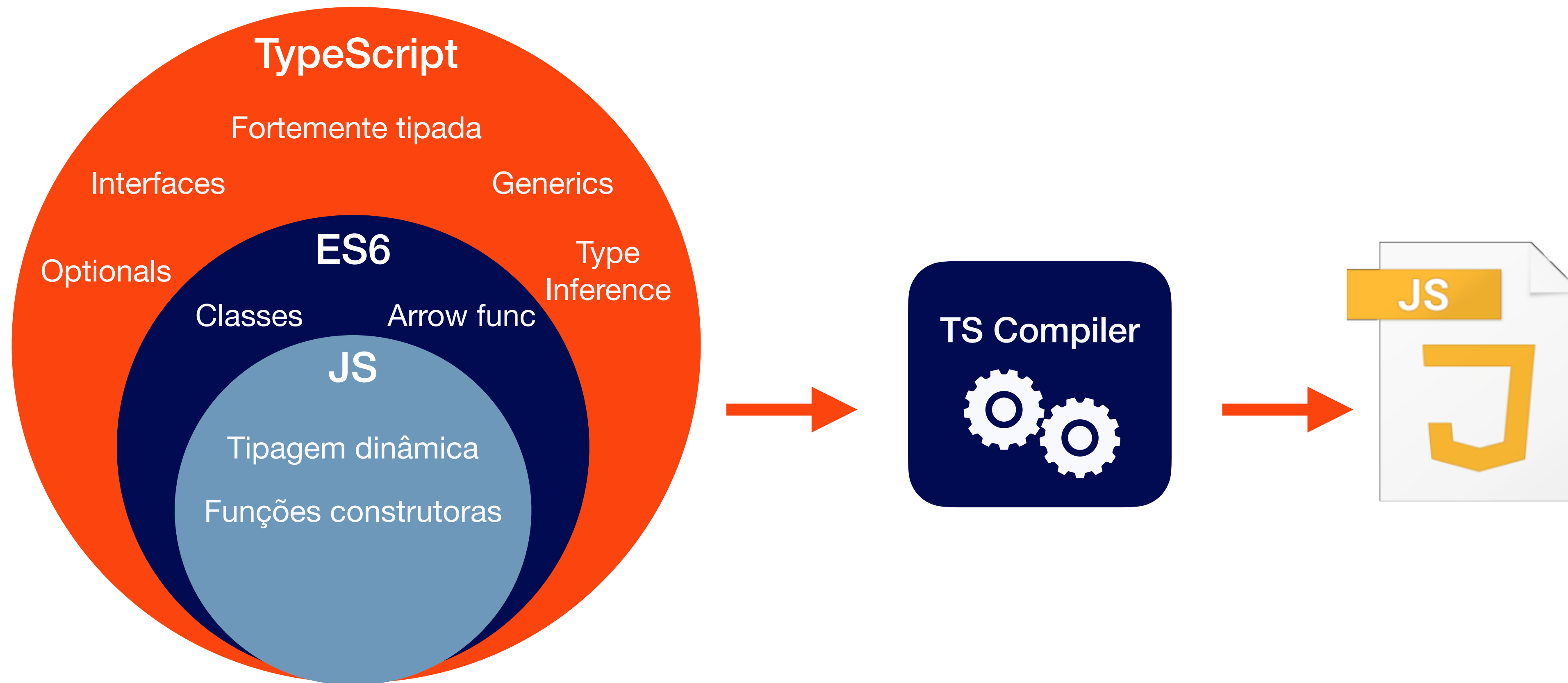
## Navegadores entendem TypeScript ?

- É necessário converter TS para JS
- Este processo é feito por um compilador ou um transpilador
- TS pode ser executado em qualquer ambiente que suporte JS





# Introdução



# Introdução

## Vantagens

- Checagem de tipos estática (opcional)
  - O compilador alerta sobre erros relacionados a tipagem
- Detecção de bugs mais cedo
  - O compilador alerta sobre erros que seriam apenas descobertos durante a execução
- Maior legibilidade
- Otimizações de código
  - Type annotation, Generics, API Documentation, Intellisense

# Introdução

## Diferenças

### JavaScript

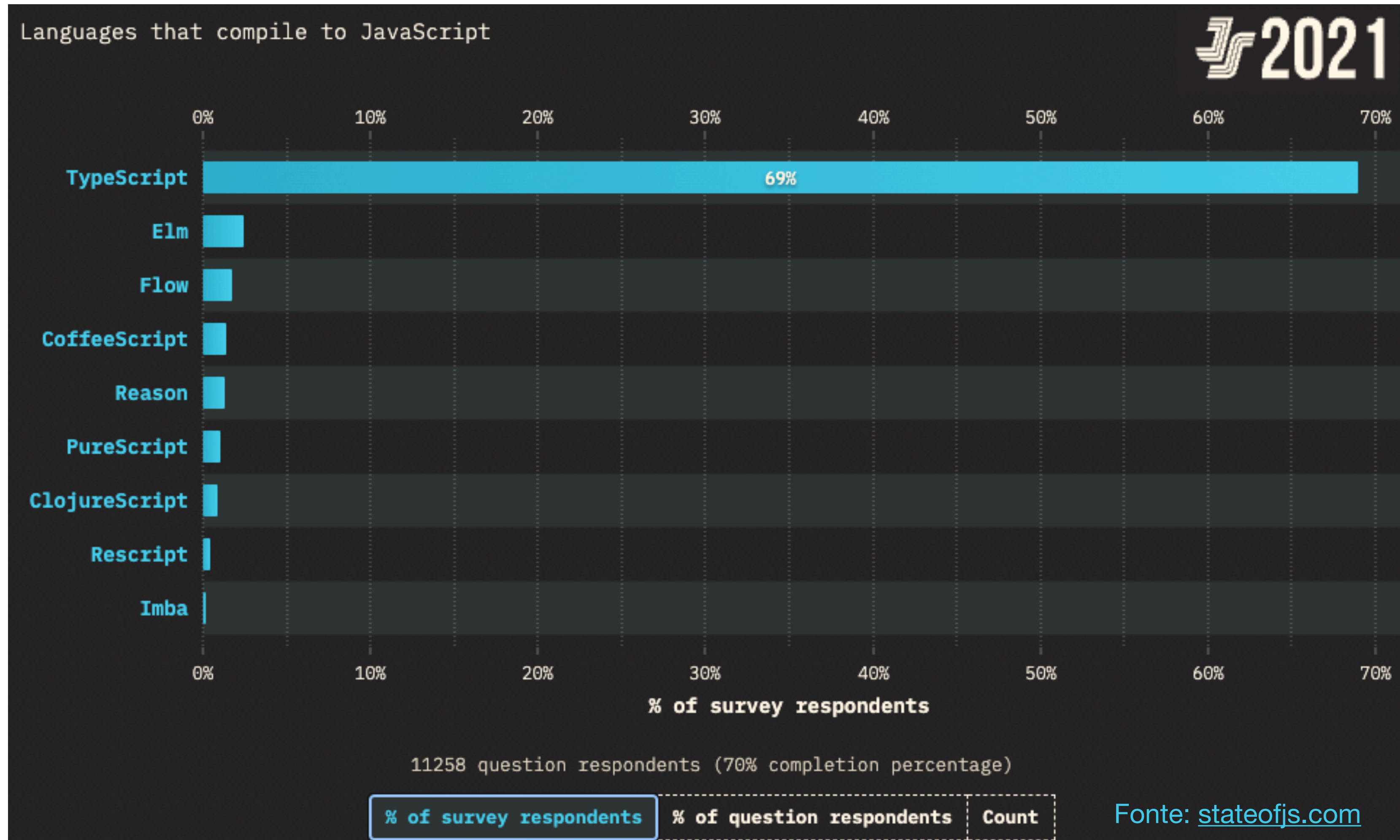
- Criada por Brendan Eich (Netscape) e lançada em 1995
- Linguagem de script leve com foco em ajudar o desenvolvimento de páginas web dinâmicas
- Fracamente tipada e c/ suporte somente a tipagem dinâmica
- Mais adequada para pequenos projetos
- Executada diretamente nos navegadores
- Bibliotecas JS funcionam por padrão

### TypeScript

- Criada pela Microsoft e lançada em 2012
- Superset de JS criada para lidar com a complexidade de código em grandes projetos OO
- Fortemente tipada c/ suporte a tipagem estática e dinâmica
- Mais adequada para aplicações maiores
- Convertida em JS capaz de ser executado por navegadores
- Bibliotecas JS funcionam por padrão

# Introdução

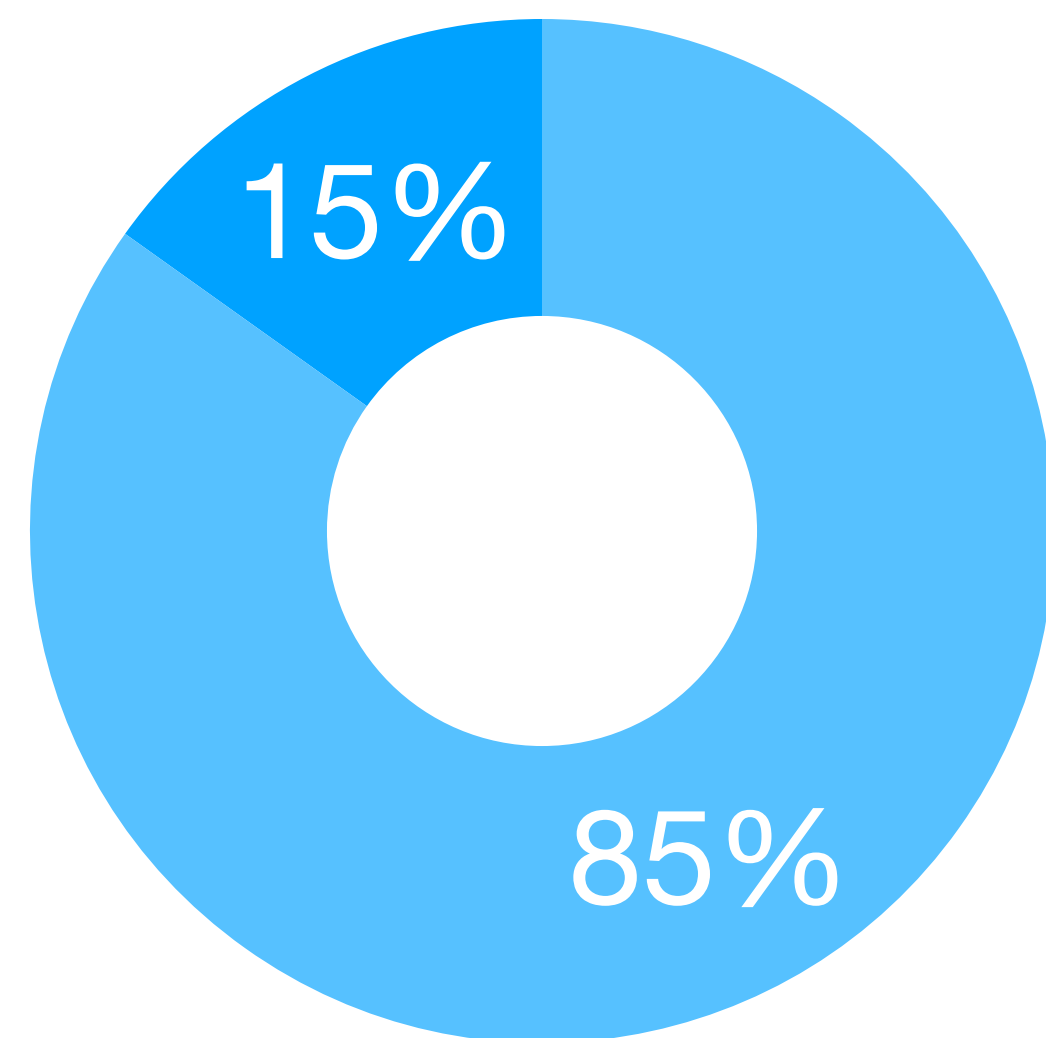
## Comunidade JS



# Introdução

## Comunidade JS

Você usou TypeScript no último ano?



● Sim

● Não

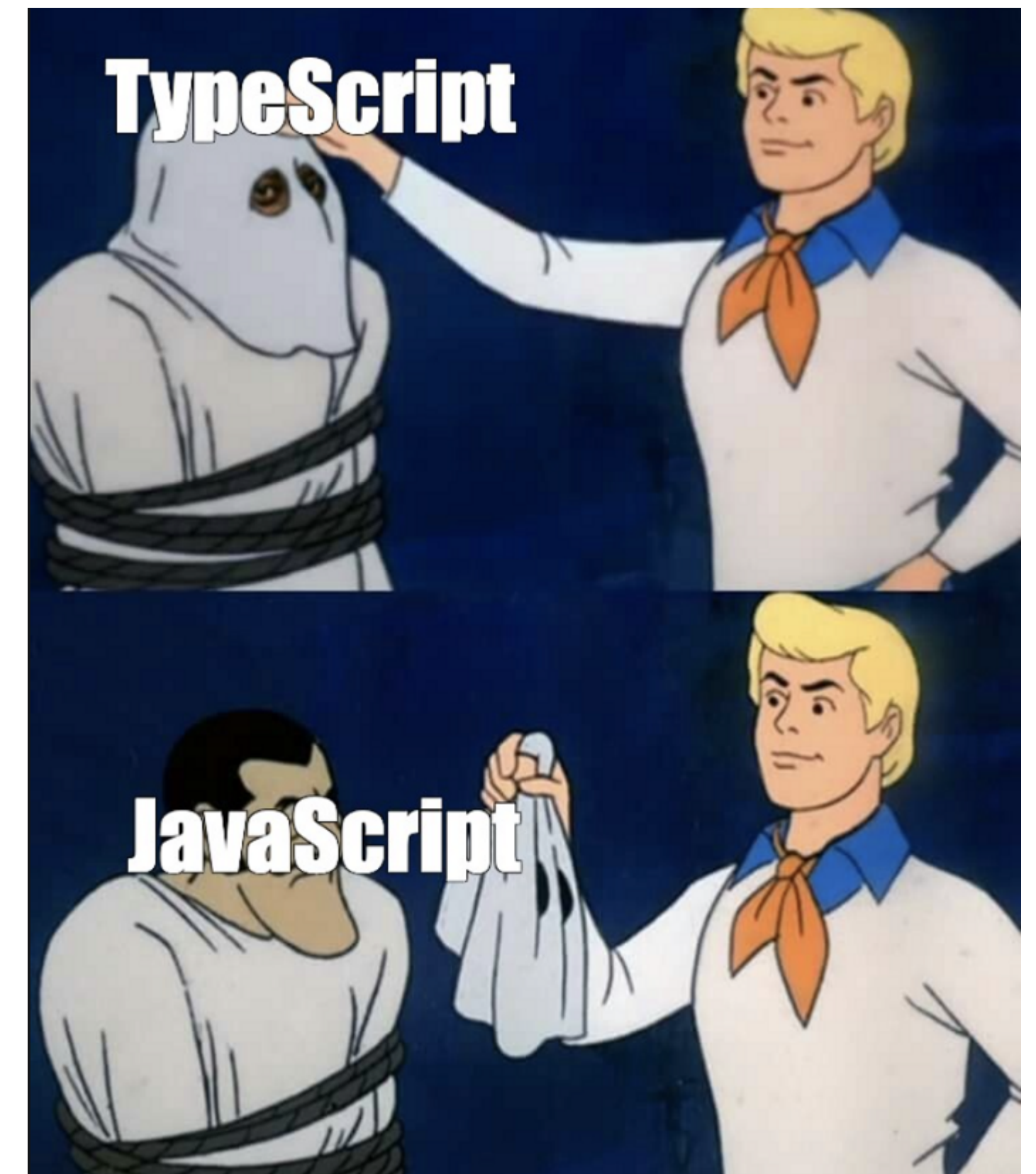
Company type	Not using TypeScript	Using TypeScript
Not provided	32.67%	67.33%
Government organization	19.35%	80.65%
Non-tech-first company	19.21%	80.79%
Software development company / developer agency	12.93%	87.07%
Tech-first / digital-first company	12.16%	87.84%

Fonte: [State of Frontend](#)

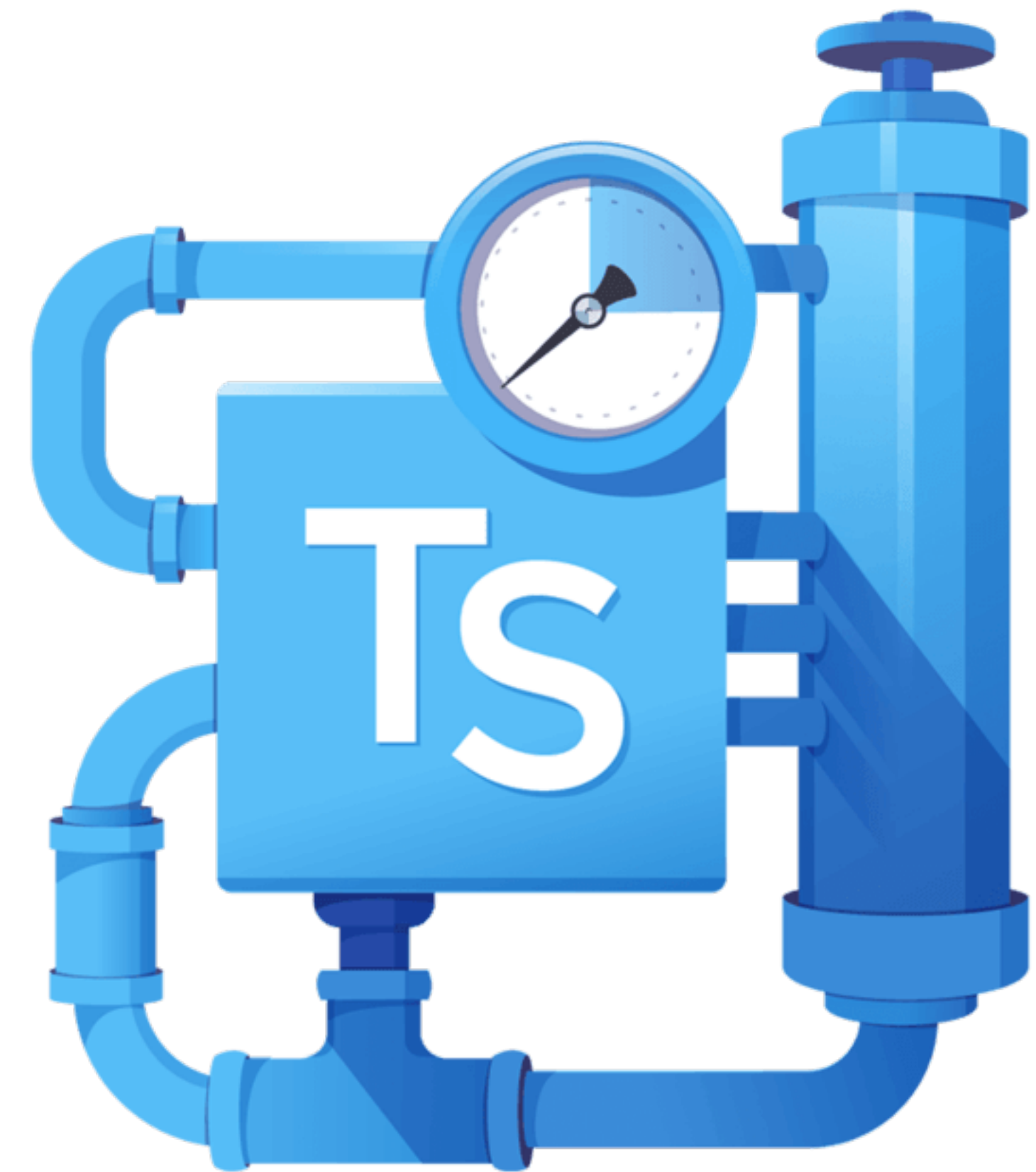
# Introdução



You can't get a run time error  
if your code doesn't even  
compile to begin with



Configurando o ambiente



# Configurando o ambiente

## Passa a passo

### 1. Instalar o Node

- Windows: <https://nodejs.org/en/download/>
- Linux: `$ sudo apt-get install node`
- Mac: `$ brew install node`
- Node Version Manager

### 2. Instalar o gerenciador de pacotes do Node

- Linux: `$ sudo apt-get install npm`

### 3. Instalar o compilador do TS

- `$ npm install -g typescript`



# Configurando o ambiente

## Compilando e executando

1. Compilando um arquivo .ts
  - `$ tsc meu_arquivo.ts`
2. Executar .js gerado com o node
  - `$ node meu_arquivo.js`

# Configurando o ambiente

**IDE recomendada**



Visual Studio Code

# Configurando o ambiente

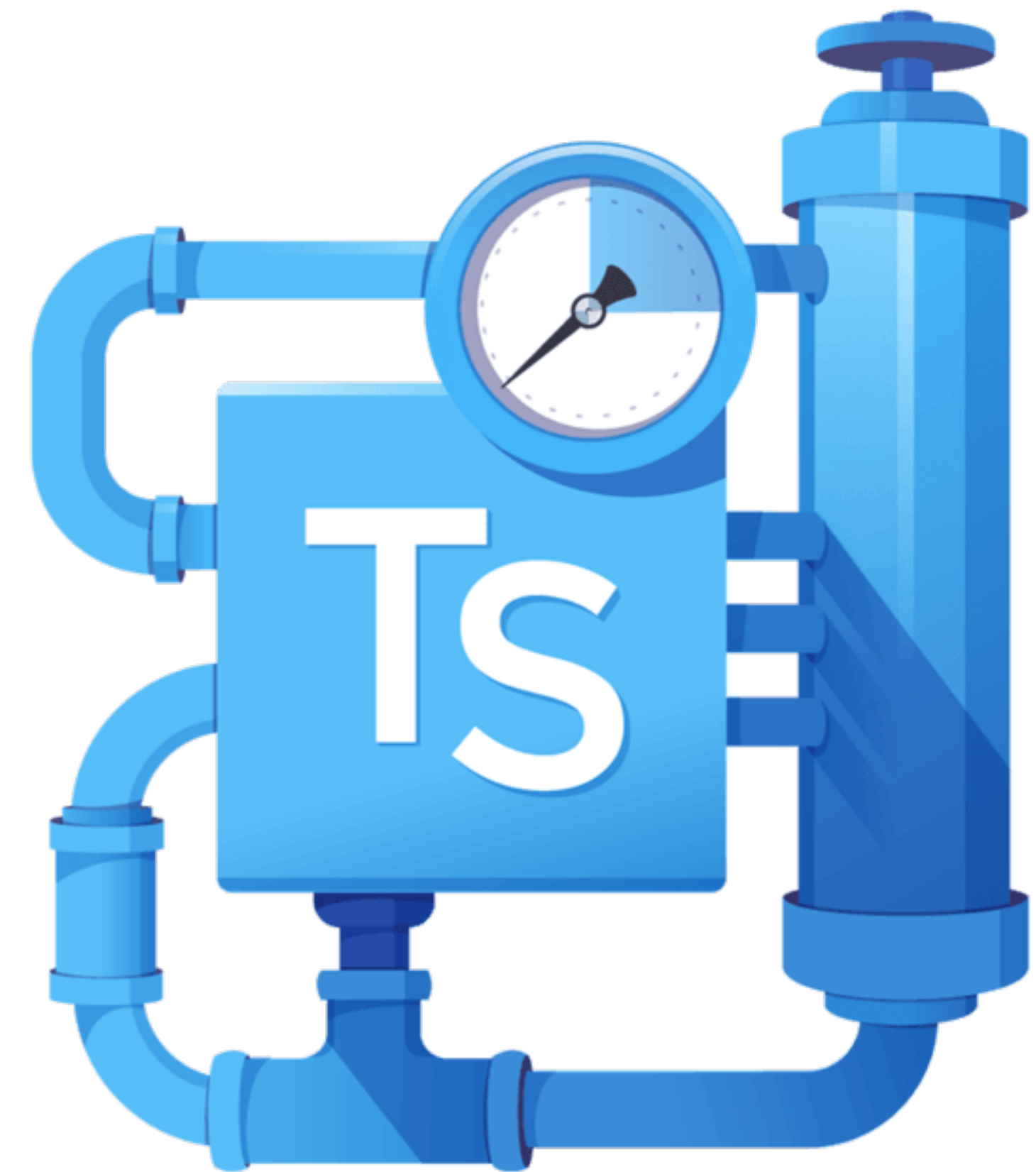
## Configurando o TypeScript

### tsconfig.json

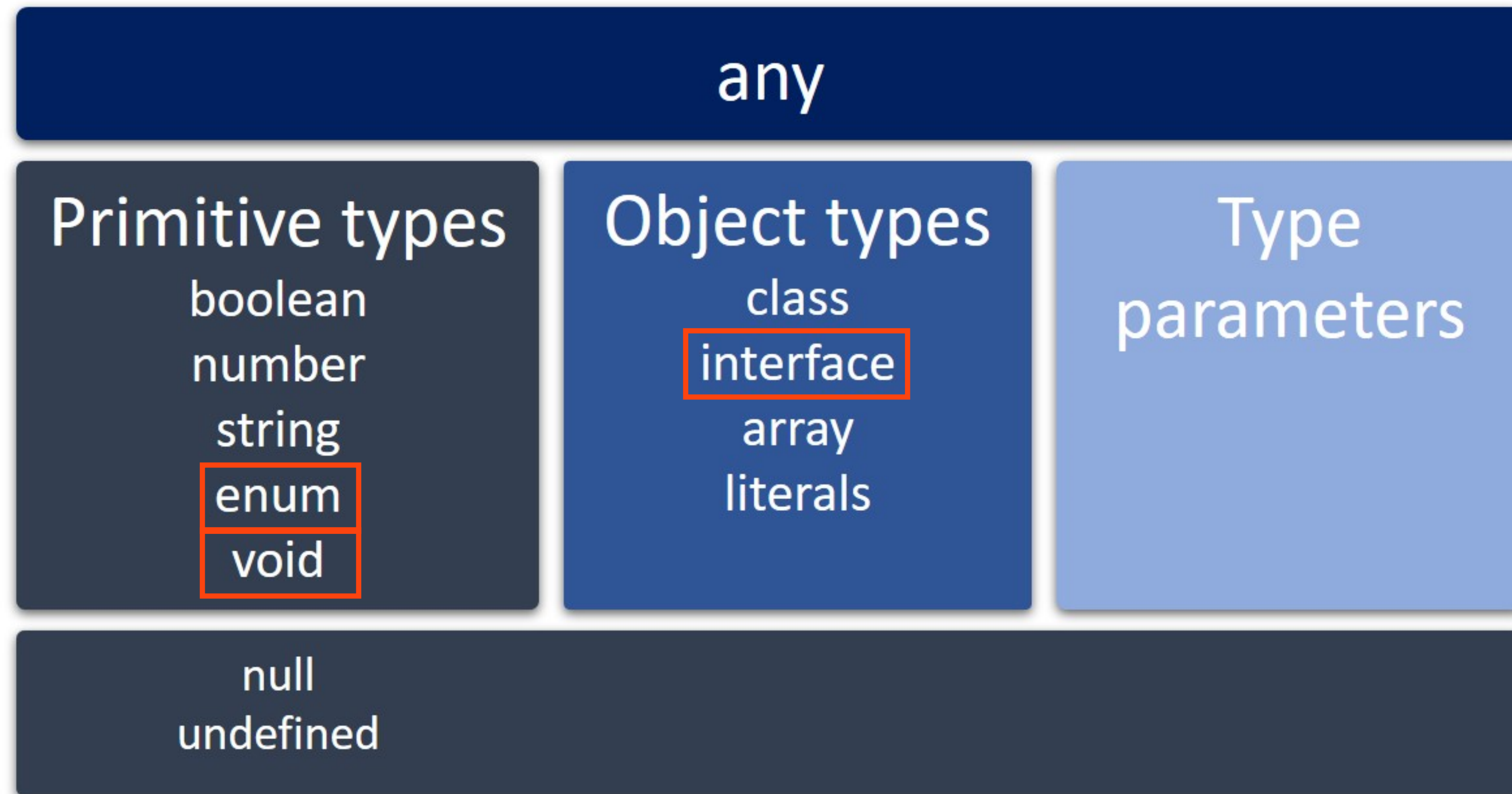
- Define a raiz do projeto e as opções de compilação requeridas pelo projeto
  - AlwaysStrict
  - TypeChecking
  - JavaScriptSupport
  - StrictNullChecks
  - Lib

Referência completa:  
<https://www.typescriptlang.org/tsconfig>

# Sistema de Tipo e Inferência de Tipo



# Sistema de Tipo e Inferência de Tipo



# Sistema de Tipo e Inferência de Tipo

## Declaração de variáveis

- É possível definir o tipo de cada variável
- Ao tentar atribuir um valor do tipo errado, o compilador acusa o erro

```
let flag: boolean;  
let x: number;  
let big: bigint = 100n;  
let firstName: string = "Mateo";  
  
x = "10"; //Erro!
```

# Sistema de Tipo e Inferência de Tipo

## Union

- Descrevem valores que podem assumir mais de um tipo
- Restringe a atribuição aos tipos especificados
- Intellisense

```
let multiType: number | boolean;  
multiType = 20;           /* Valid  
multiType = true;        /* Valid  
multiType = "twenty";    /* Invalid
```

# Sistema de Tipo e Inferência de Tipo

## Union

- Tipos literais

```
type testResult = "pass" | "fail" | "incomplete";
let myResult: testResult;
myResult = "incomplete";    /* Valid
myResult = "pass";         /* Valid
myResult = "failure";

type dice = 1 | 2 | 3 | 4 | 5 | 6;
let diceRoll: dice;
diceRoll = 1;              /* Valid
diceRoll = 2;              /* Valid
diceRoll = 7;              /* Invalid
```



# Sistema de Tipo e Inferência de Tipo

## Type Guards

Type	Predicate
string	<code>typeof s === "string"</code>
number	<code>typeof n === "number"</code>
boolean	<code>typeof b === "boolean"</code>
undefined	<code>typeof undefined === "undefined"</code>
function	<code>typeof f === "function"</code>
array	<code>Array.isArray(a)</code>

# Sistema de Tipo e Inferência de Tipo

## Enum

- Fornecem uma maneira fácil de trabalhar com um conjunto restrito de valores
- Um enum, é um nome simbólico para um conjunto de valores
  - São tratados como um tipo de dado
- Quando uma função ou variável aceita um número limitado de valores, considere utilizar enum

```
enum Direcoes {  
    Cima,  
    Baixo,  
    Esquerda,  
    Direita  
}
```

# Sistema de Tipo e Inferência de Tipo

## Enum

- Vantagens:
  - Ajuda na redução de erros relacionado
  - Facilita mudanças futuras
  - Aumenta a clareza e a legibilidade do código quando utilizadas com sabedoria

# Sistema de Tipo e Inferência de Tipo

## Tuplas

- São tipos especiais de Array
- Permitem a criação de Arrays com tipos mistos
- Dimensão e tipos restritos

```
let person1: [string, number] = ['Marcia', 35];  
let person2: [string, number] = ['Marcia', 35, true]; // Erro  
let person3: [string, number] = [35, 'Marcia']; // Erro
```

# Sistema de Tipo e Inferência de Tipo

## Funções

```
let addNumbers = function (x: number, y: number): number {  
    return x + y;  
}
```

```
let addNumbers1 = (x: number, y: number): number => x + y;
```

```
function addNumbers2 (x: number, y?: number): number {  
    return x + y;  
}
```

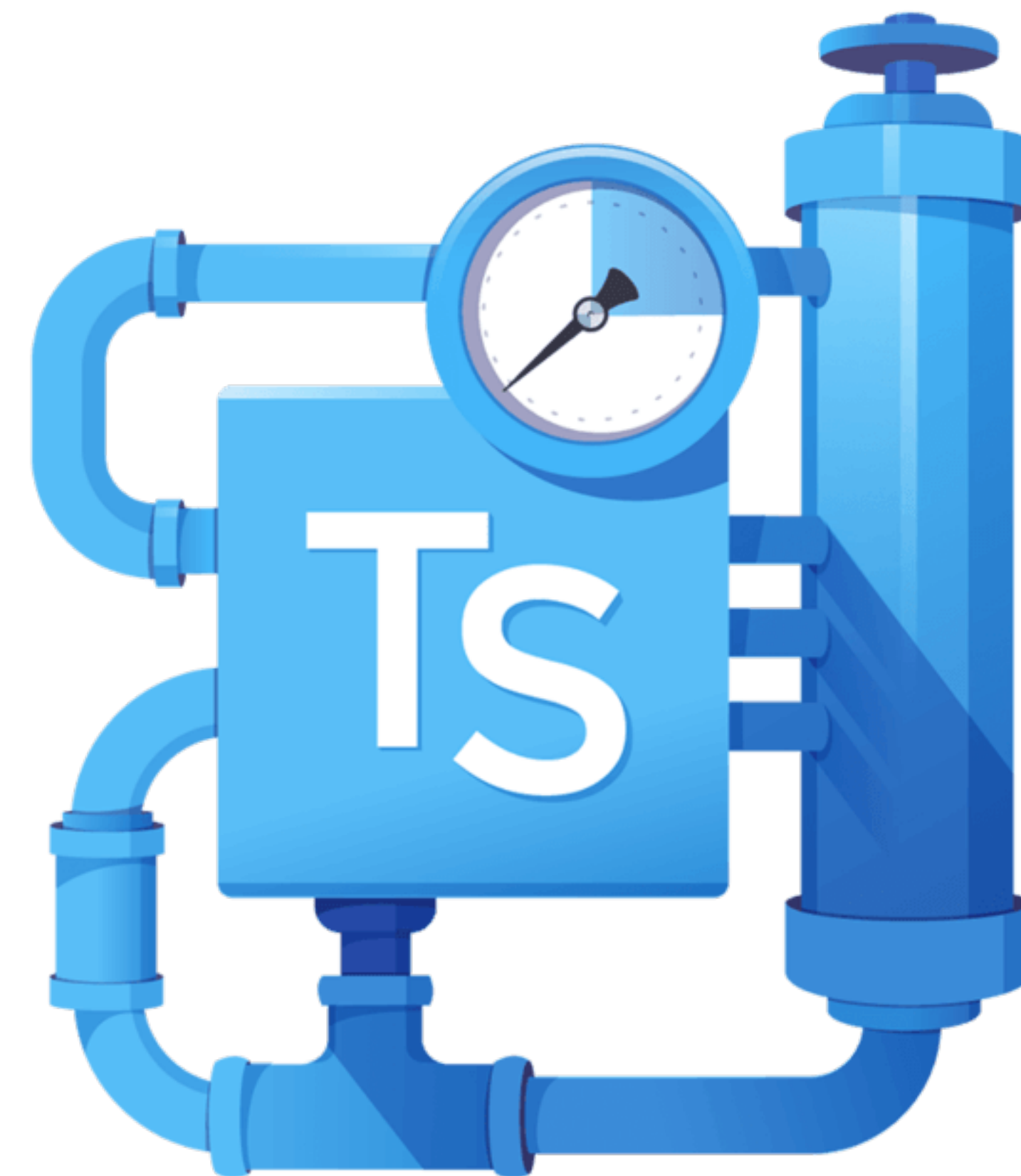
```
function addNumbers3 (x: number, y = 25): number {  
    return x + y;  
}
```

# Sistema de Tipo e Inferência de Tipo

## Funções

```
interface Message {  
  text: string;  
  sender: string;  
}  
  
function displayMessage({text, sender}: Message) {  
  console.log(`Message from ${sender}: ${text}`);  
}
```

# Orientação a Objetos



# Orientação a Objetos

## Classes

- Classes em TS possuem modificadores de acesso
  - public (default), protected, private
- É possível definir métodos e atributos de classe
- Podemos definir classes abstratas



# Orientação a Objetos

## Interfaces

- São construções de *design-time*
- São removidas durante a transpilação para JS
- É possível usá-las sem a necessidade de envolvermos classes
  - Definem uma estrutura de dados

```
interface Sundae extends IceCream {  
    sauce: 'chocolate' | 'caramel' | 'strawberry';  
    nuts?: boolean;  
    whippedCream?: boolean;  
    instructions?: boolean;  
}
```

# Referências

- <https://www.typescriptlang.org>
- [TypeScript vs. JavaScript - Avelon Pang](#)
- [Typescript vs Javascript - Difference you should know](#)
- <https://2020.stateofjs.com/en-US/>
- [Mini Curso gratuito de TypeScript - Willian Justen](#)
- [Build JavaScript applications using TypeScript](#)

Por hoje é só