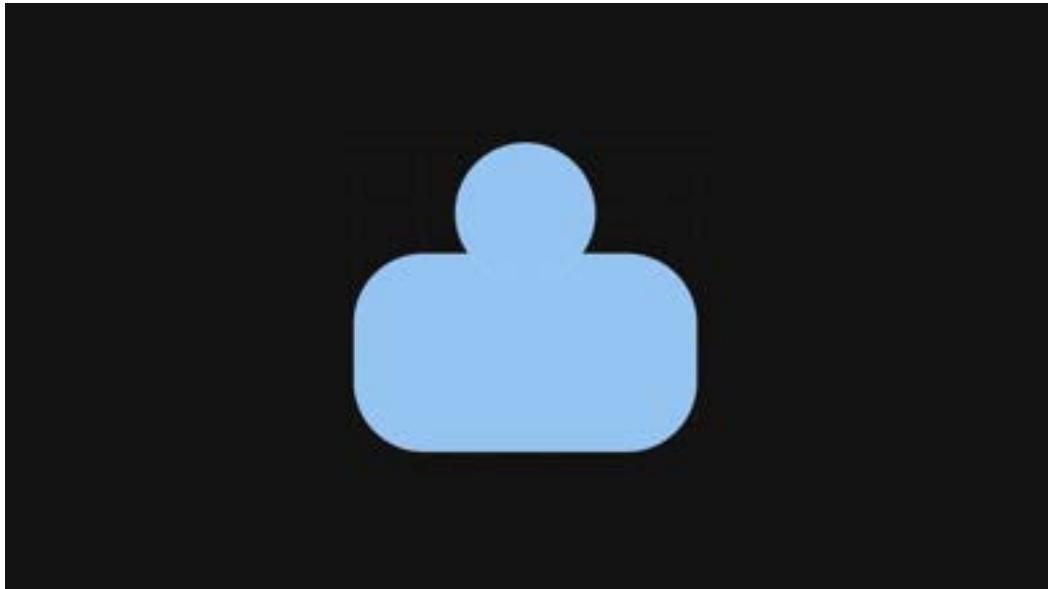




ARGUS ACADEMY

▶ TypeScript



ARGUS ACADEMY

www.argus-academy.com

Sobre a Argus Academy

A Argus Academy é uma plataforma completa de ensino digital que oferece cursos de alta qualidade em diversas áreas de TI, com ênfase especial em programação.

Nossa missão é democratizar o acesso à educação de qualidade, capacitando você para alcançar seus objetivos profissionais e pessoais.

Com que você irá aprender

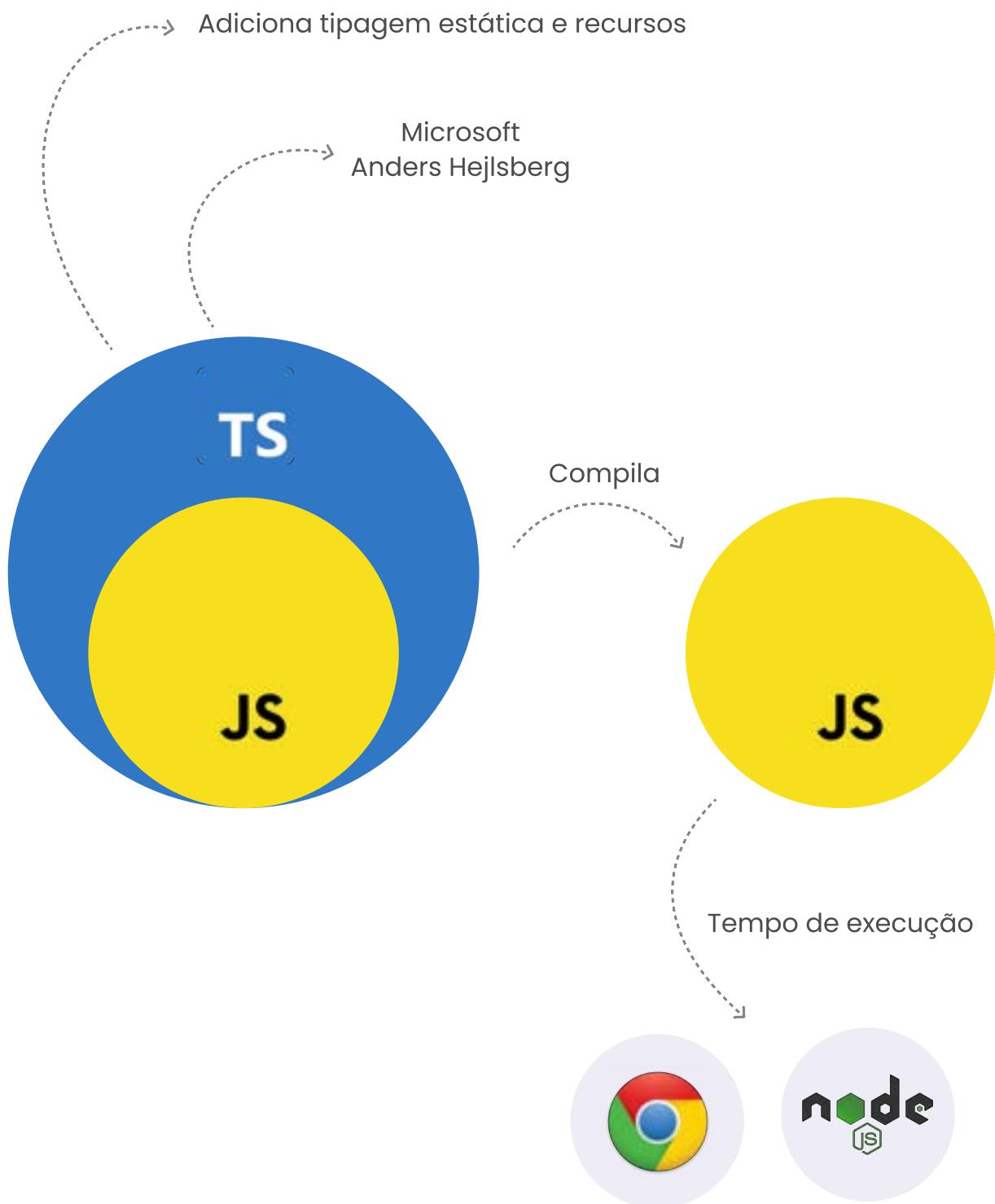


Jorge Sant'Ana é programador, instrutor, entusiasta de tecnologia e fundador da Argus Academy e Argus Technology.

Formado em Tecnologia para Gestão de Negócios pela Fatec e Pós Graduado em Gestão de Projetos de TI pela Fundação Vanzolini.

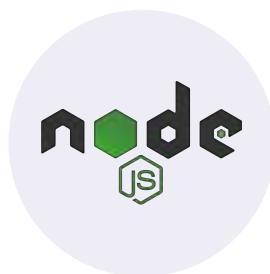
► Introdução

O que é TypeScript?



- ▶ Introdução

Introdução ao NodeJS, NPM e VS Code



Plataforma de execução de
código JavaScript



Gerenciador de pacotes do
NodeJS



Editor de código fonte

- ▶ Introdução

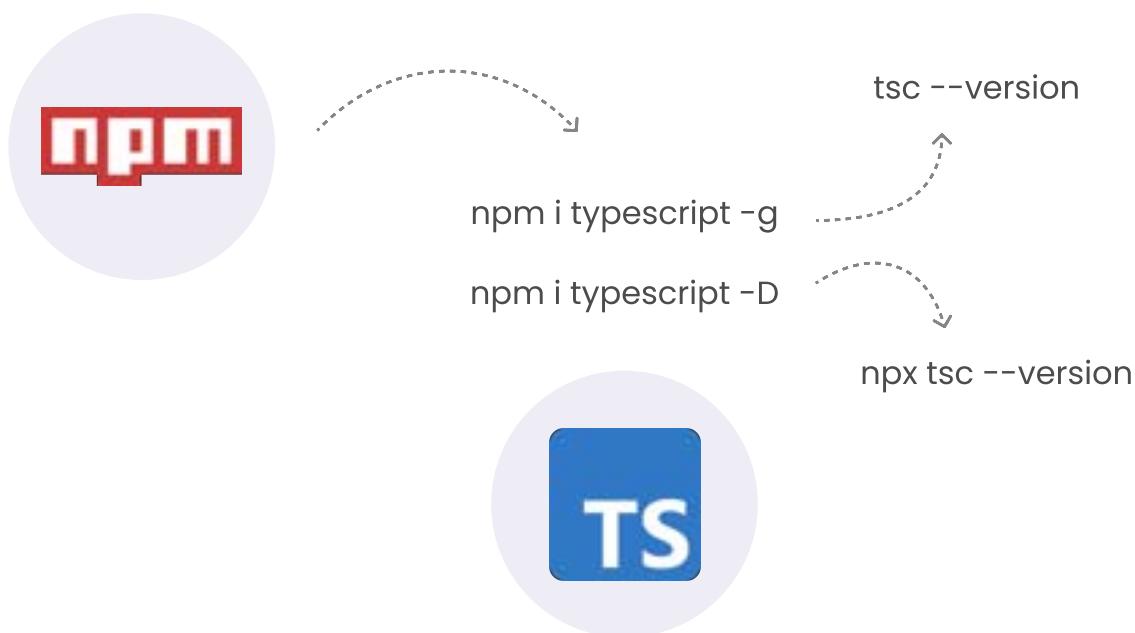
Iniciando o projeto com o NPM



npm init -y

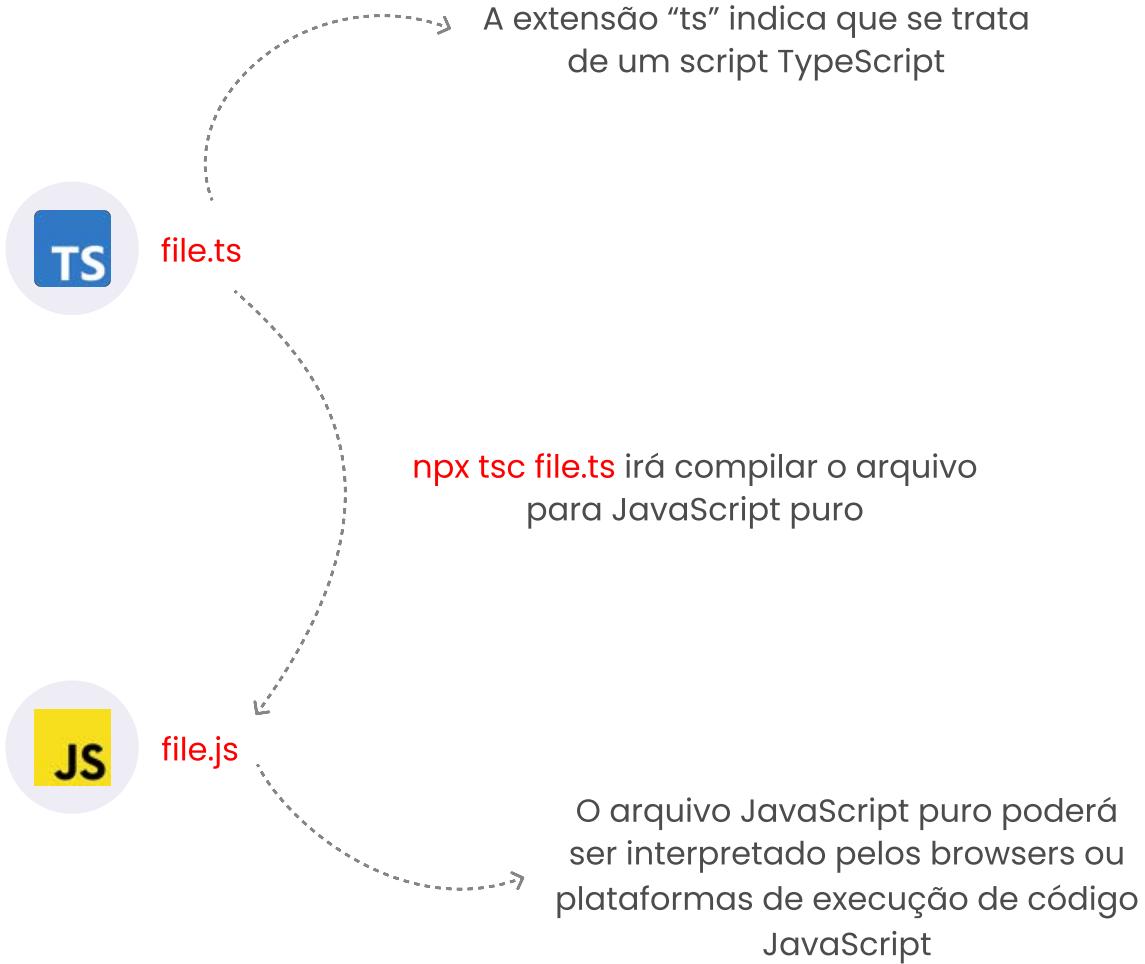
- ▶ Introdução

Instalando o TypeScript no projeto



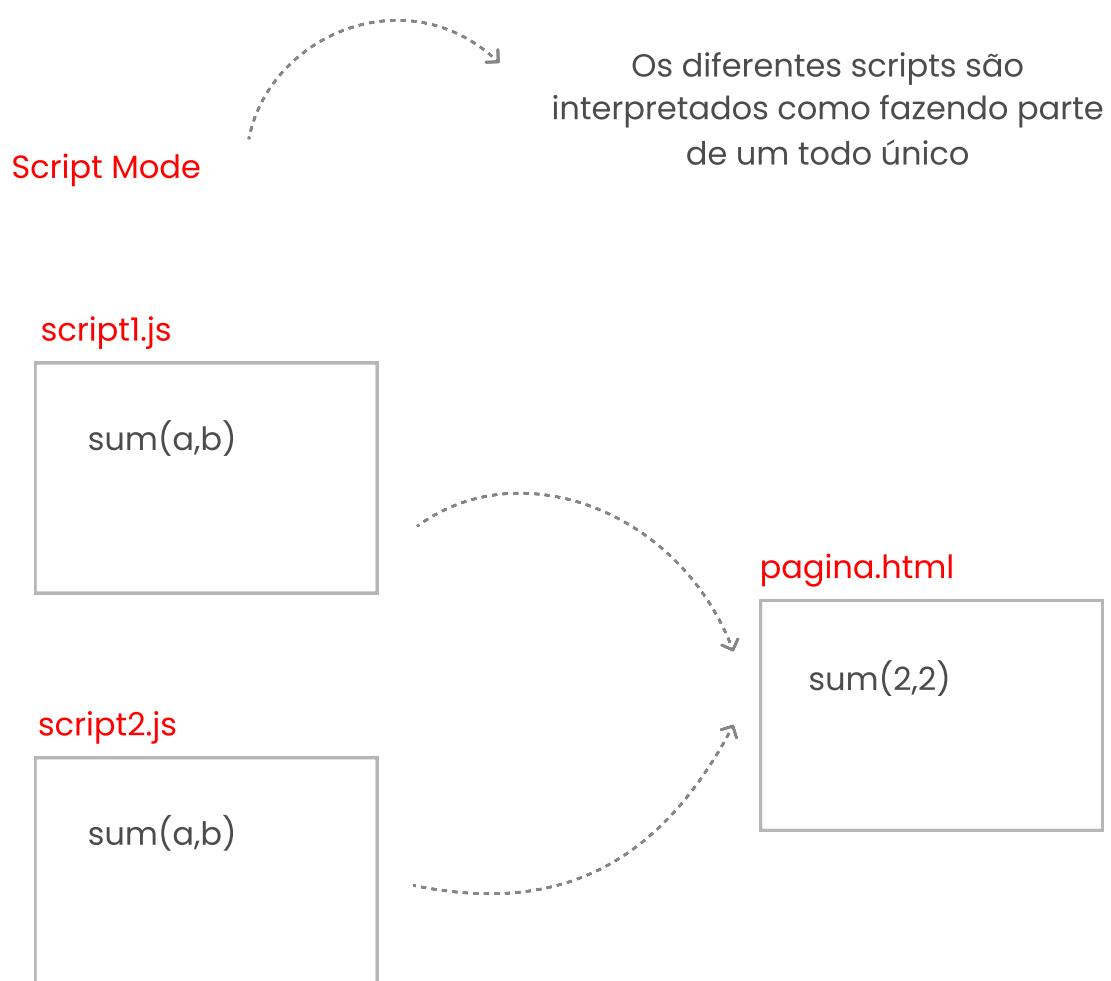
- ▶ Introdução

Compilando TypeScript para JavaScript



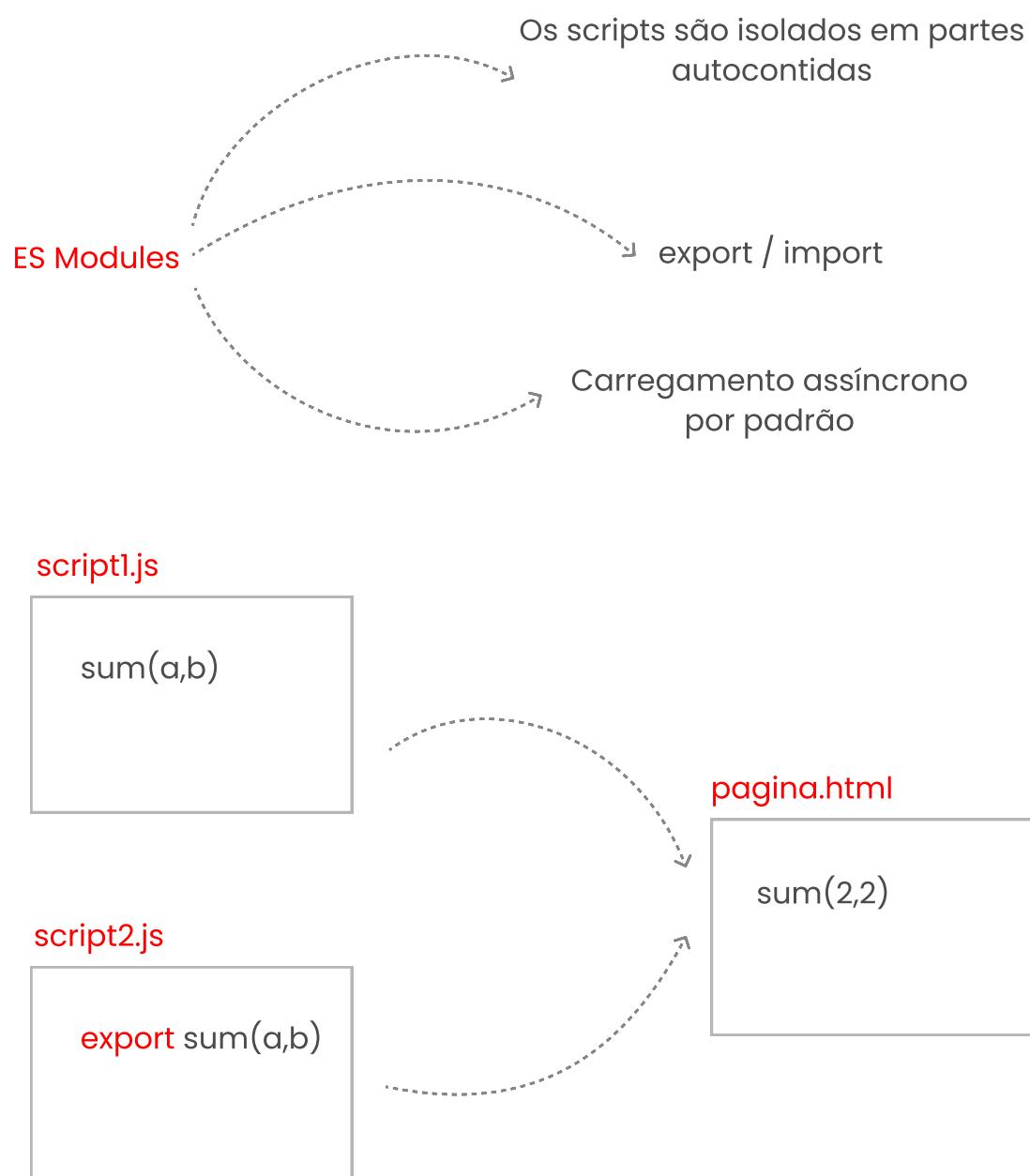
- ▶ Introdução

Script Mode



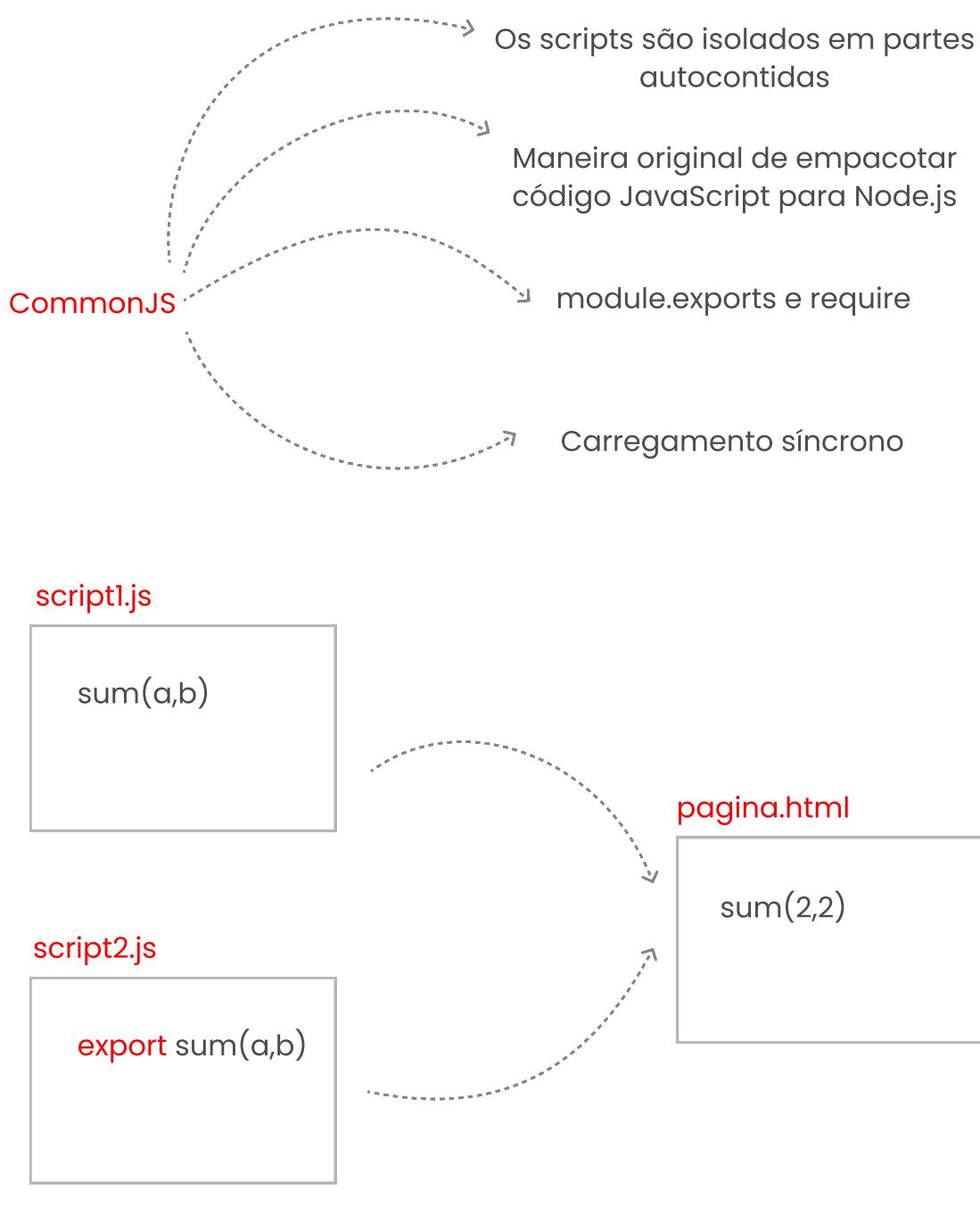
- ▶ Introdução

ES Modules



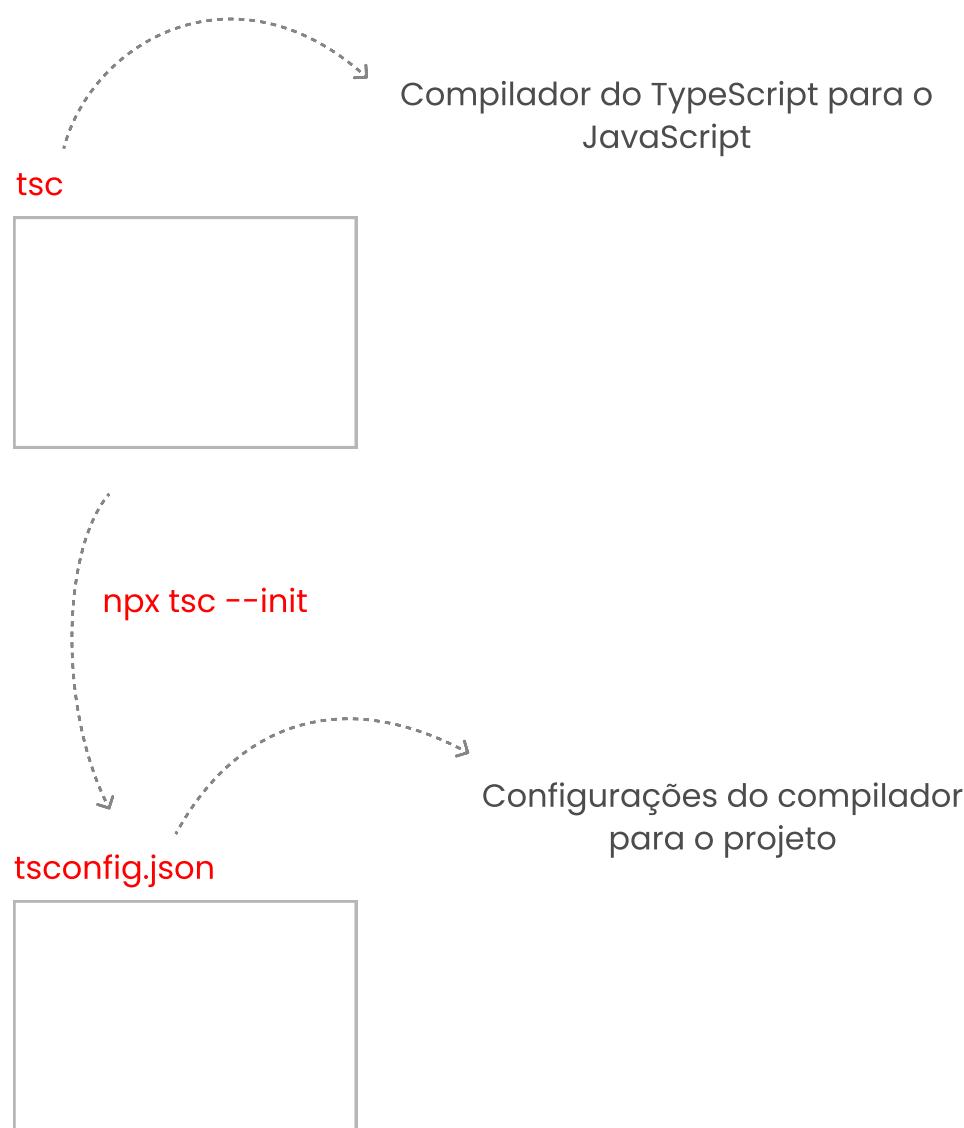
- ▶ Introdução

CommonJS



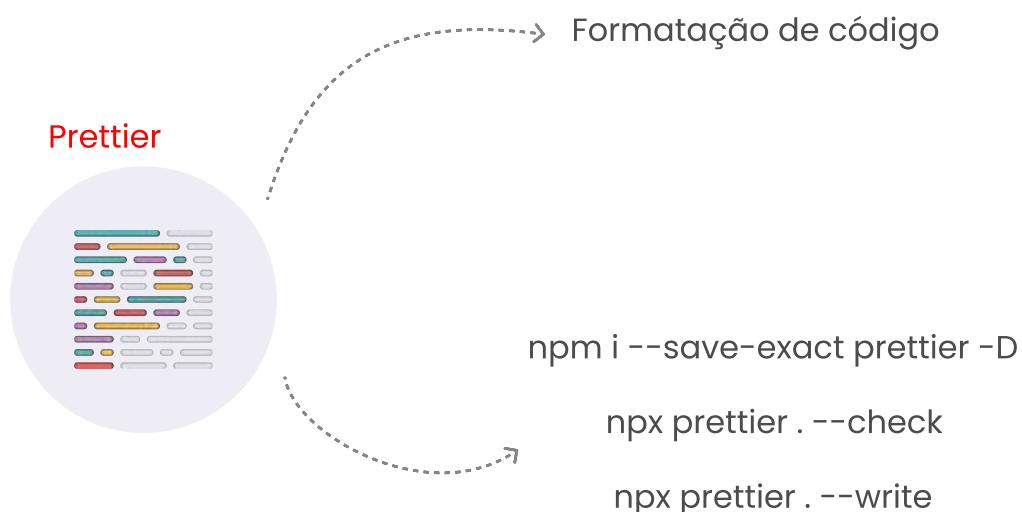
- ▶ Introdução

Iniciando o TypeScript no projeto (introdução ao tsconfig.json)



- ▶ Introdução

Prettier - Introdução, instalação e uso



- ▶ Introdução

VS Code – Format on Save e End of Line



- ▶ Introdução

Versionando o projeto com o Git



- ▶ TypeScript - Tipos Básicos

Type Inference

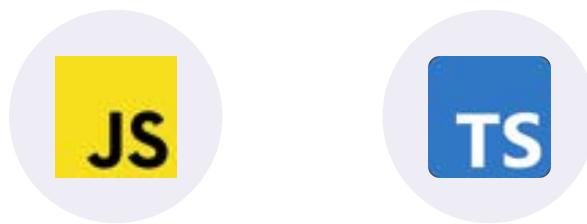
Type Inference



Deduz automaticamente o tipo
da variável

- ▶ TypeScript - Tipos Básicos

Type Annotation



Number	number
String	string
BigInt	bigint
Boolean	boolean
Symbol	symbol
Null	null
Undefined	undefined
Object	object

```
export function display(product: string, price: number) {  
    console.log(product, price);  
}
```

Utilize a anotação de tipo apenas quando a inferência de tipo não for possível

- ▶ TypeScript - Tipos Básicos

Tipo Array



number

string

bigint

boolean

symbol

null

undefined

object

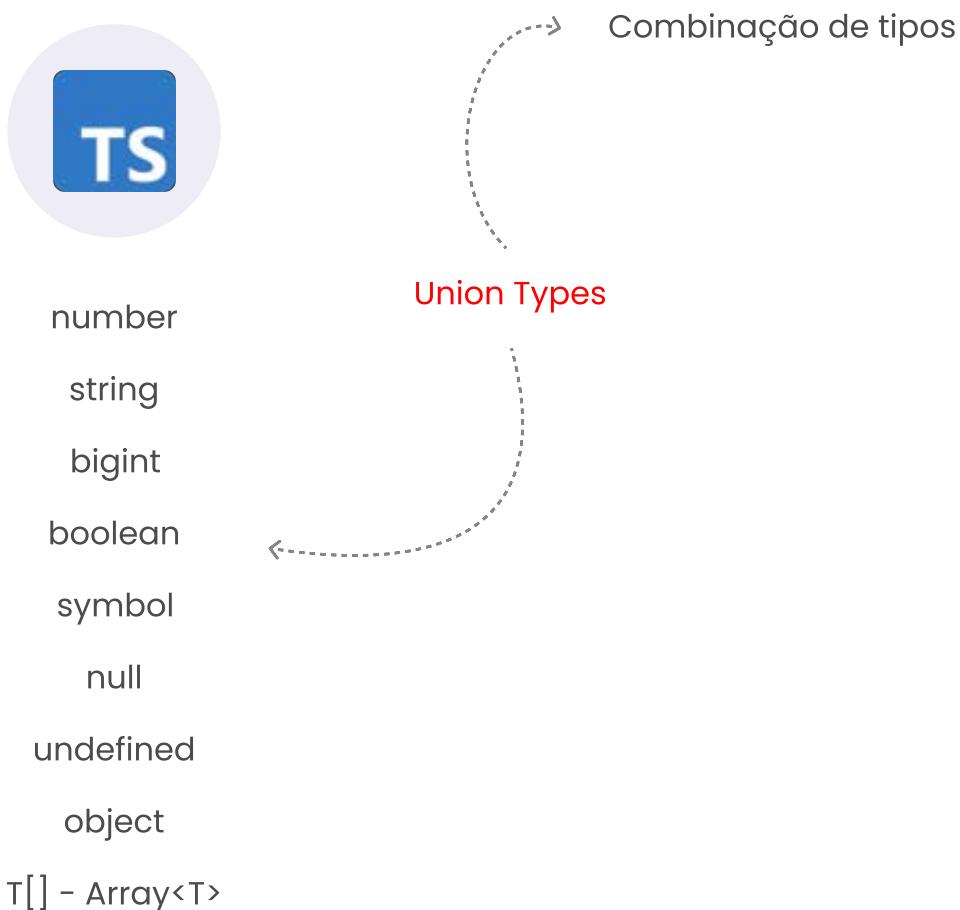
T[] - Array<T>

Interface para manipular tipos genéricos

Type Annotation convencional

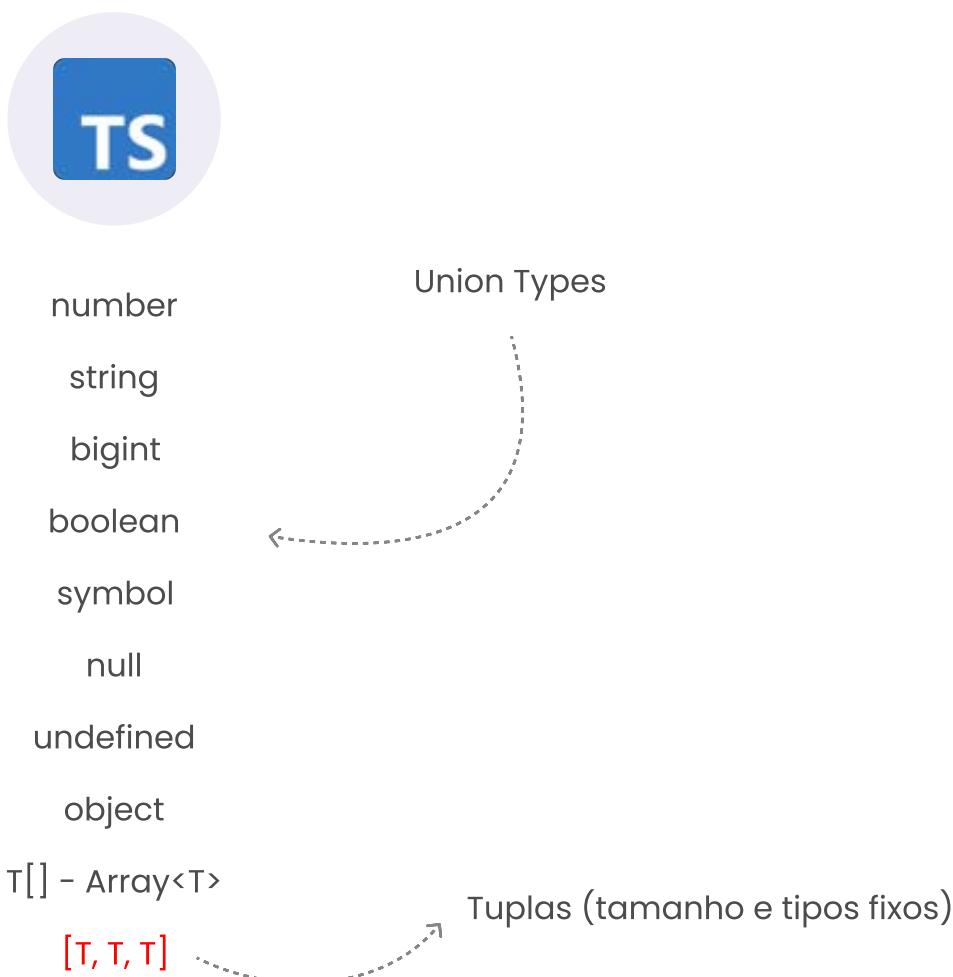
- TypeScript - Tipos Básicos

Union Types



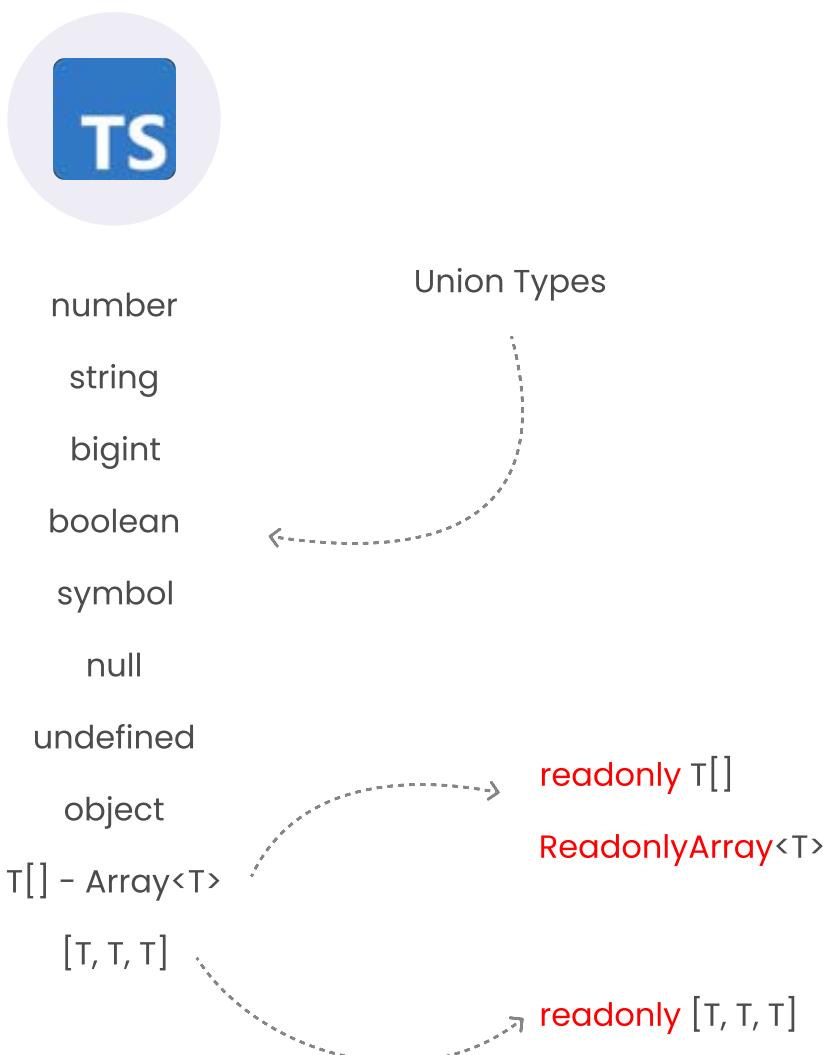
- TypeScript - Tipos Básicos

Tipo Tuple



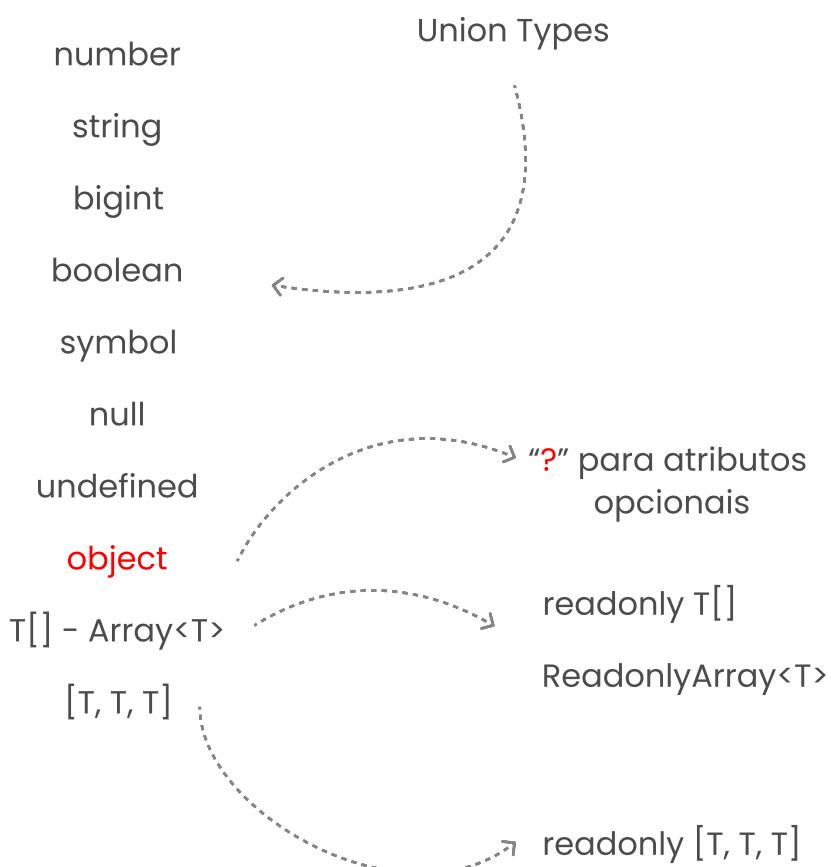
- TypeScript - Tipos Básicos

Readonly em Arrays e Tuples



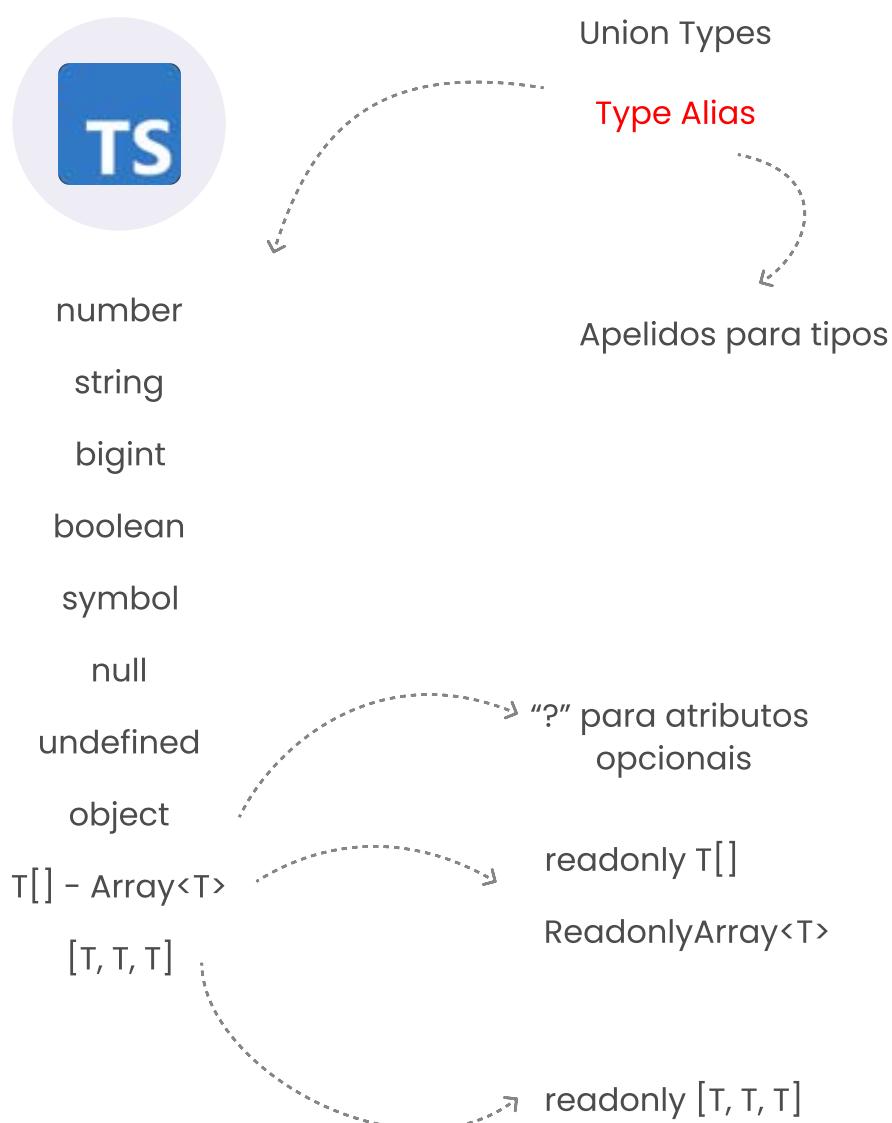
- TypeScript - Tipos Básicos

Tipo Object - Inferência e Anotação de Tipo



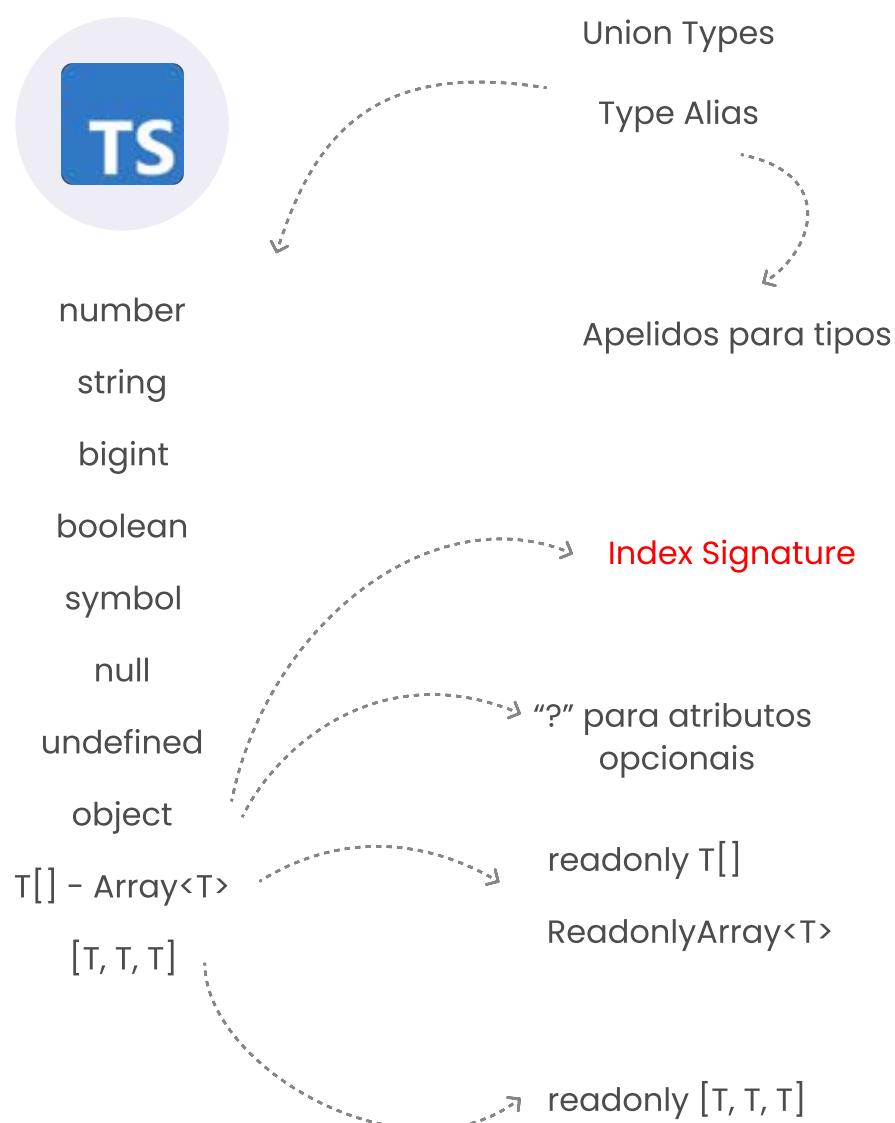
▶ TypeScript - Tipos Básicos

Type Alias



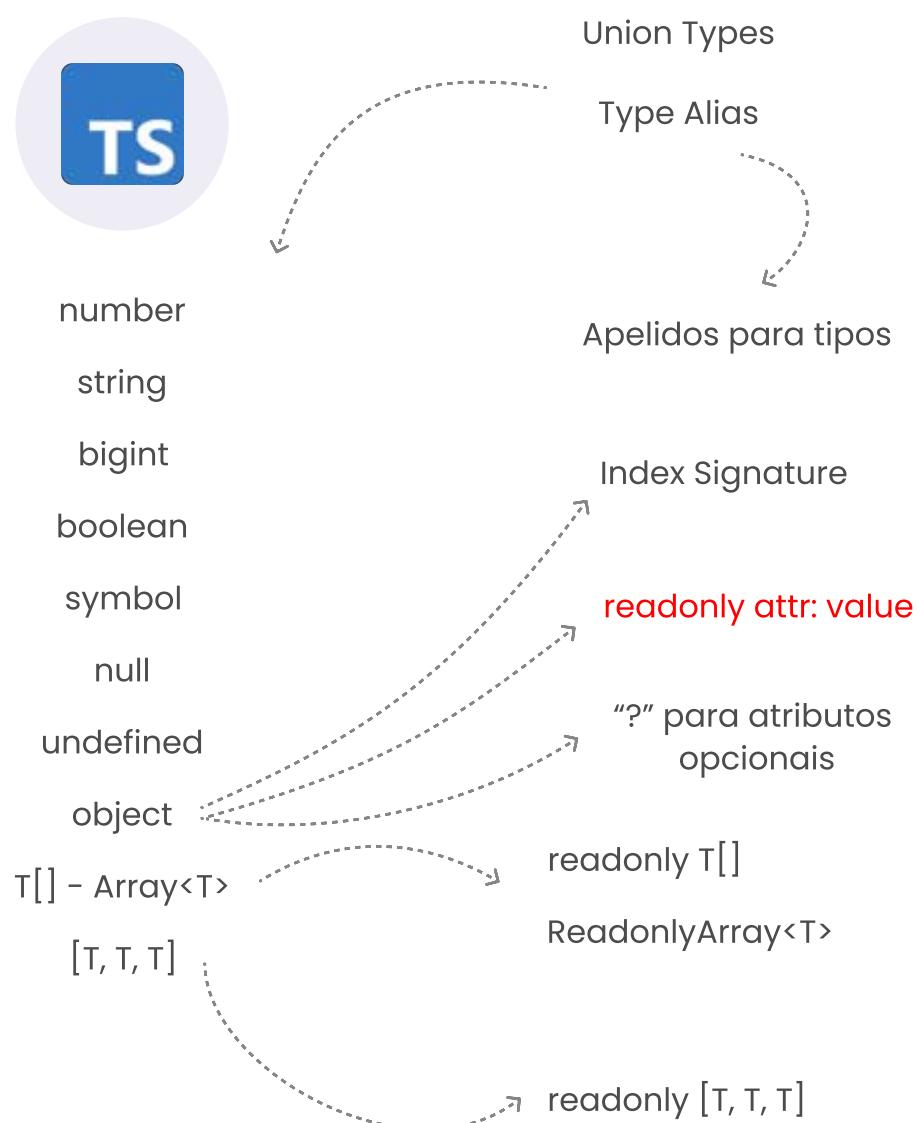
- TypeScript - Tipos Básicos

Tipo Object - Index Signature



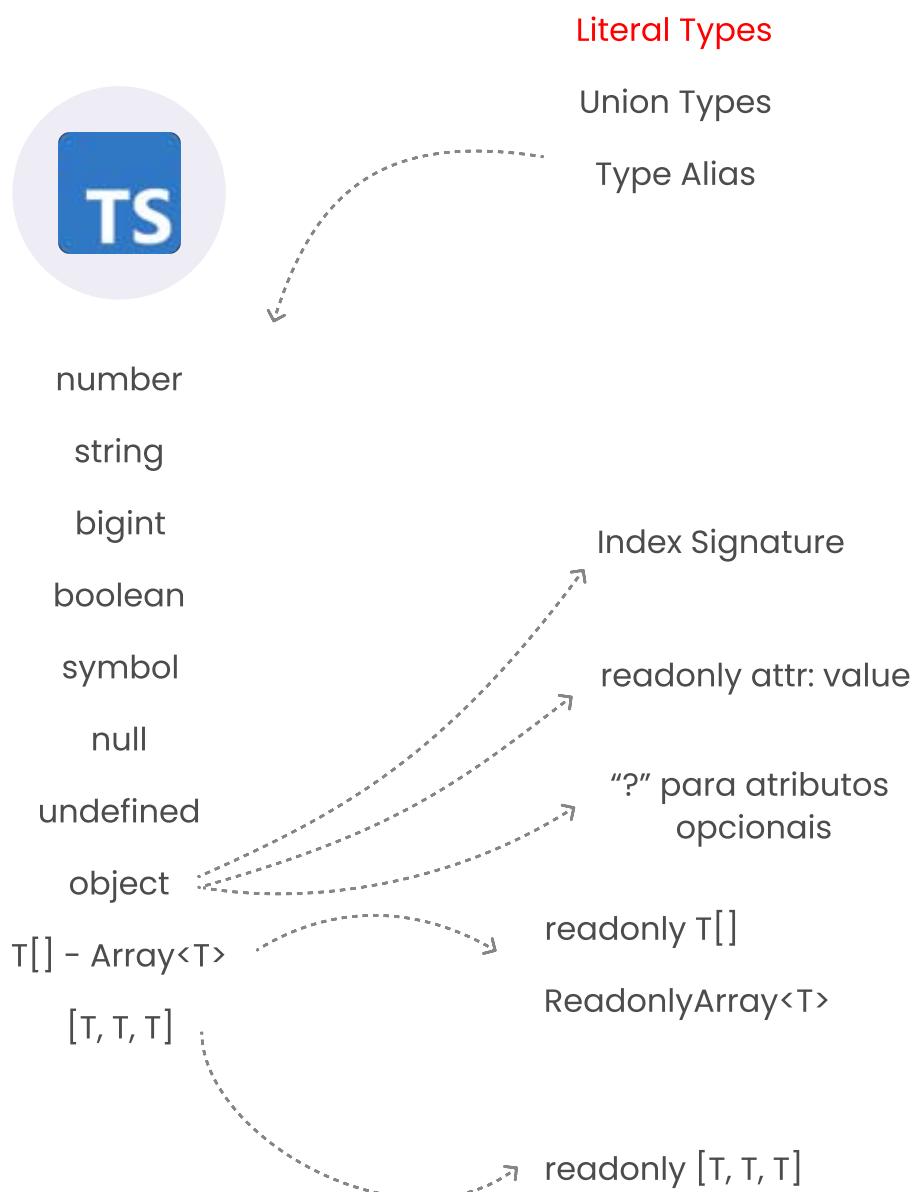
- TypeScript - Tipos Básicos

Tipo Object - Readonly



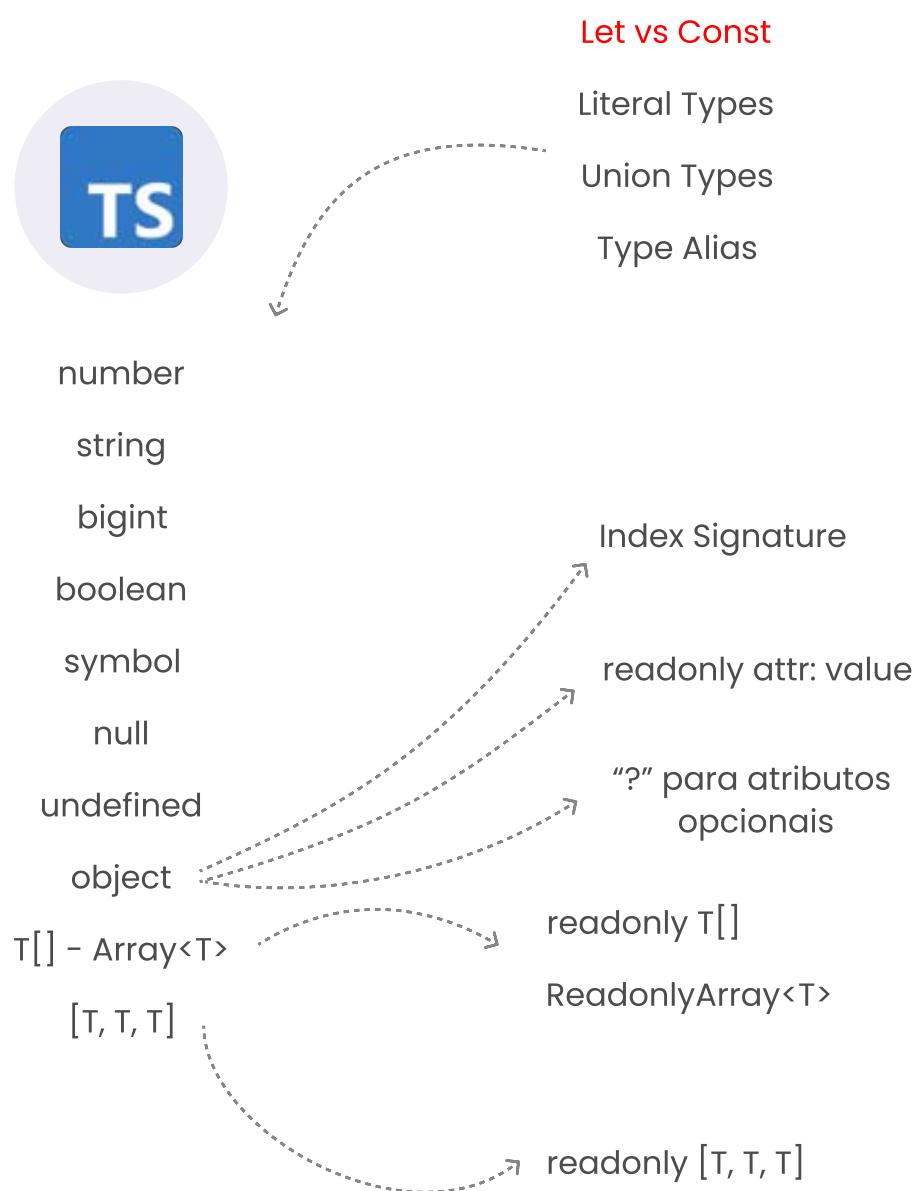
▶ TypeScript - Tipos Básicos

Literal Types



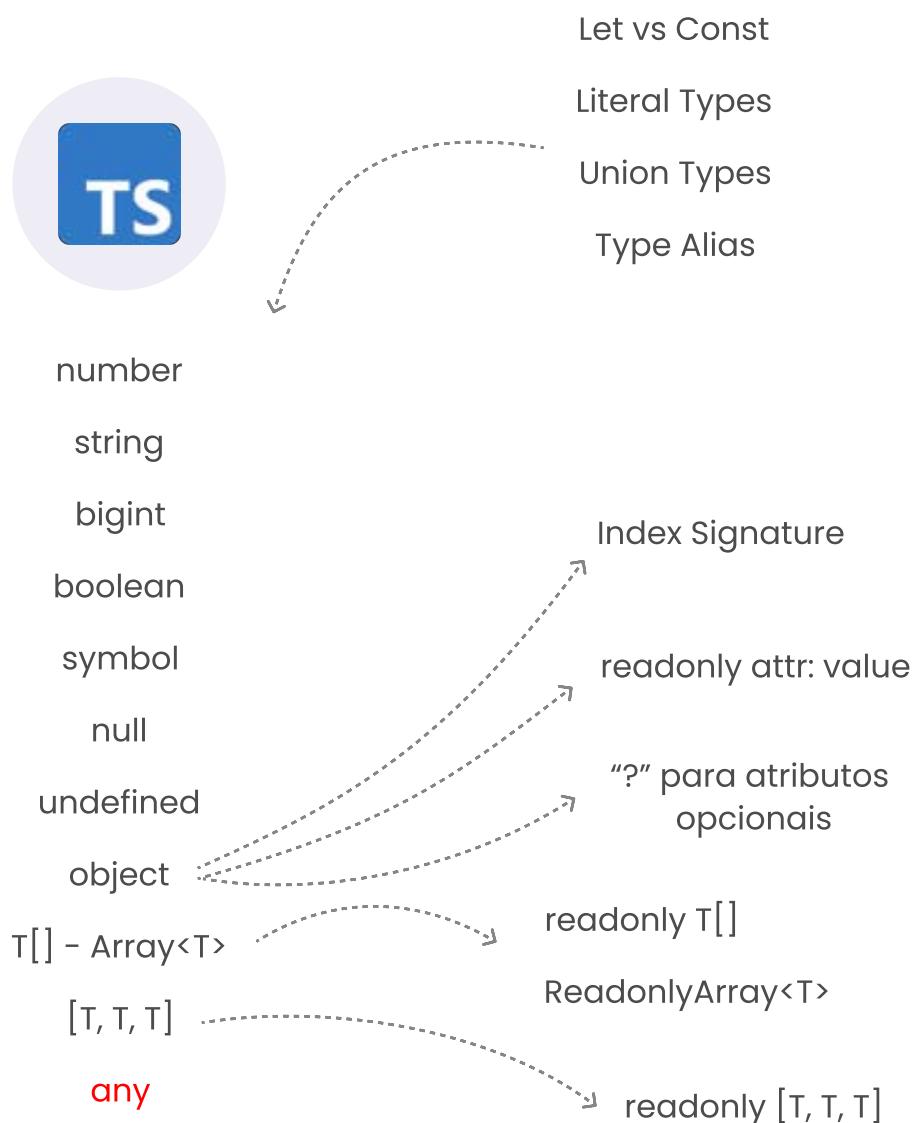
▶ TypeScript - Tipos Básicos

Let vs Const



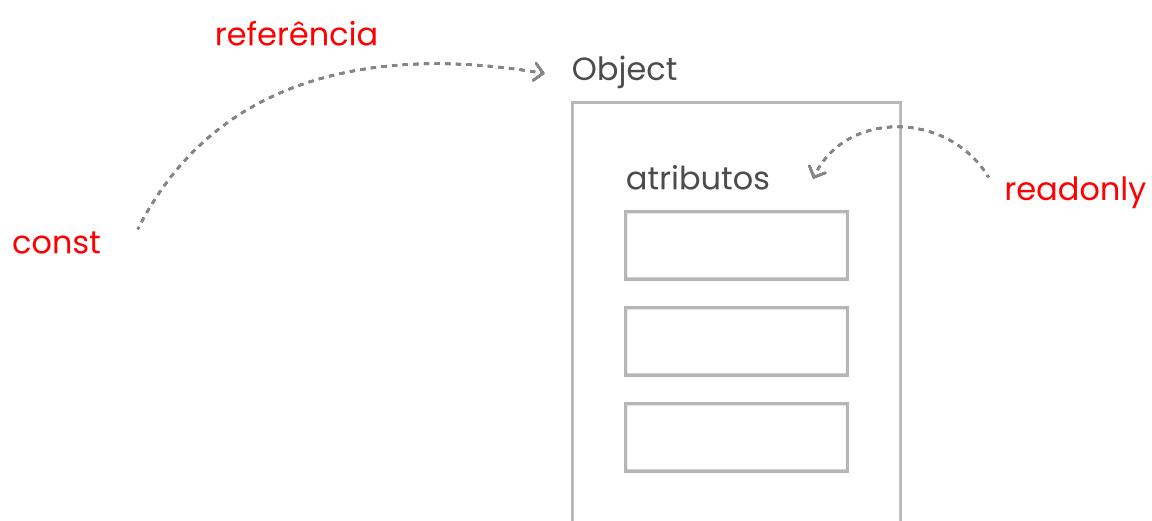
▶ TypeScript - Tipos Básicos

Tipo Any



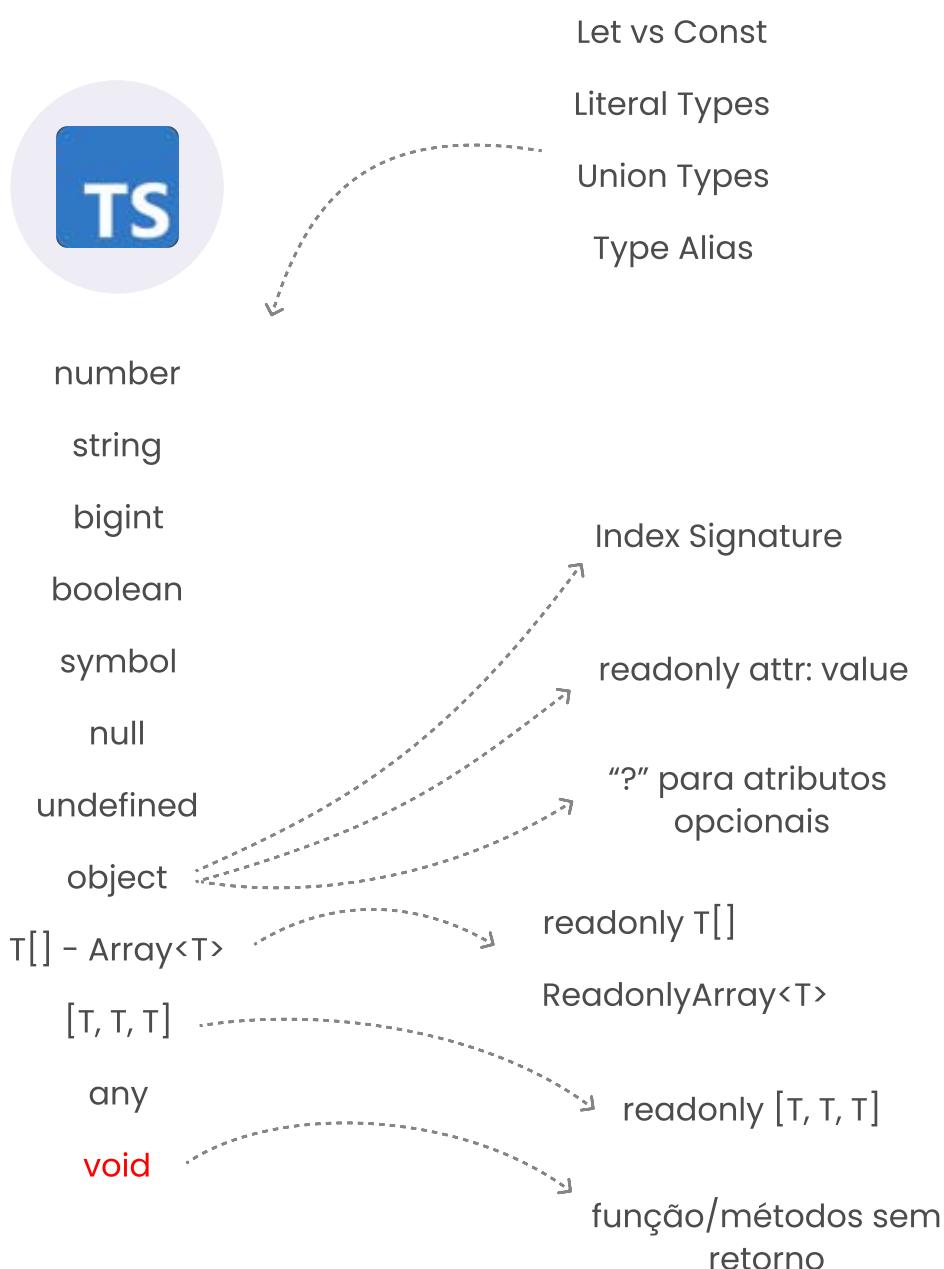
- ▶ TypeScript - Tipos Básicos

Const no Contexto de Objetos



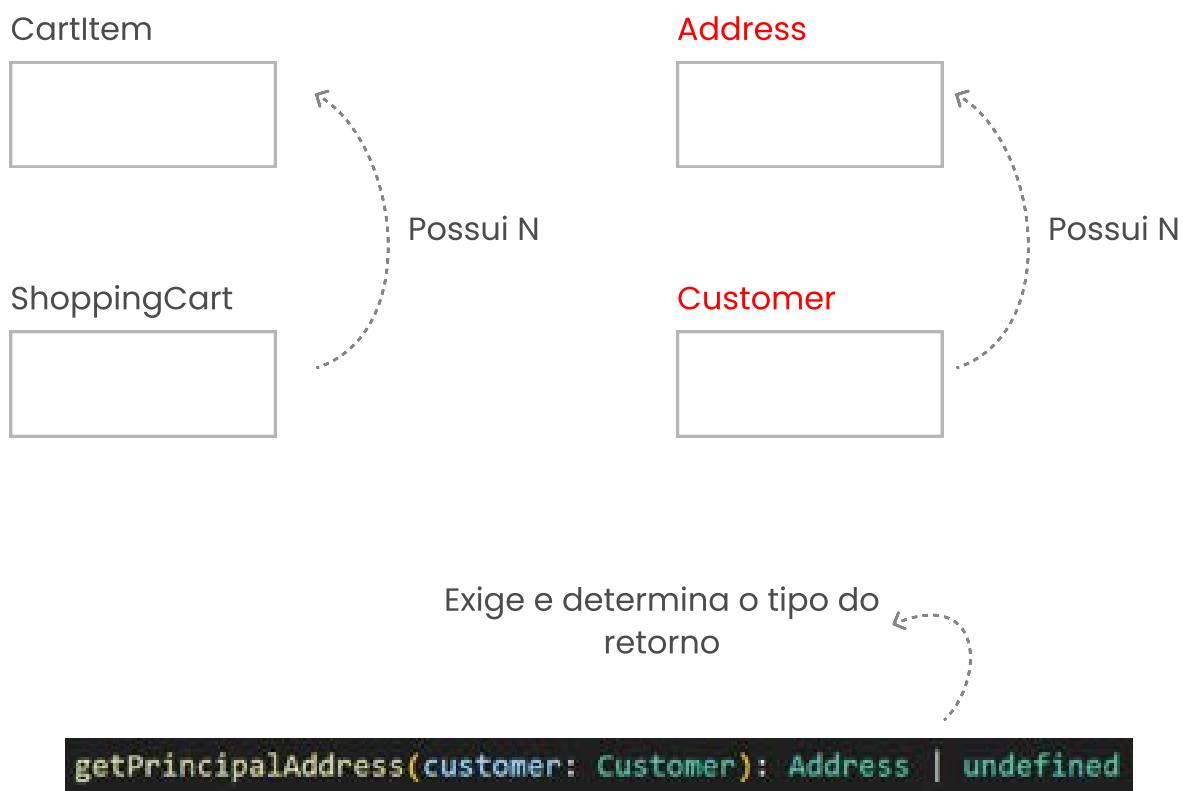
- TypeScript - Tipos Básicos

Função - Tipo Void



- ▶ TypeScript - Tipos Básicos

Função - Tipo Return



- ▶ TypeScript - Tipos Básicos

Função - Type Annotation

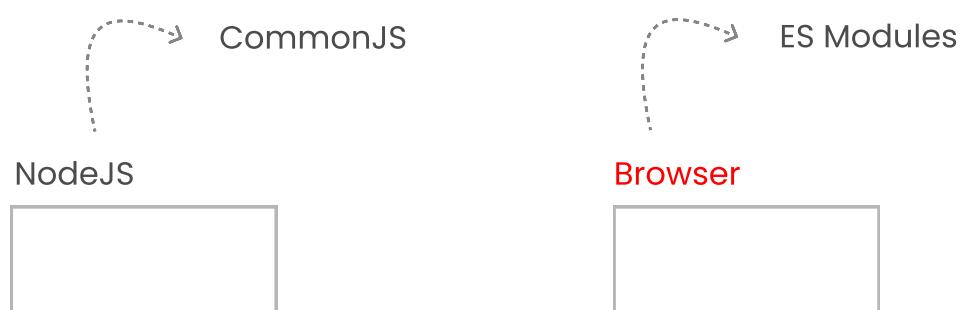
```
let calculateTotal: (sc: ShoppingCart) => number;
```



```
calculateTotal = function (shoppingCart: ShoppingCart): number {
```

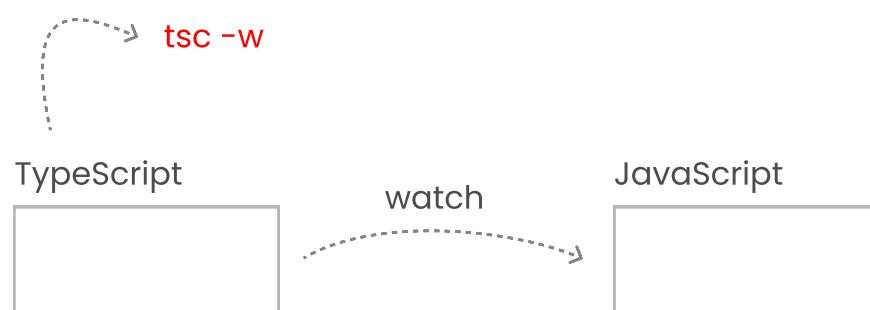
- ▶ TypeScript - Tipos Básicos

Executando os scripts compilados no Browser



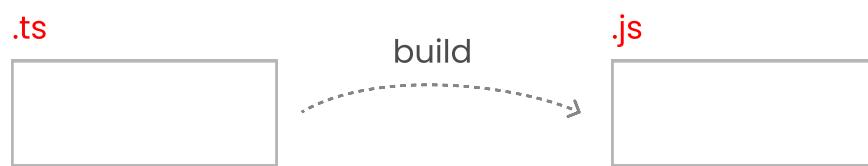
- ▶ TypeScript - Tipos Básicos

Compilação automática



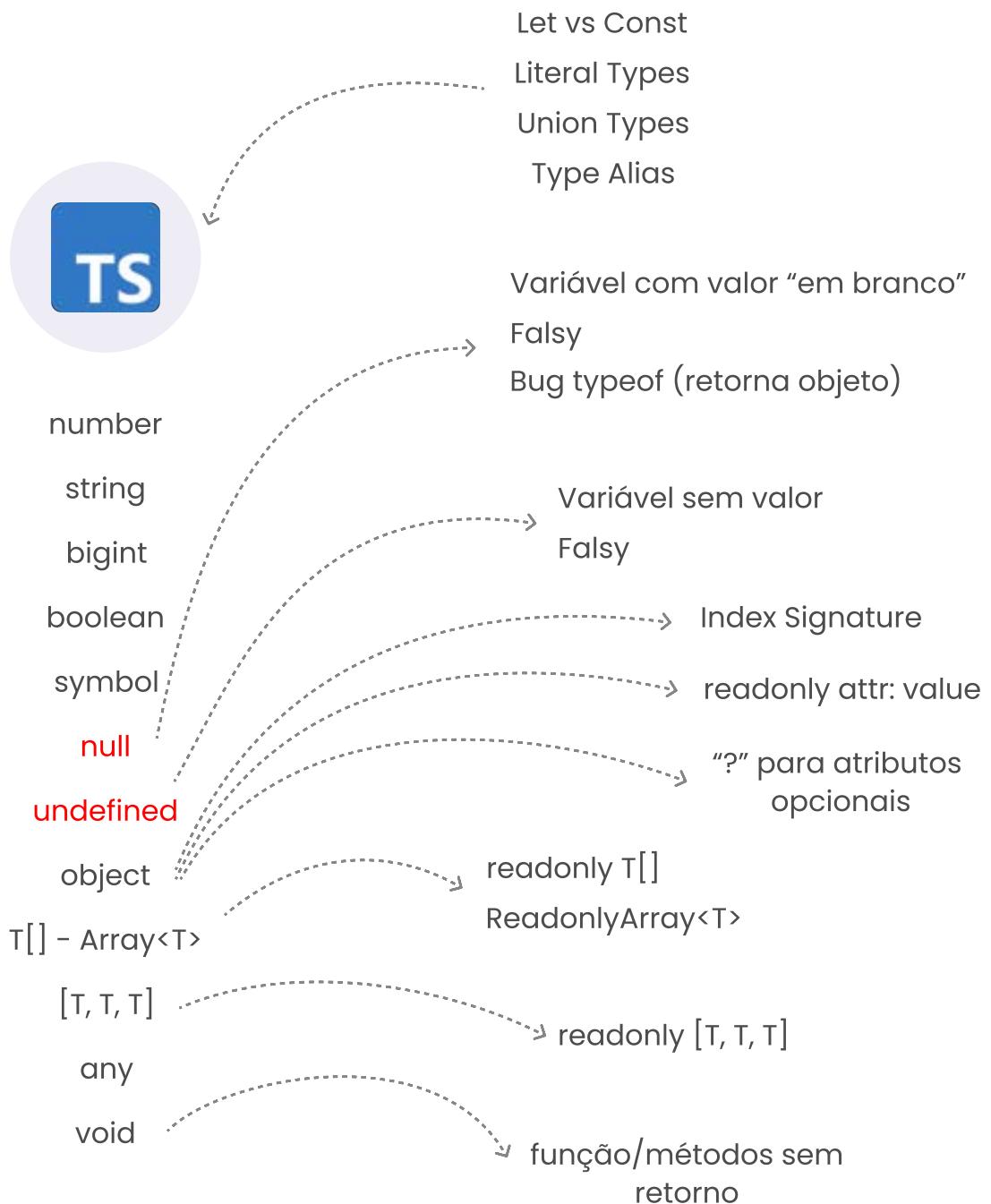
- ▶ TypeScript - Tipos Básicos

Extensões TS e JS Antes e Depois do Build



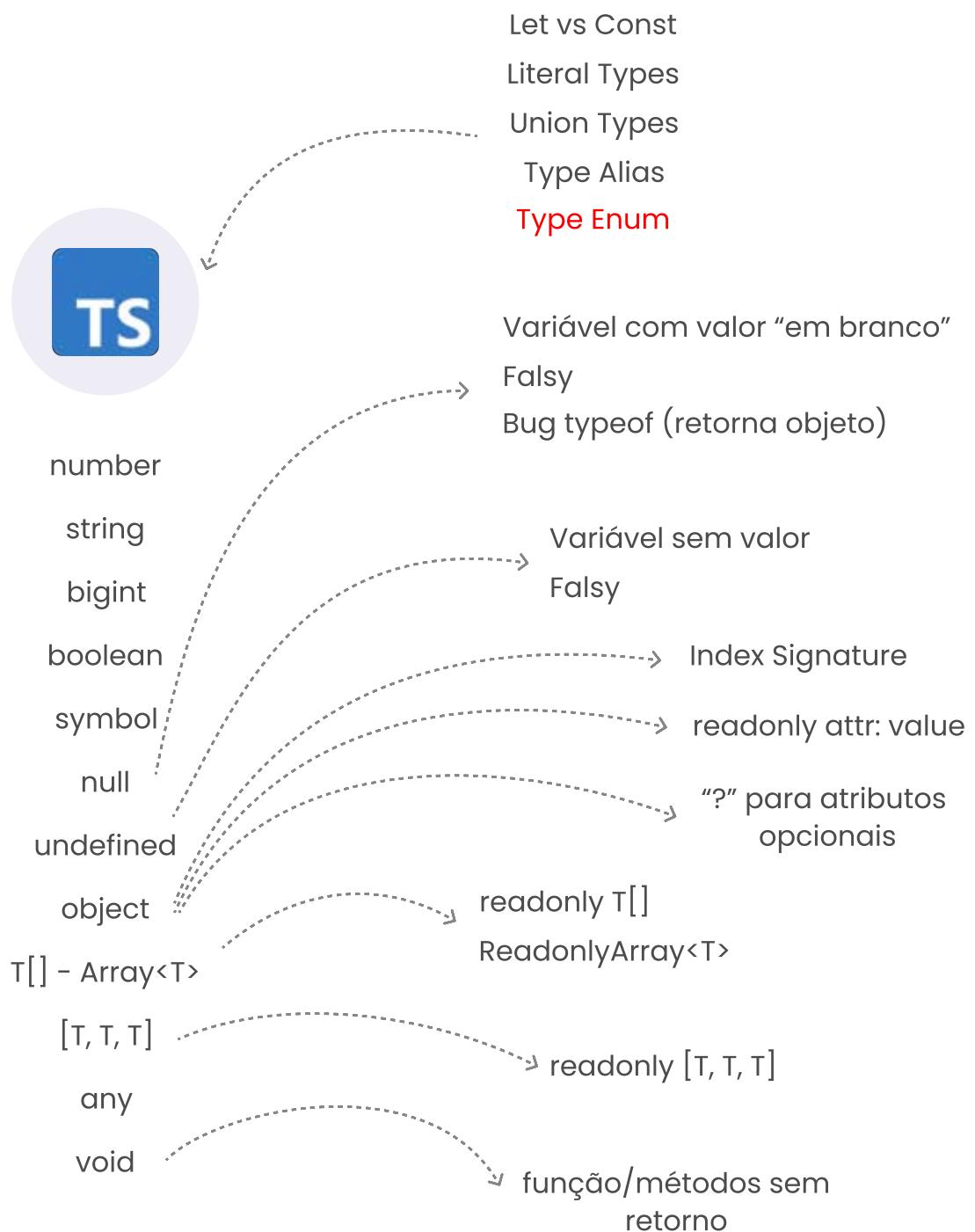
▶ TypeScript - Tipos Básicos

Tipos Null e Undefined



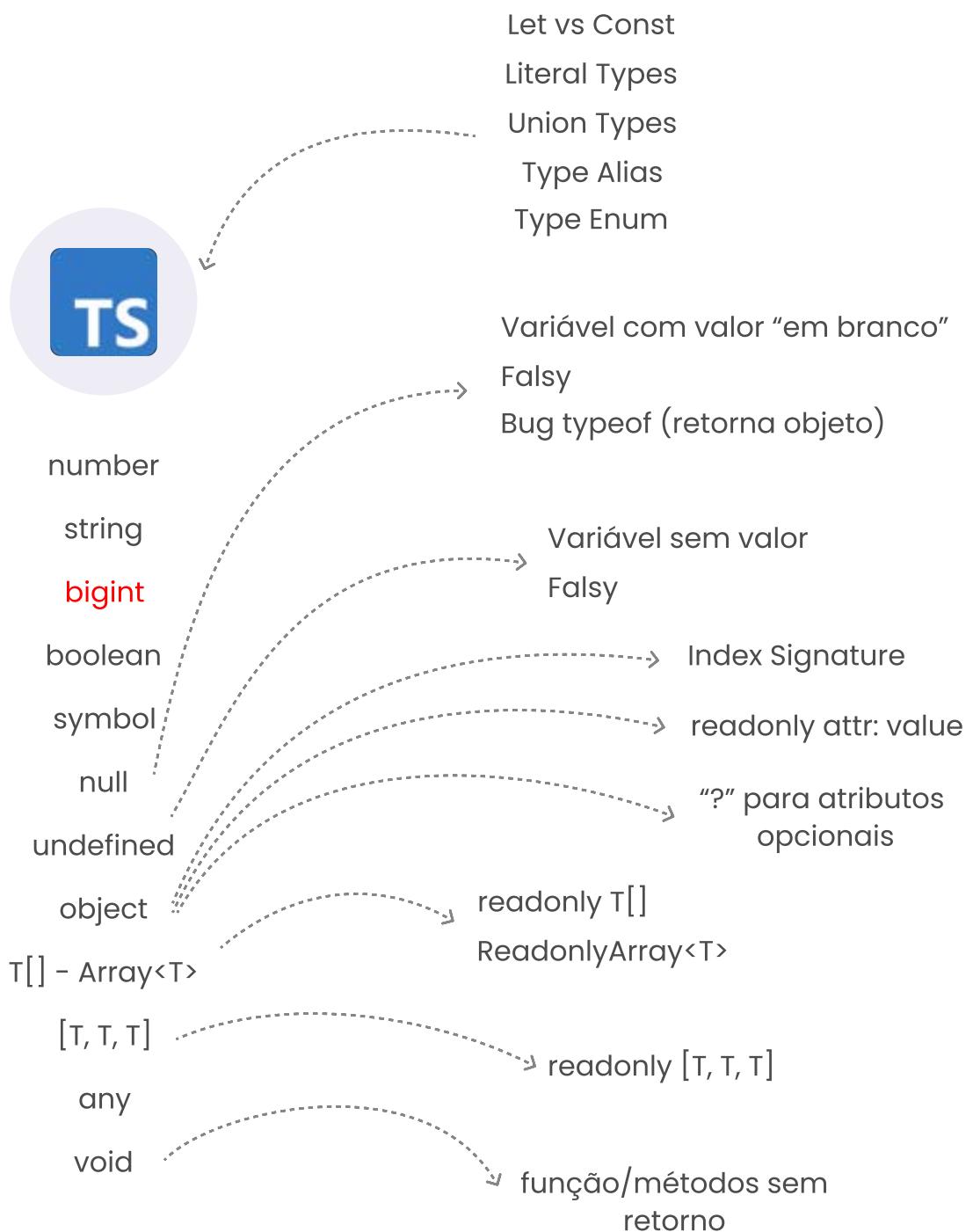
▶ TypeScript - Tipos Básicos

Tipo Enum



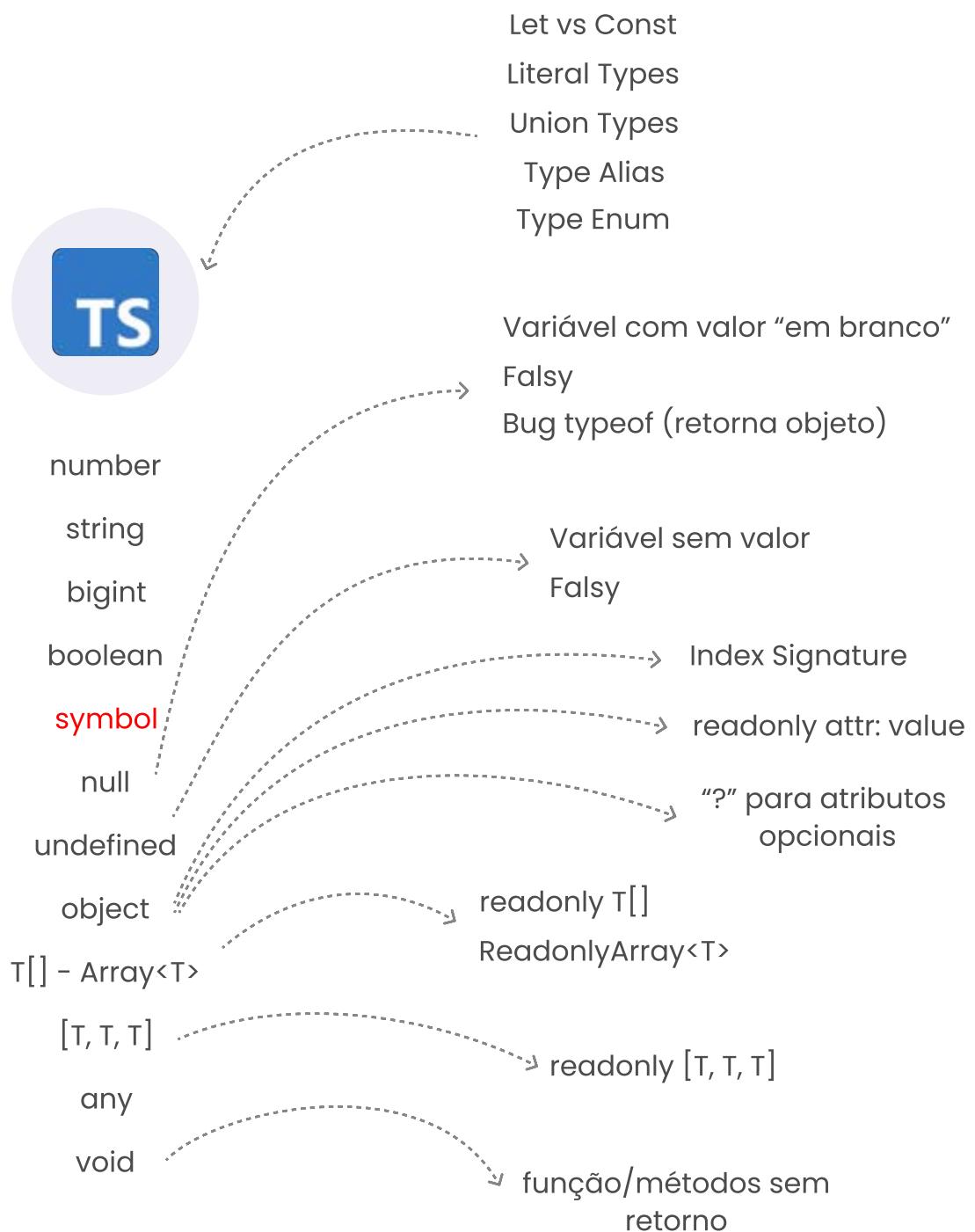
▶ TypeScript - Tipos Básicos

Tipo BigInt



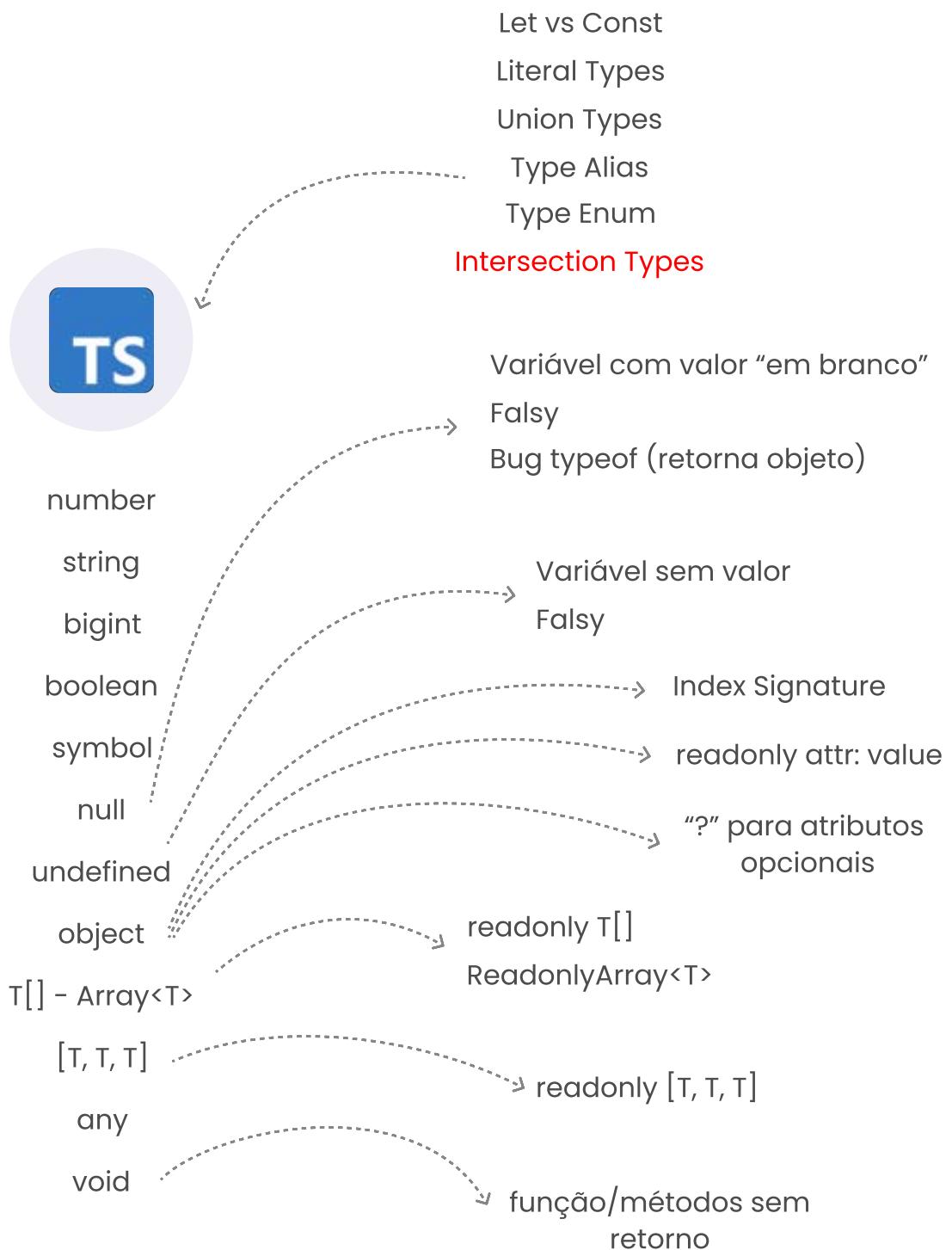
▶ TypeScript - Tipos Básicos

Tipo Symbol



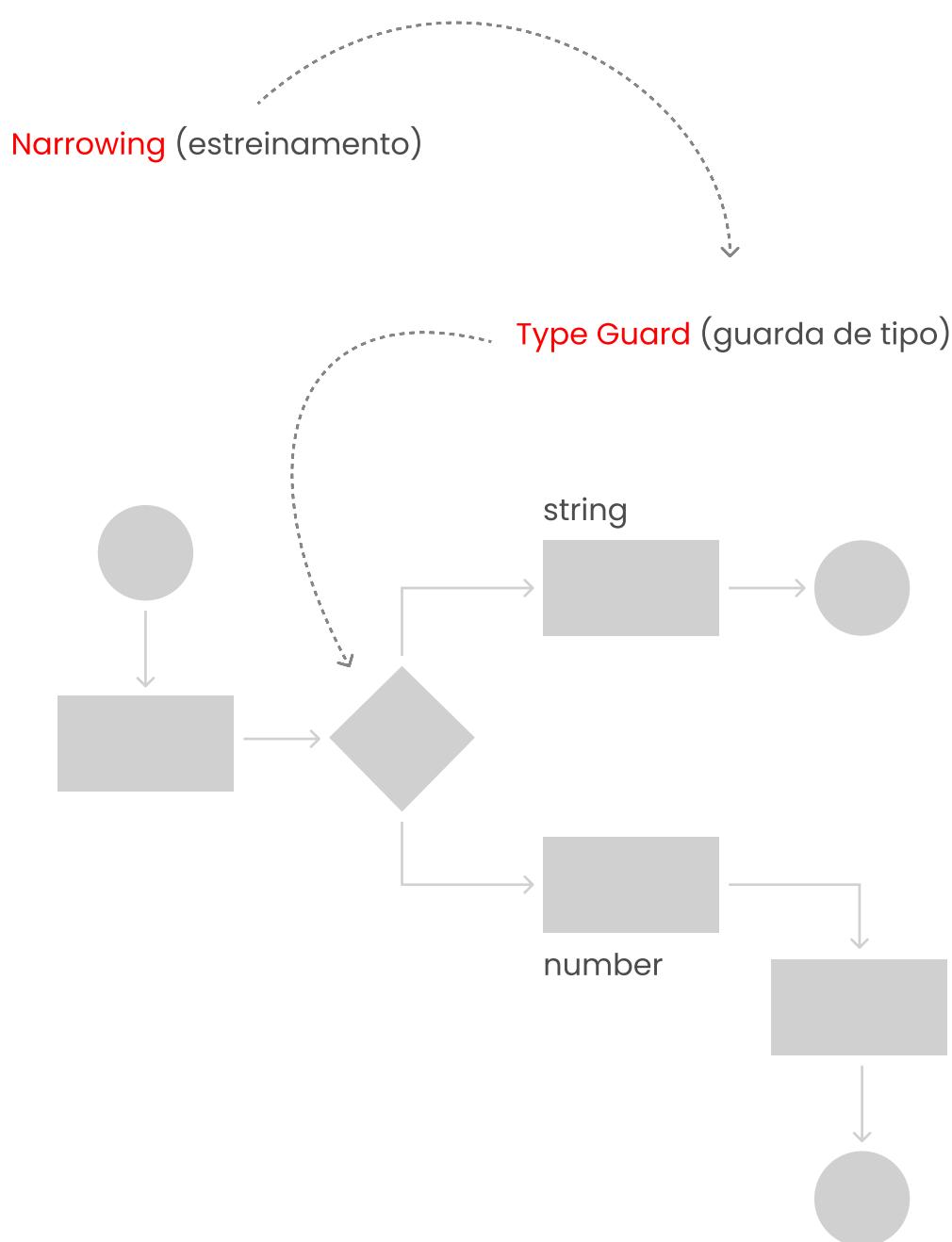
▶ TypeScript - Tipos Básicos

Intersection Types



▶ TypeScript – Narrowing

Type Guard



- ▶ TypeScript - Narrowing

typeof

Narrowing (estreinamento)

Type Guard (guarda de tipo)

typeof

Retorna o tipo da variável

- ▶ TypeScript - Narrowing

ValoresTruthy e Falsy

Narrowing (estreinamento)

Type Guard (guarda de tipo)

`typeof`

truthy e falsy

Retorna o tipo da variável

Valores booleanos

- ▶ TypeScript - Narrowing

Optional Chaining (?)

optional chaining (?)

Leitura segura de propriedades de
objetos

- ▶ TypeScript - Narrowing

Non-null Assertion (!)

non-null assertion (!)

Indica que a variável ou expressão
não é null ou undefined

- ▶ TypeScript - Narrowing

Type Assertion (as)

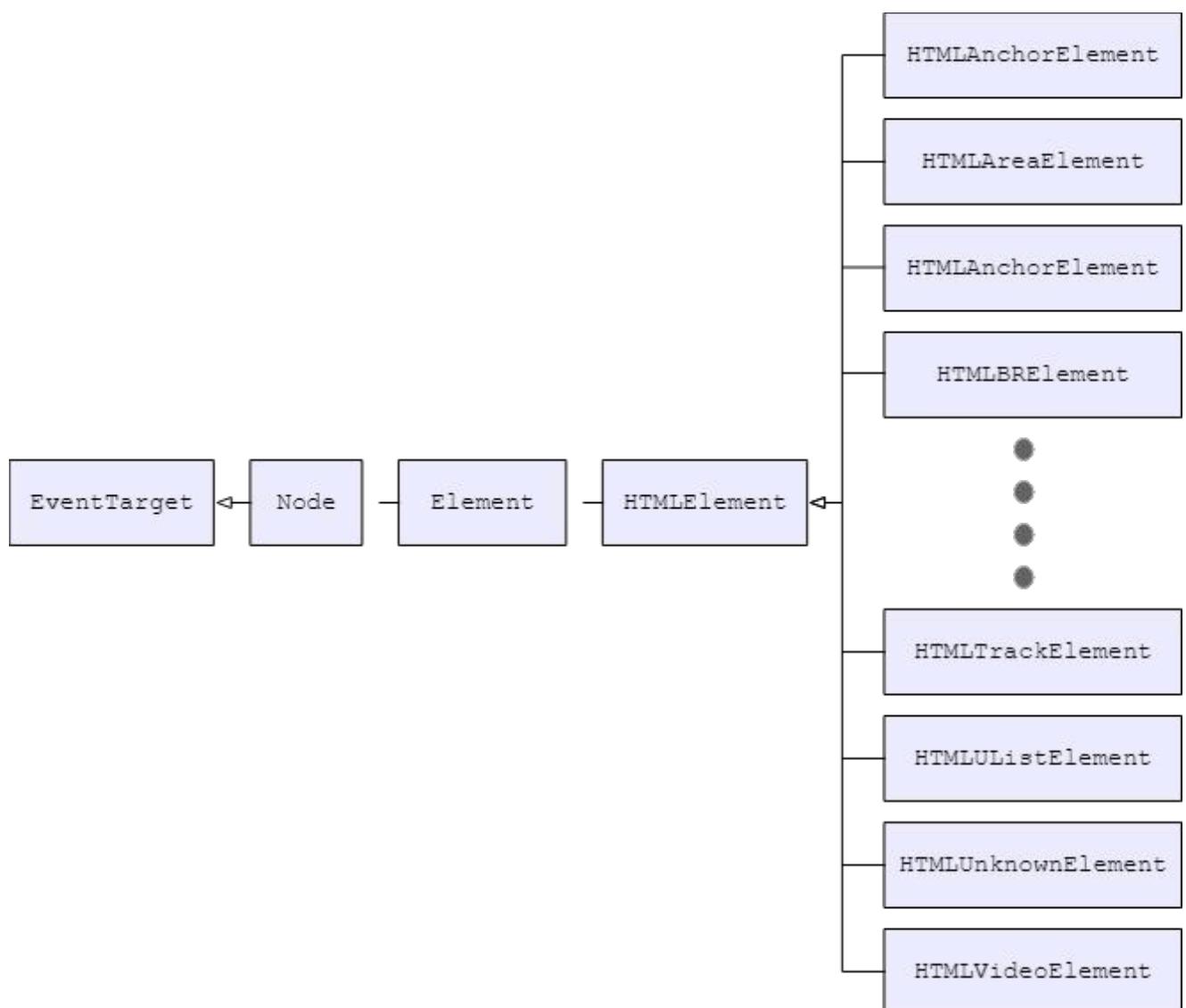
type assertion (as)



Definição de tipo mais específico

- ## ► TypeScript - Narrowing

Interfaces dos elementos HTML



Fonte: https://developer.mozilla.org/en-US/docs/Web/API/HTML_DOM_API

- ▶ TypeScript - Narrowing

Verificando o Tipo do Objeto (`Instanceof`) Parte 1

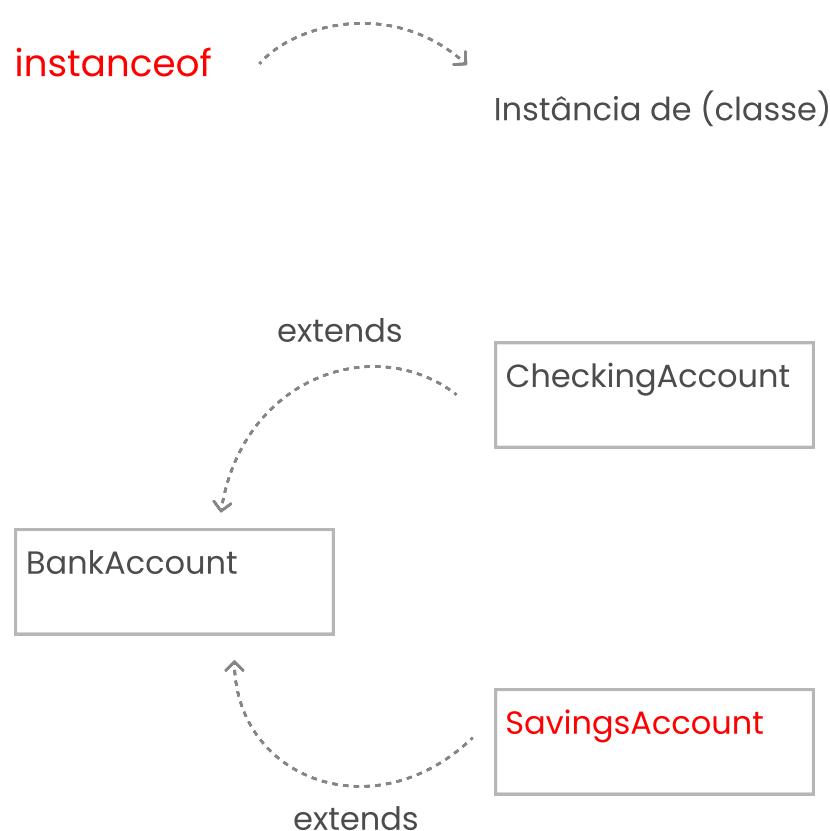
`instanceof` ↗
Instância de (classe)

BankAccount

CheckingAccount

- ▶ TypeScript - Narrowing

Verificando o Tipo do Objeto (Instanceof) Parte 2



- ▶ TypeScript - Narrowing

Verificando Propriedades de Objetos e Índices de Arrays (in)

in



Verifica se propriedade existe em objeto e ou
se índice existe em array

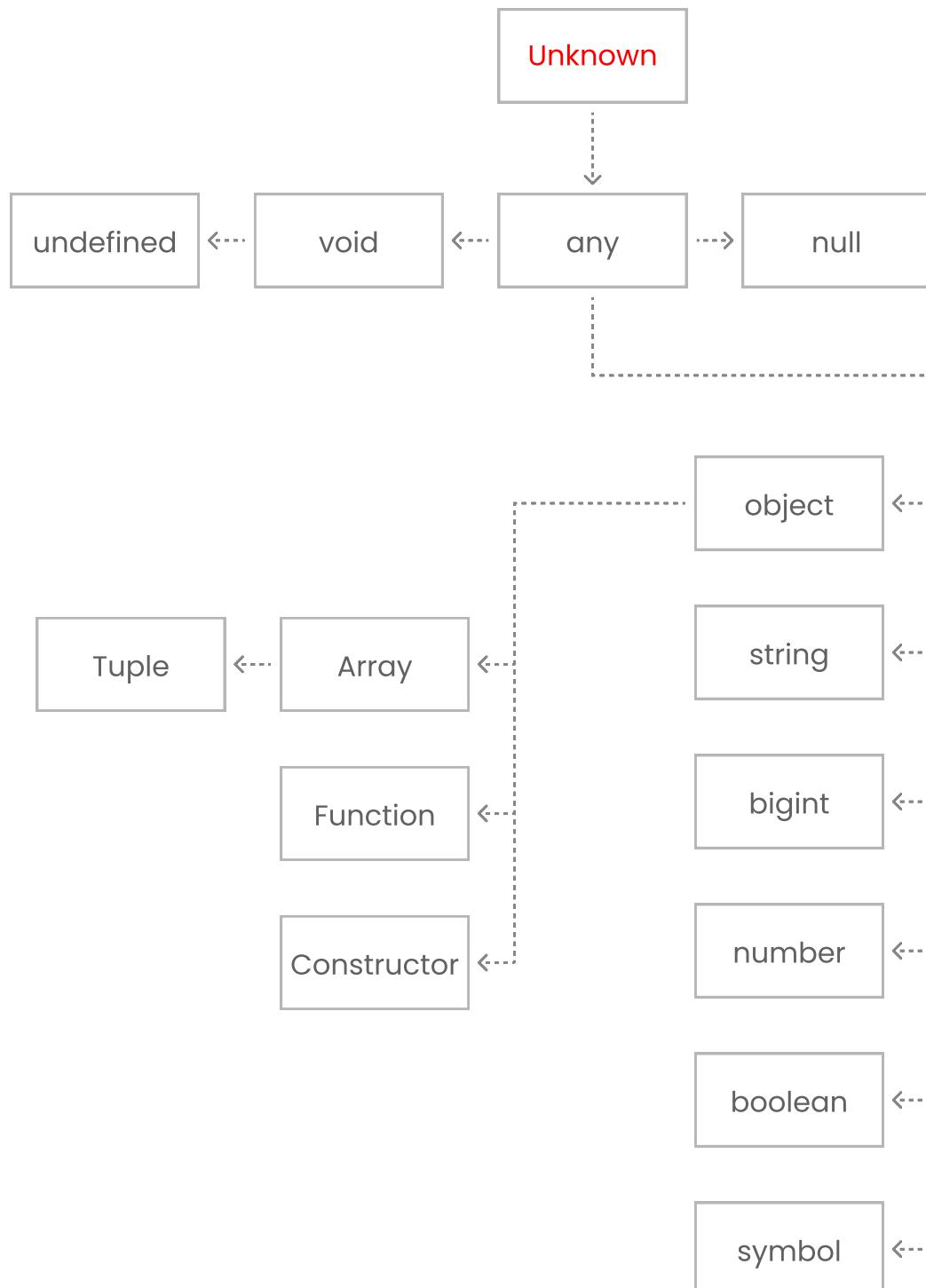
- ▶ TypeScript - Narrowing

Array.isArray

Array.isArray  Verifica se o objeto é um Array

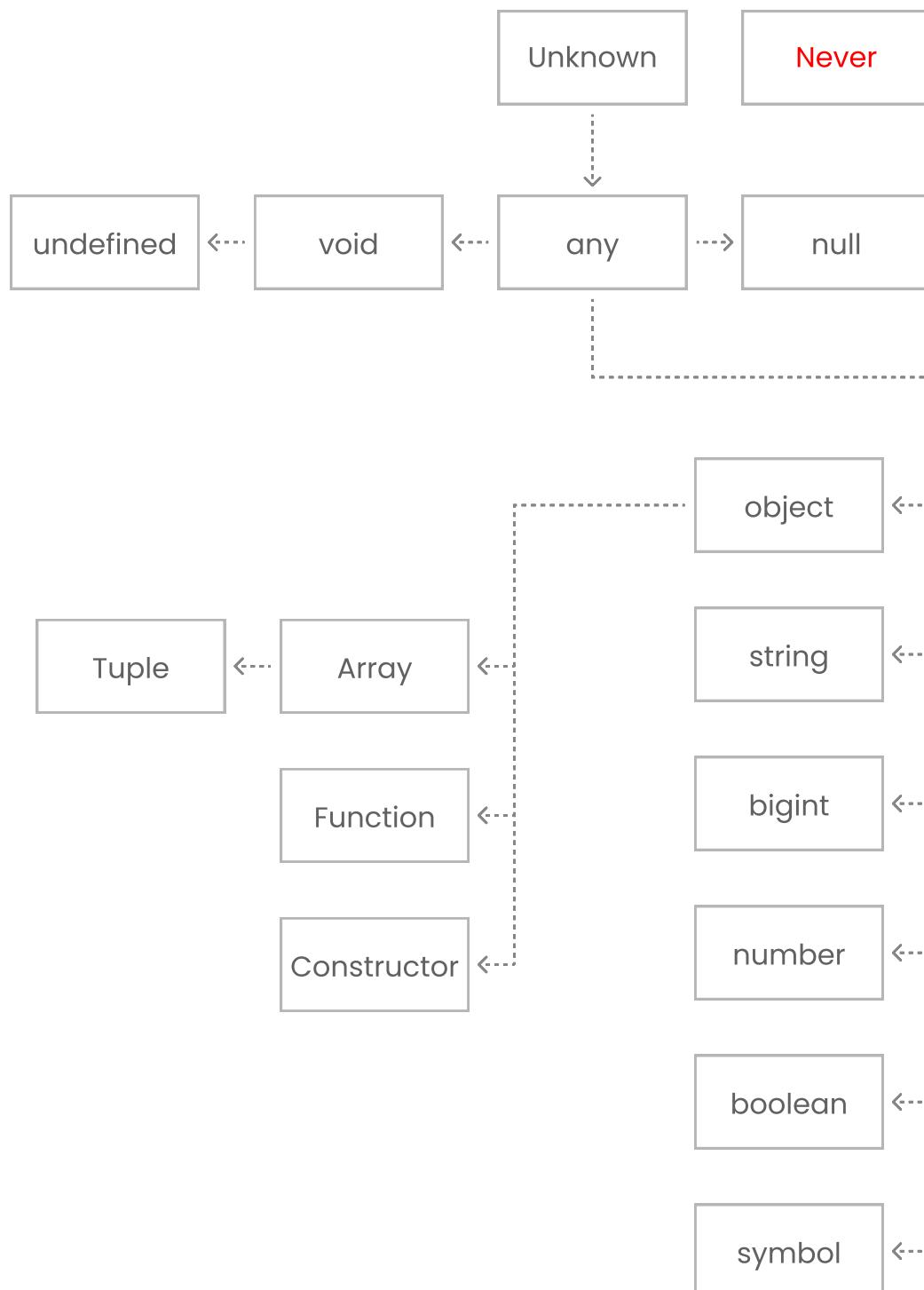
▶ TypeScript – Narrowing

Hierarquia de Tipos e o Tipo Unknown



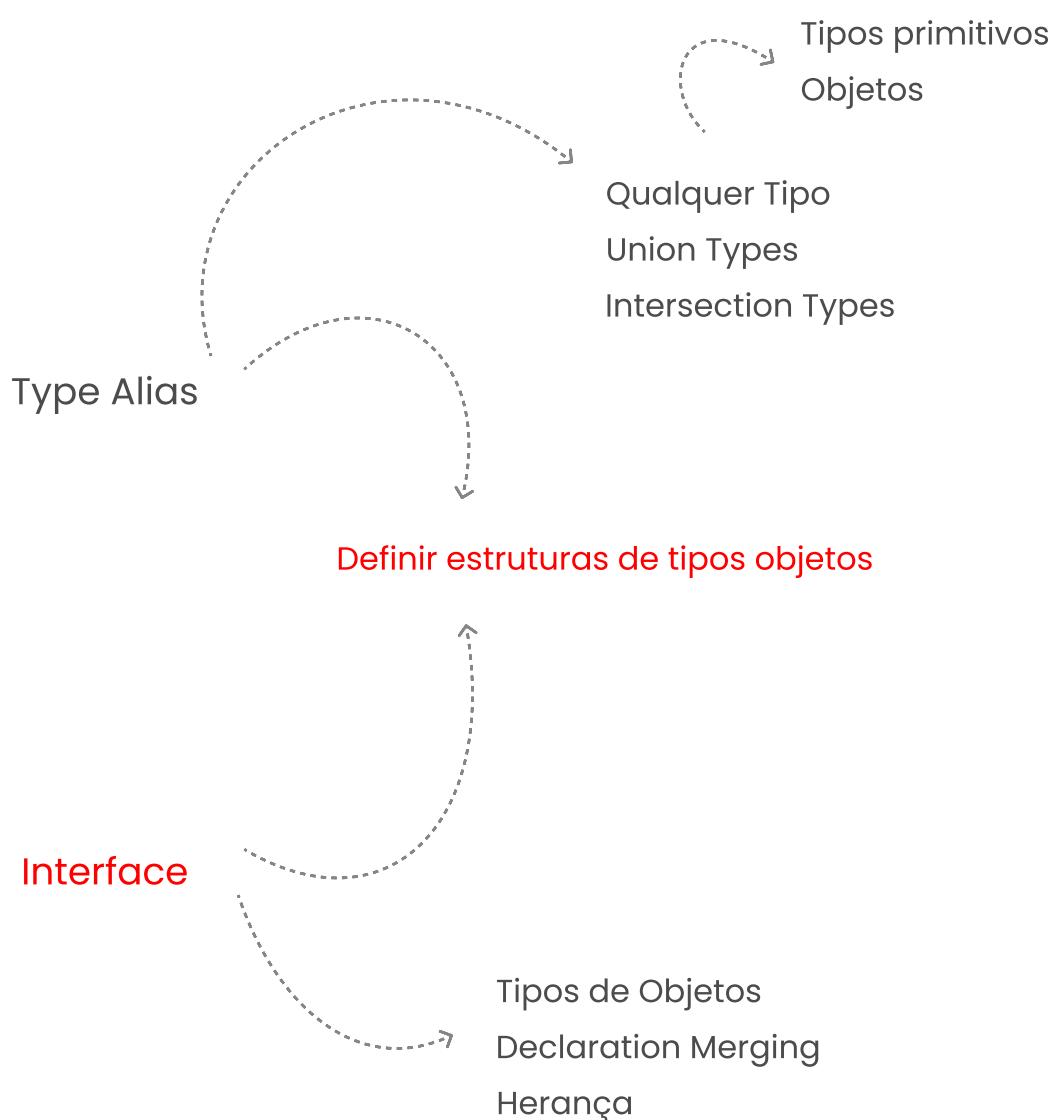
▶ TypeScript – Narrowing

Hierarquia de Tipos e o Tipo Never



- ▶ TypeScript - Interface

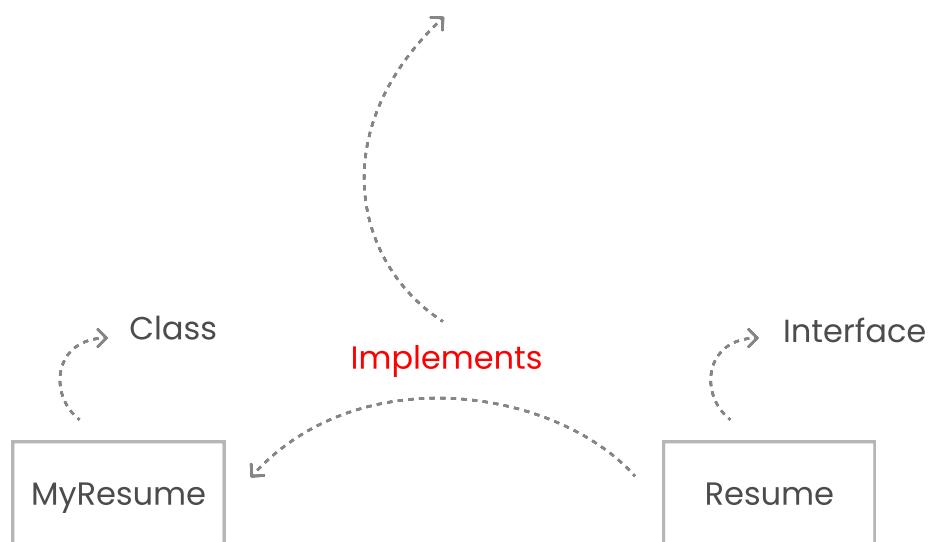
Introdução a Interfaces



- ▶ TypeScript - Interface

Implementando Interfaces (Implements)

Indica que a Classe deve seguir a estrutura definida na Interface



- ▶ TypeScript - Interface

Merge de Múltiplas Declarações

Resume

```
fullName: string  
email: string
```



Resume

```
dateOfBirth?: Date  
sumary?: string
```



Declaration Merging

Resume

```
fullName: string  
email: string  
dateOfBirth?: Date  
sumary?: string
```

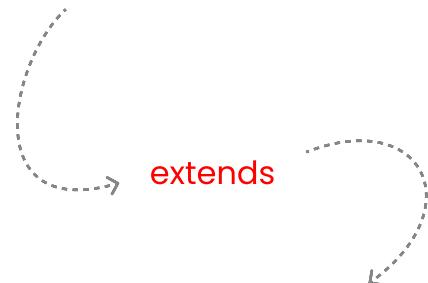


- ▶ TypeScript - Interface

Estendendo Interfaces (Extends)

Resume

```
skills: Skill[]  
addSkill: (skill: Skill) => boolean
```

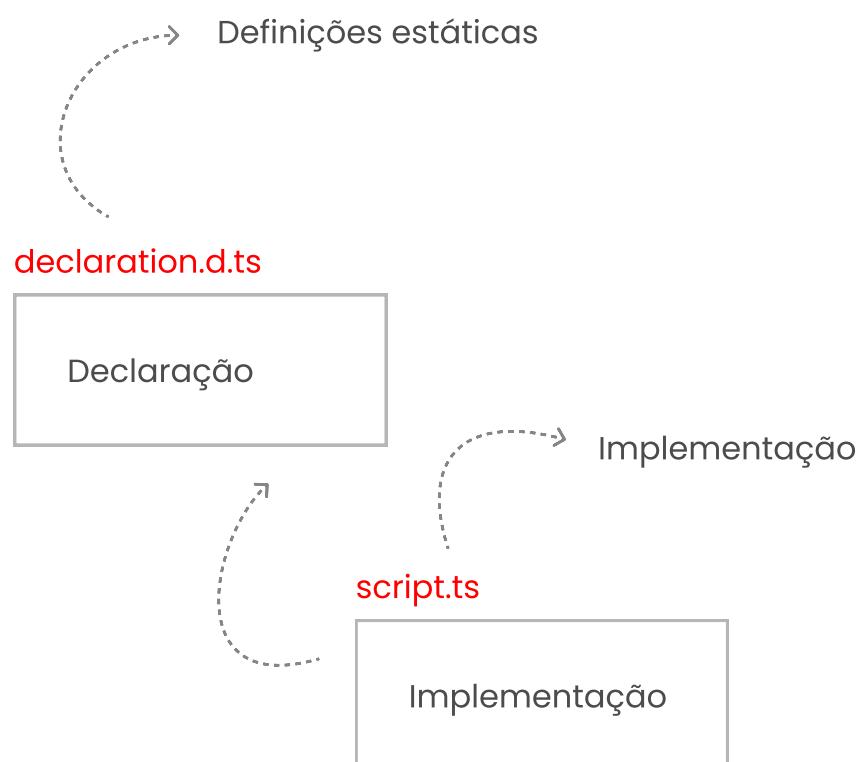


PersonalInfo

```
fullName: string  
email: string  
dateOfBirth?: Date  
sumary?: string
```

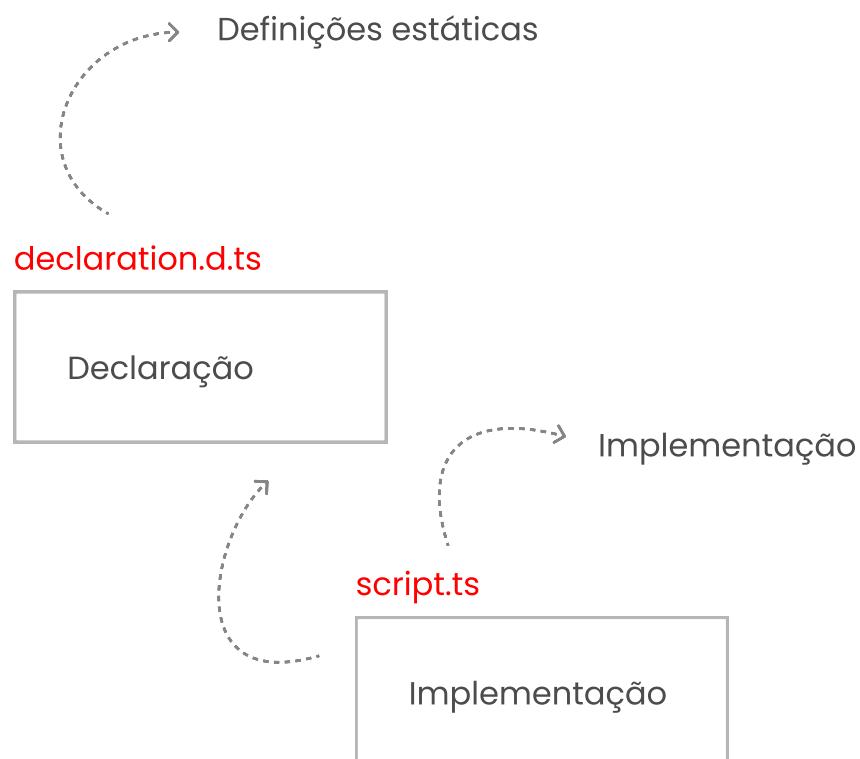
- ▶ TypeScript - Interface

Arquivos de Declaração (Declaration Files)



- ▶ TypeScript - Interface

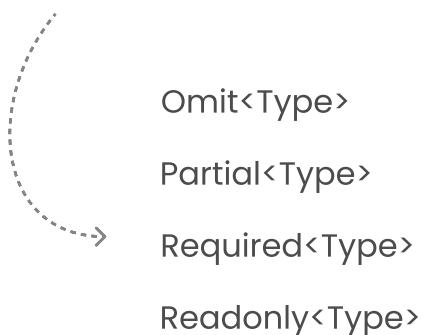
Gerando arquivos de declarações automaticamente



- ▶ TypeScript - Utility Types

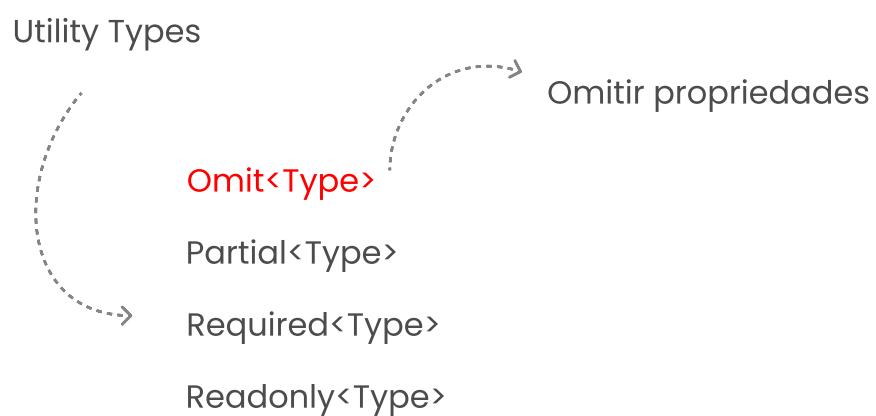
Introdução aos Tipos Utilitários (Utility Types)

Utility Types



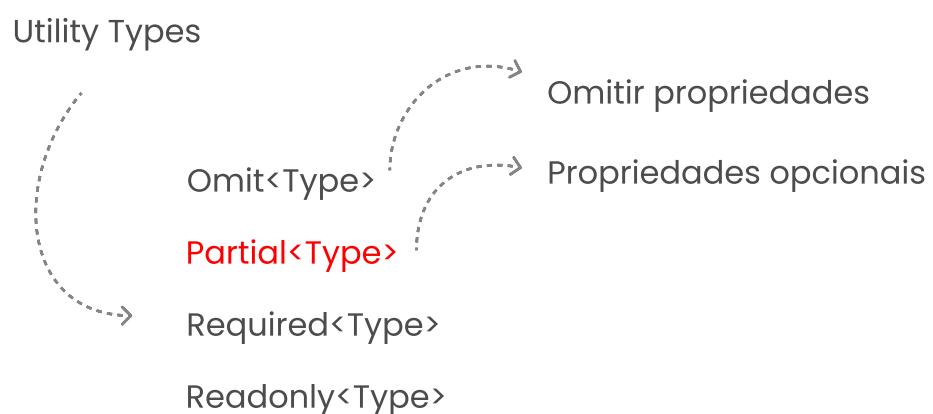
▶ TypeScript - Utility Types

Omit<Type>



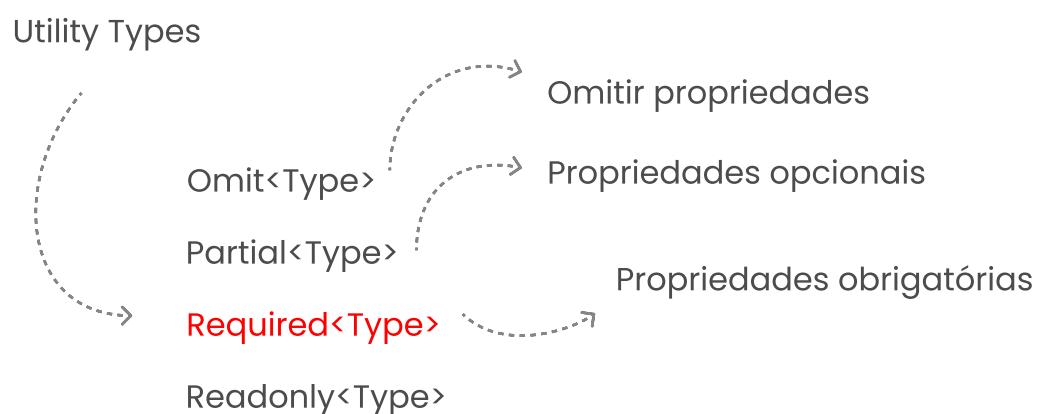
- ▶ TypeScript - Utility Types

Partial<Type>



▶ TypeScript - Utility Types

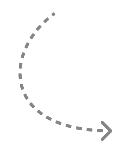
Required<Type>



- ▶ TypeScript - Utility Types

Type Predicate (is)

Type Predicate



Função especial que indica ao compilador do TypeScript que um valor é de um tipo específico

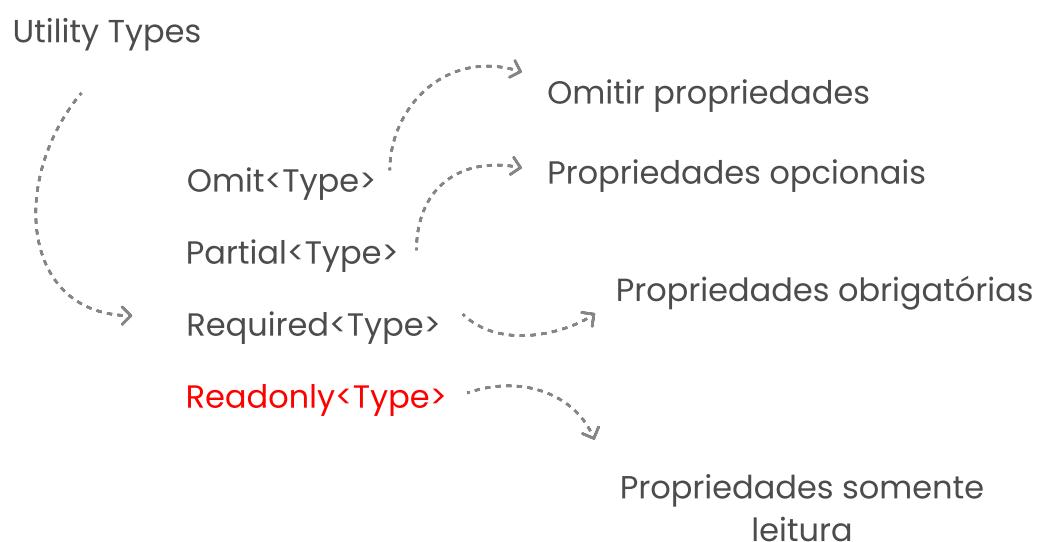


A função verifica o tipo do dado e retorna um valor booleano

```
function nomeDaFuncao(variavel: unknown): variavel is TipoEsperado {  
    // lógica de verificação  
}
```

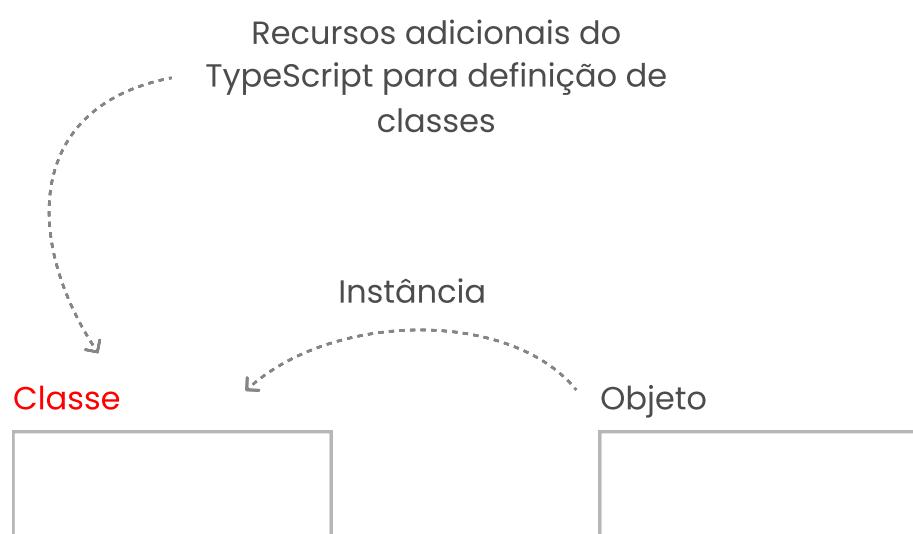
▶ TypeScript - Utility Types

Readonly<Type>



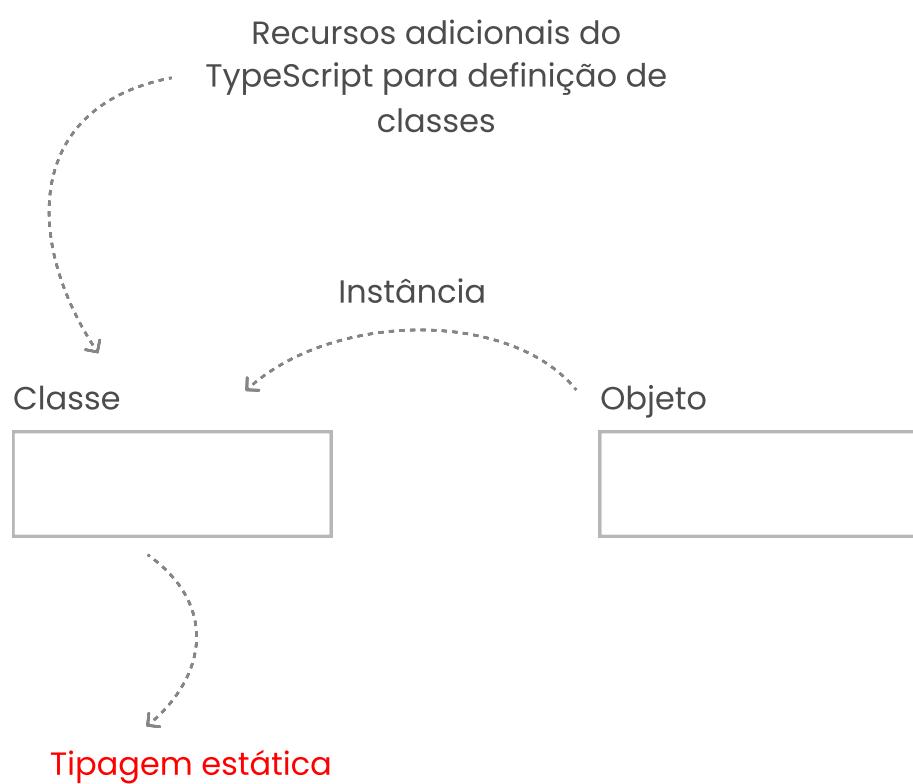
- ▶ TypeScript – Classes

Recursos Adicionais para Classes



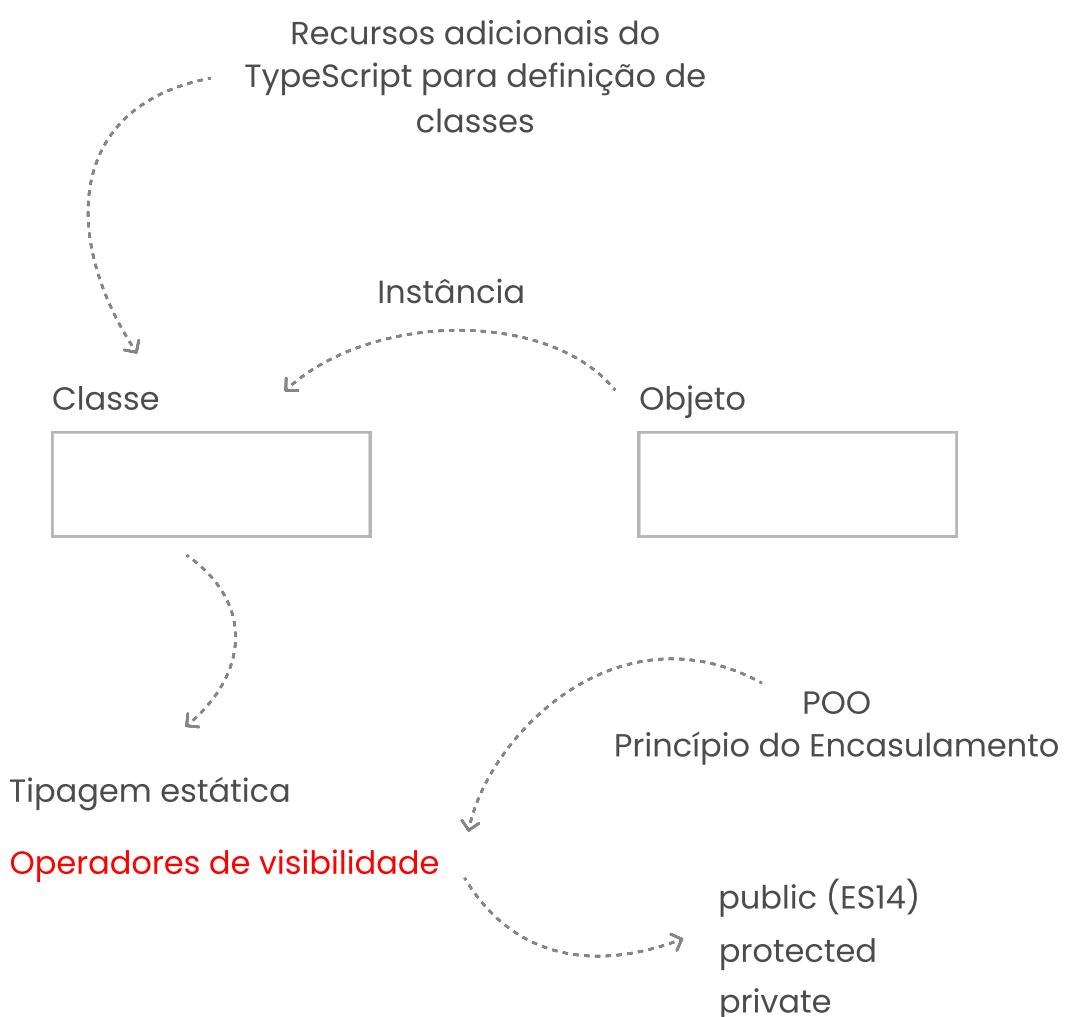
- ▶ TypeScript - Classes

Tipagem Estática



- ▶ TypeScript - Classes

Operadores de Visibilidade



- ▶ TypeScript - Classes

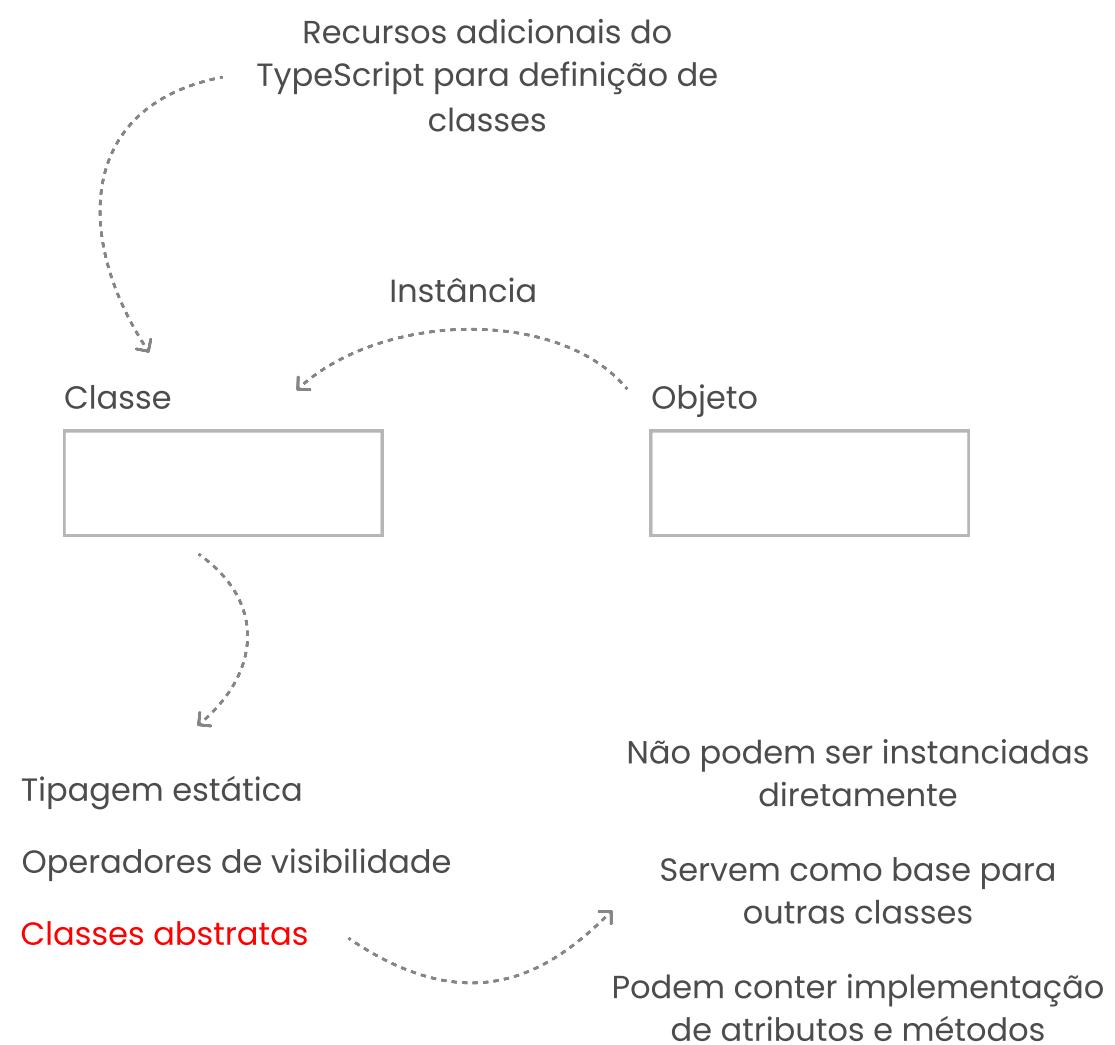
Operadores de Visibilidade Public, Protected e Private

Operadores de visibilidade	public (ES14)	protected	private
Acesso na Classe	✓	✓	✓
Acesso em Subclasses	✓	✓	✗
Acesso fora da Classe	✓	✗	✗

Com Instância (object)
Sem Instância (static)

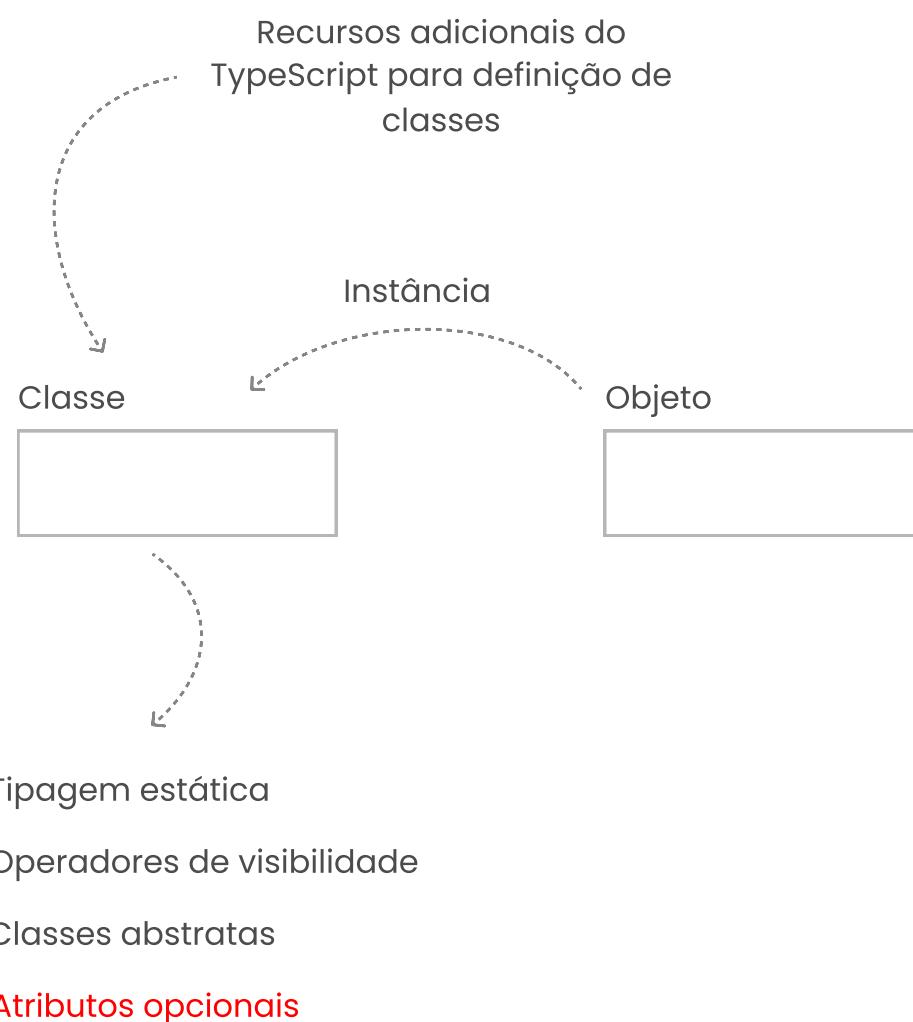
- ▶ TypeScript - Classes

Classes Abstratas



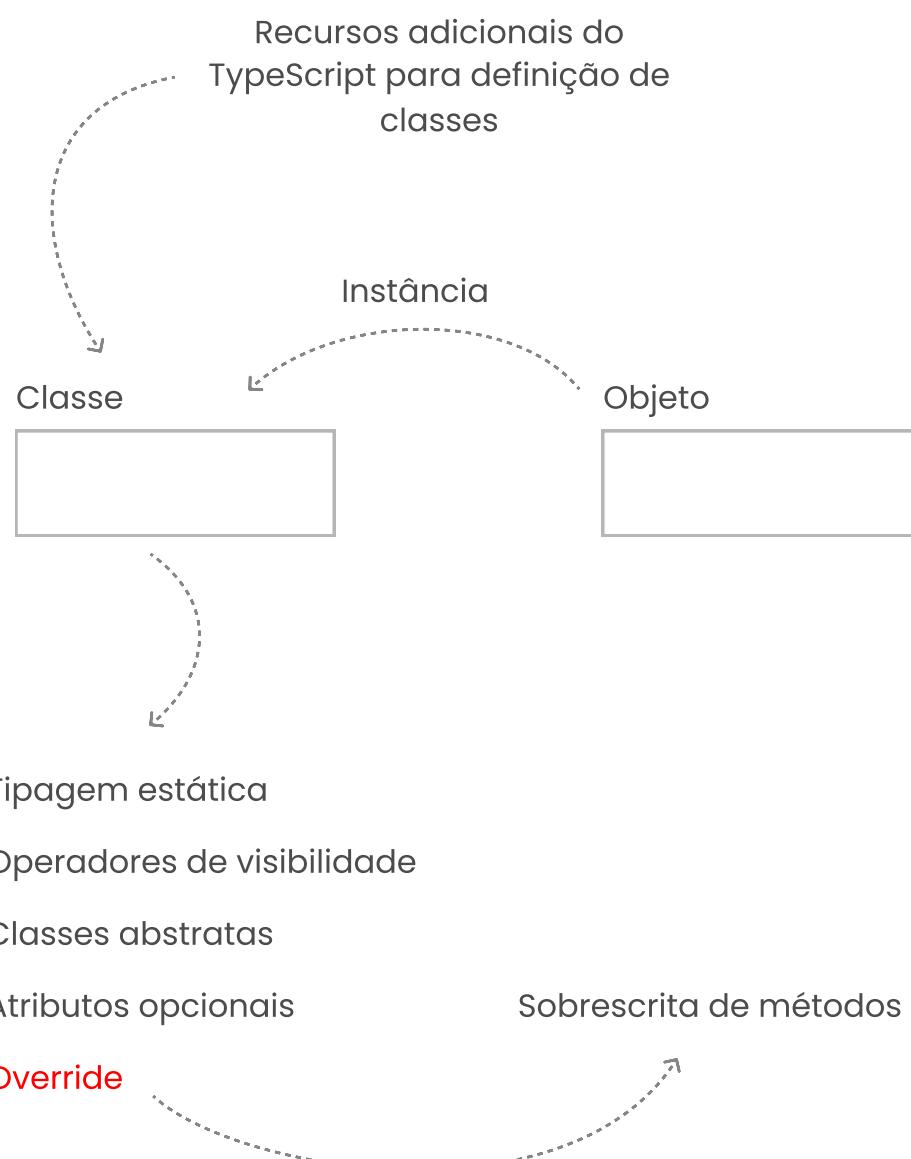
- ▶ TypeScript - Classes

Atributos Opcionais



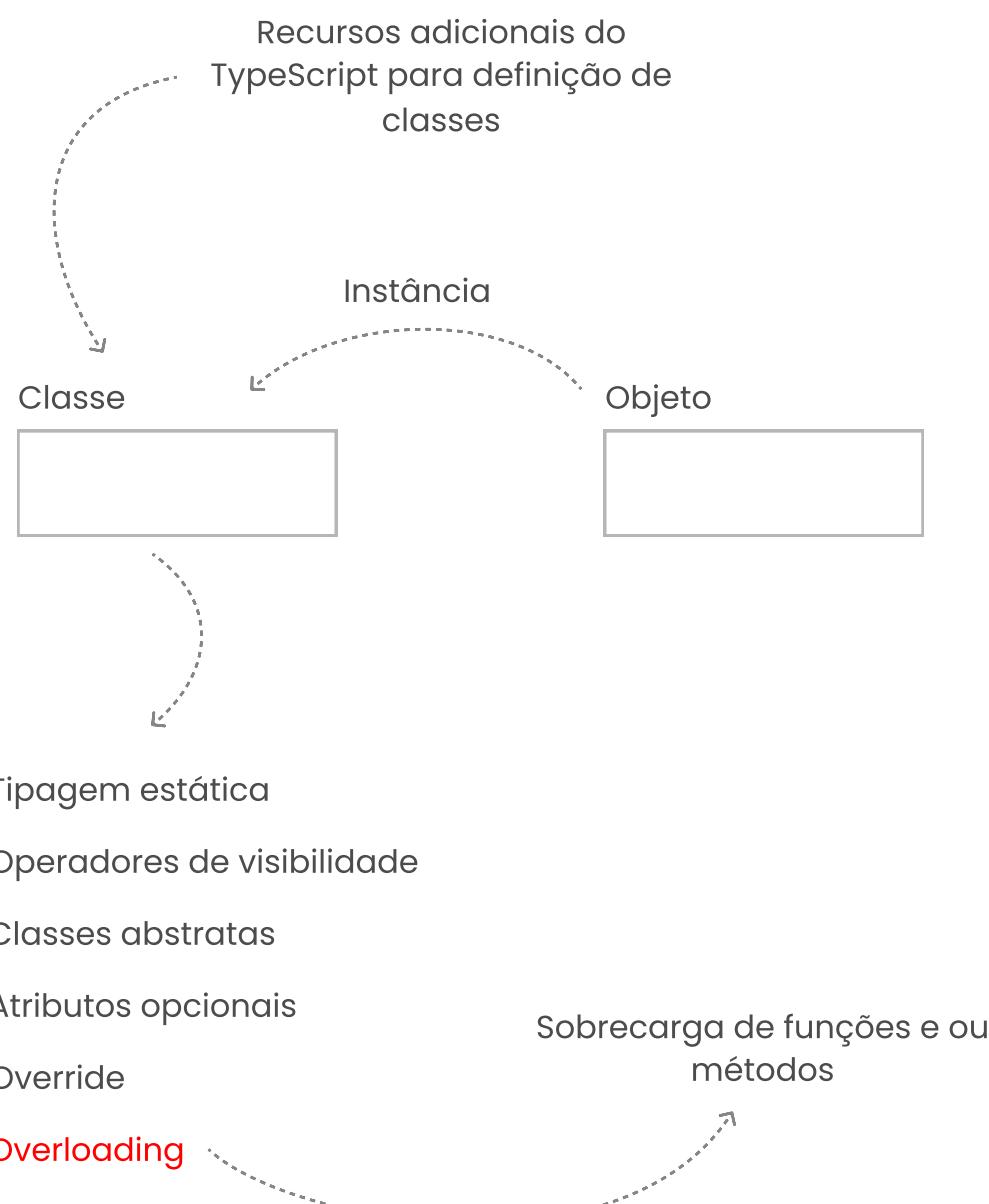
- ▶ TypeScript - Classes

Sobrescrita de Métodos (Override)



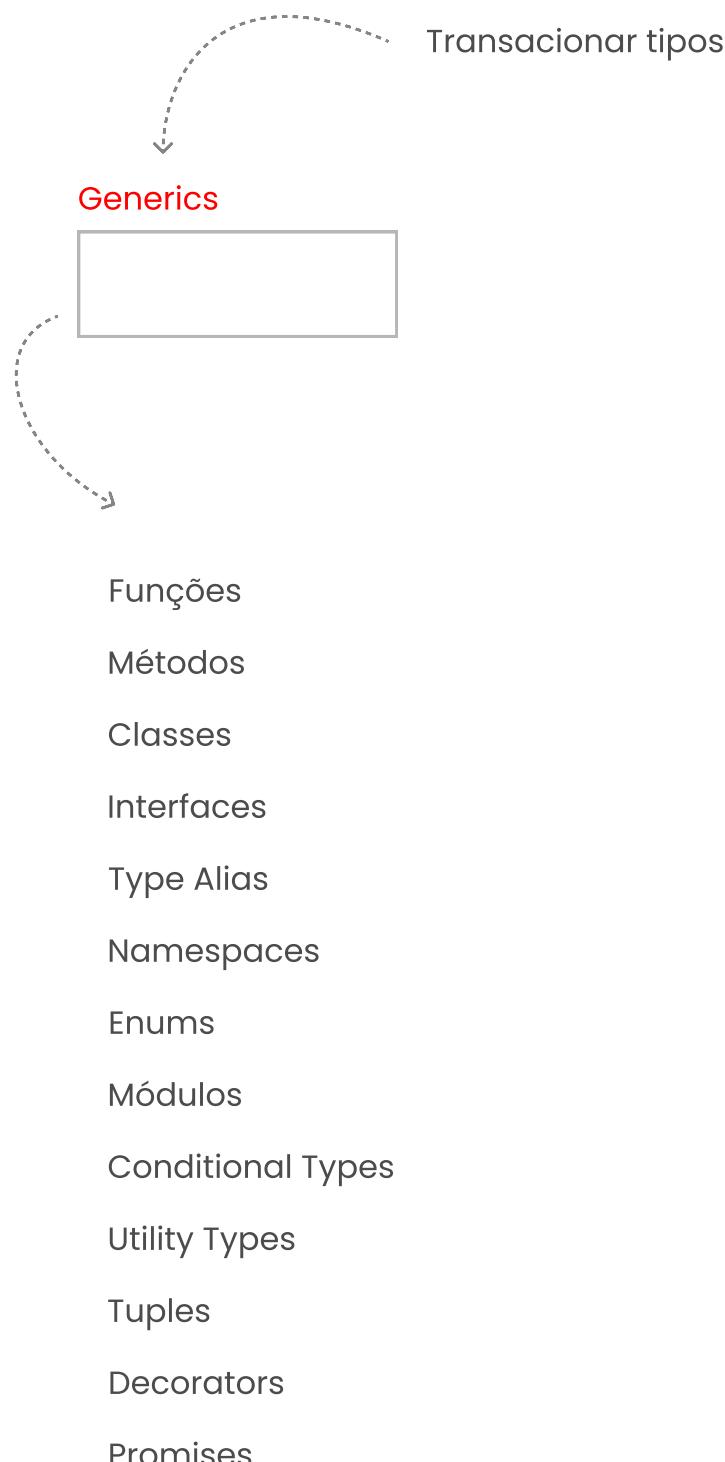
- ▶ TypeScript - Classes

Sobrecarga de Métodos (Overloading)



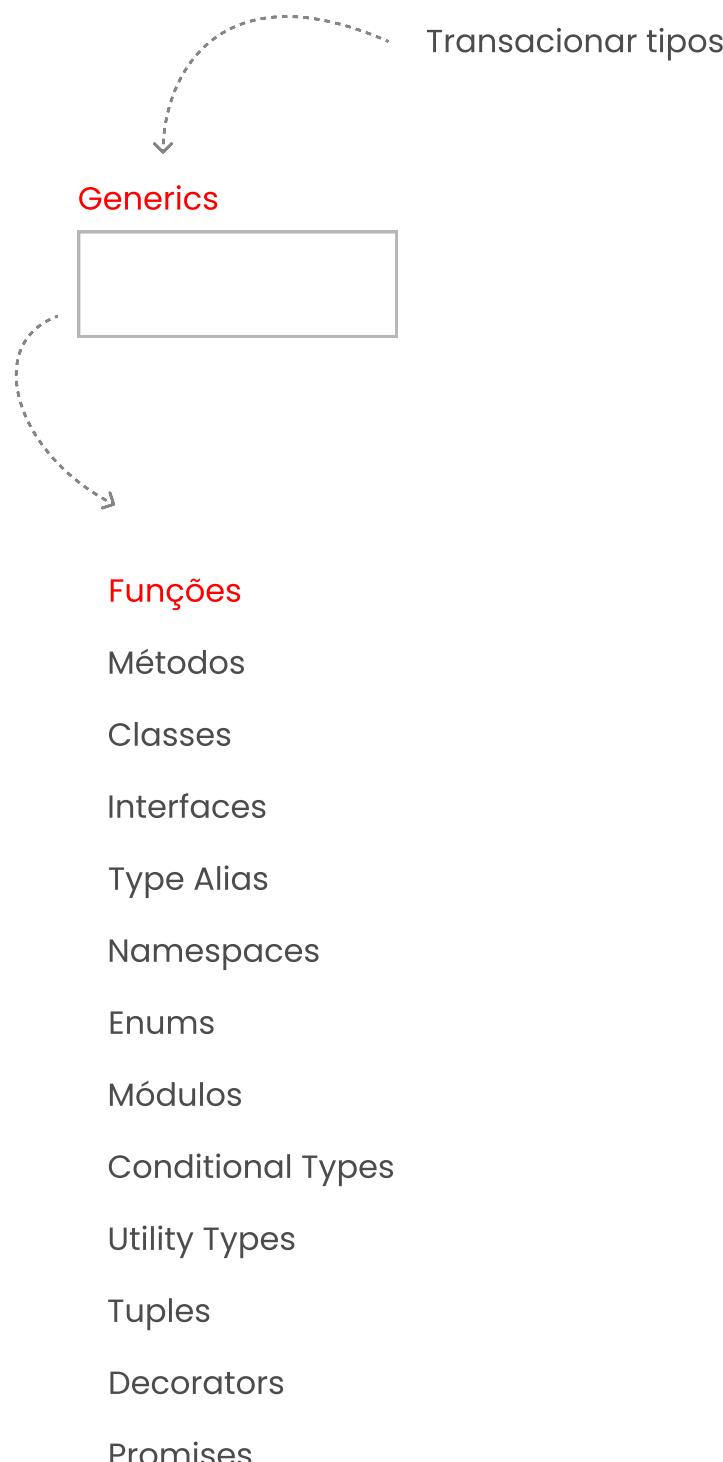
- ▶ TypeScript - Generics

Introdução aos Generics



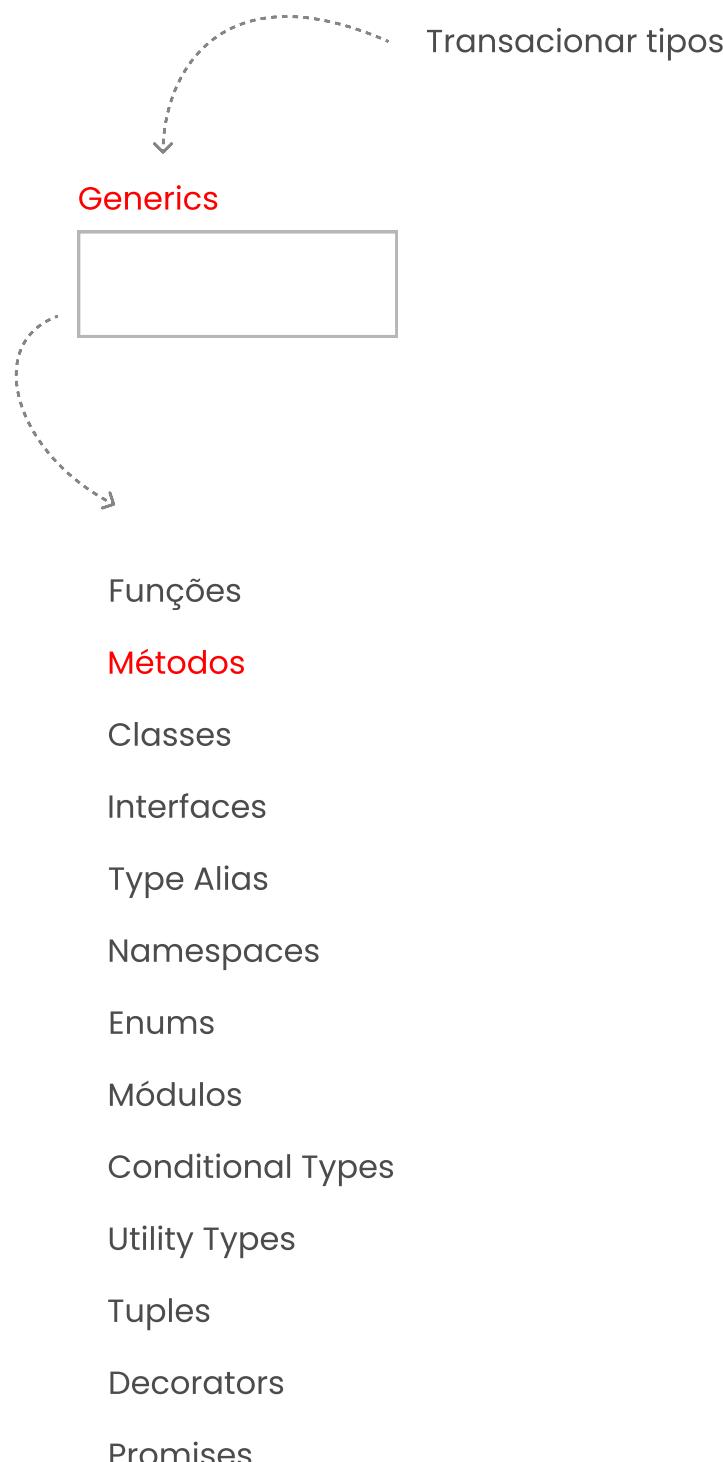
- ▶ TypeScript - Generics

Generics em Funções



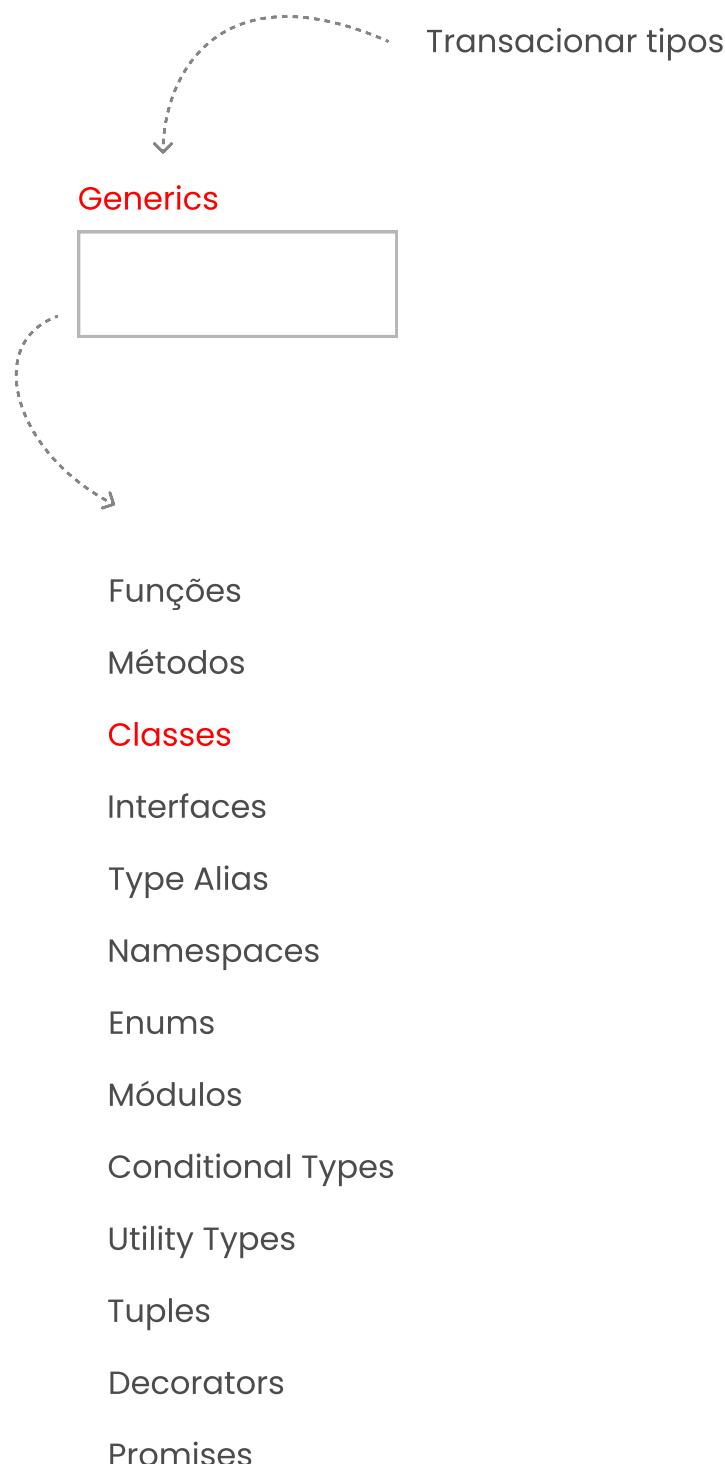
- ▶ TypeScript - Generics

Generics em Métodos



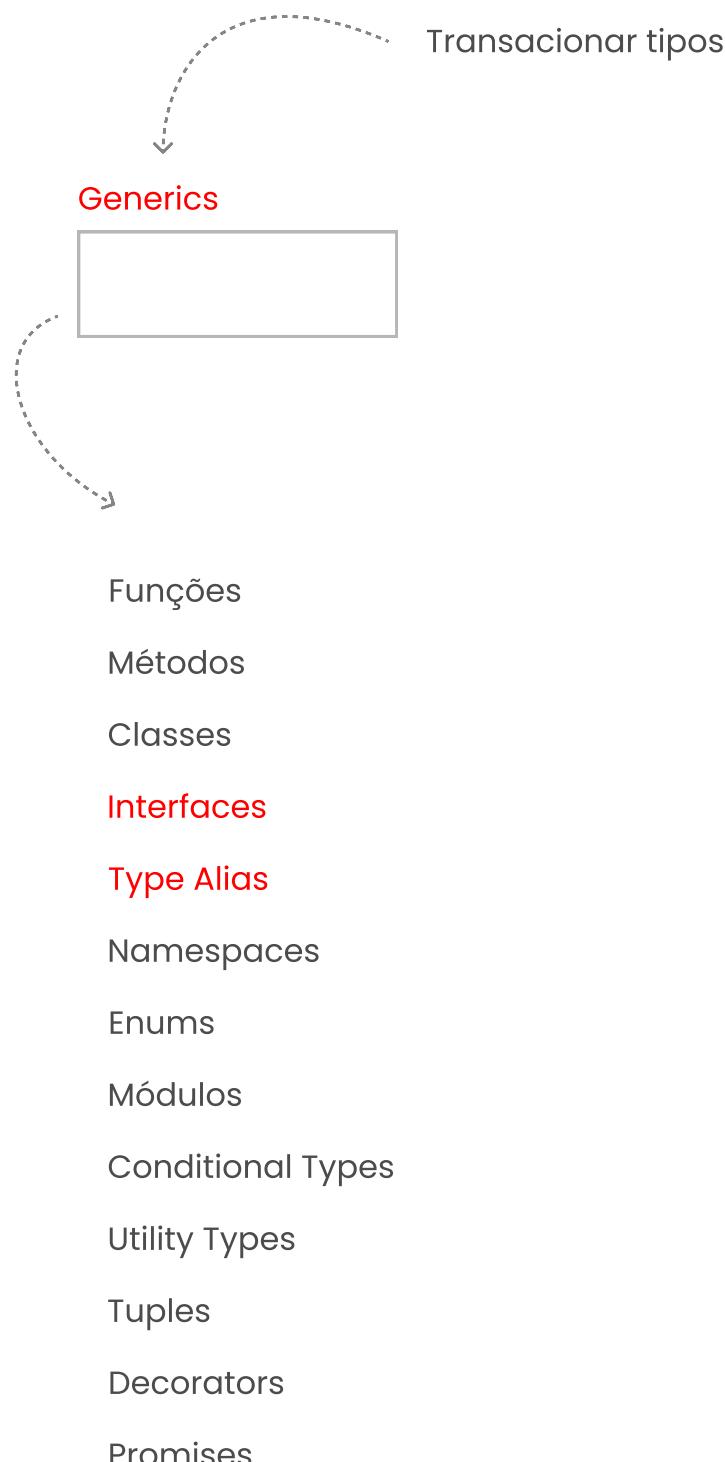
- TypeScript - Generics

Generics em Classes



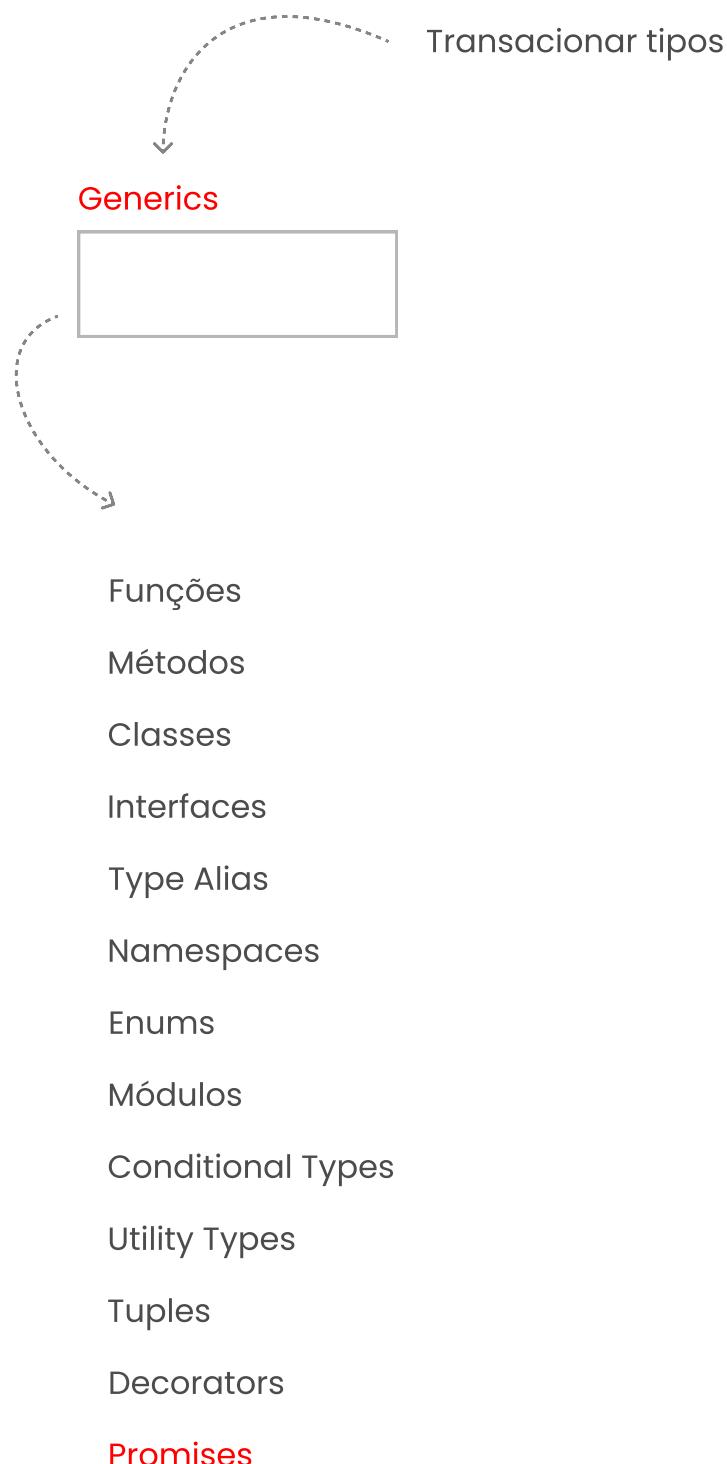
- ▶ TypeScript - Generics

Generics em Type Alias e Interfaces



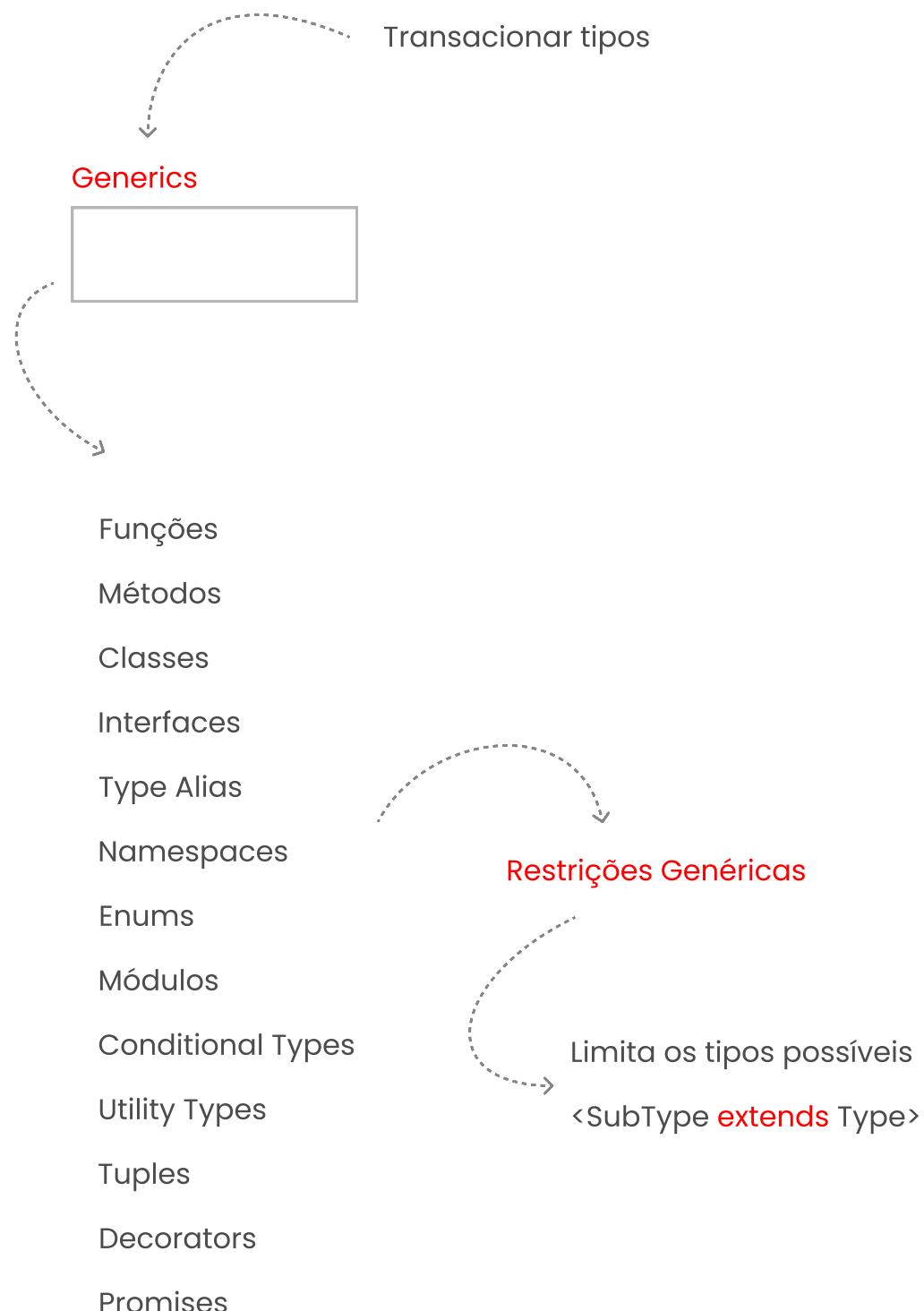
- ▶ TypeScript - Generics

Generics em Promises



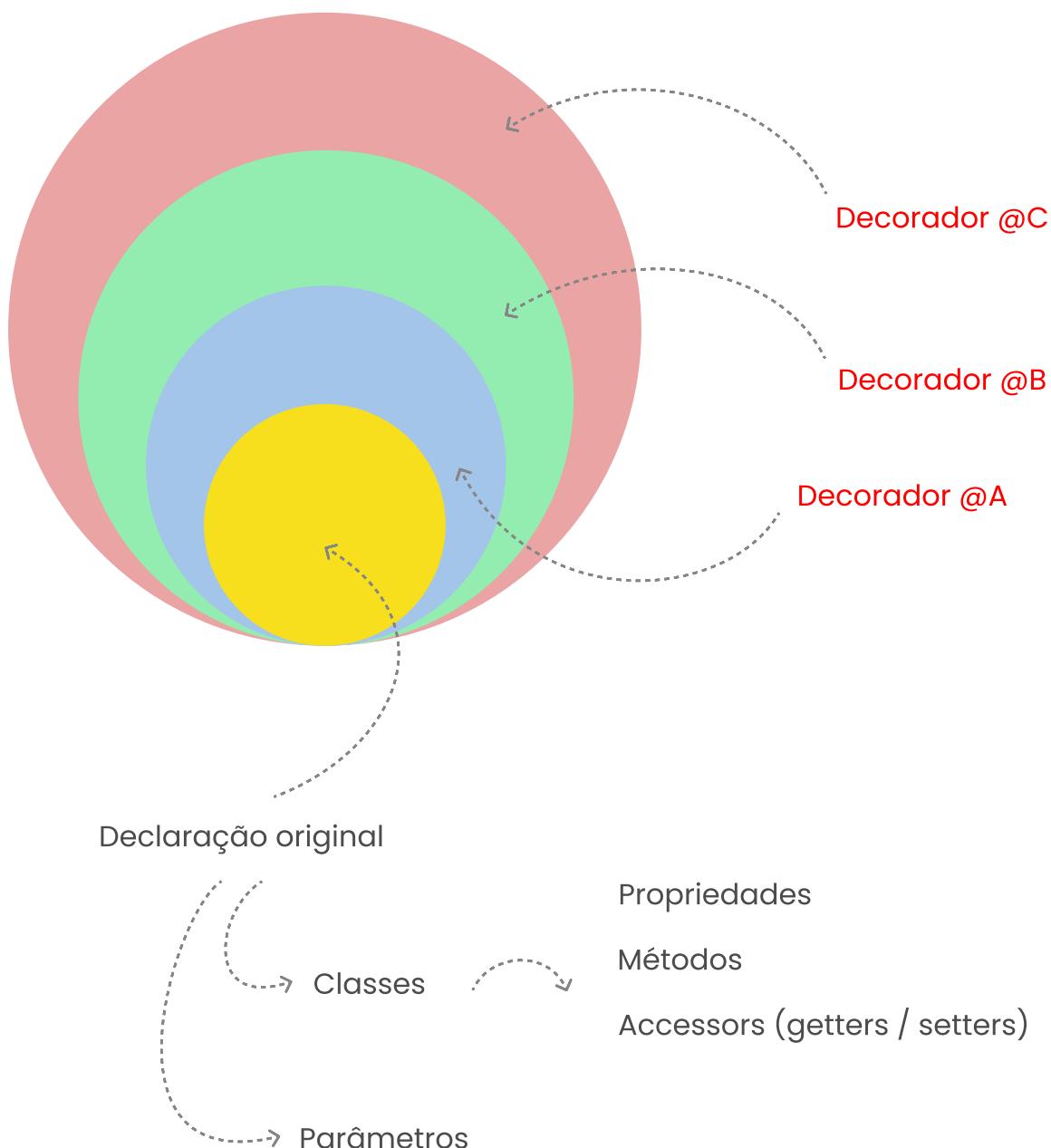
- ▶ TypeScript - Generics

Restrições Genéricas



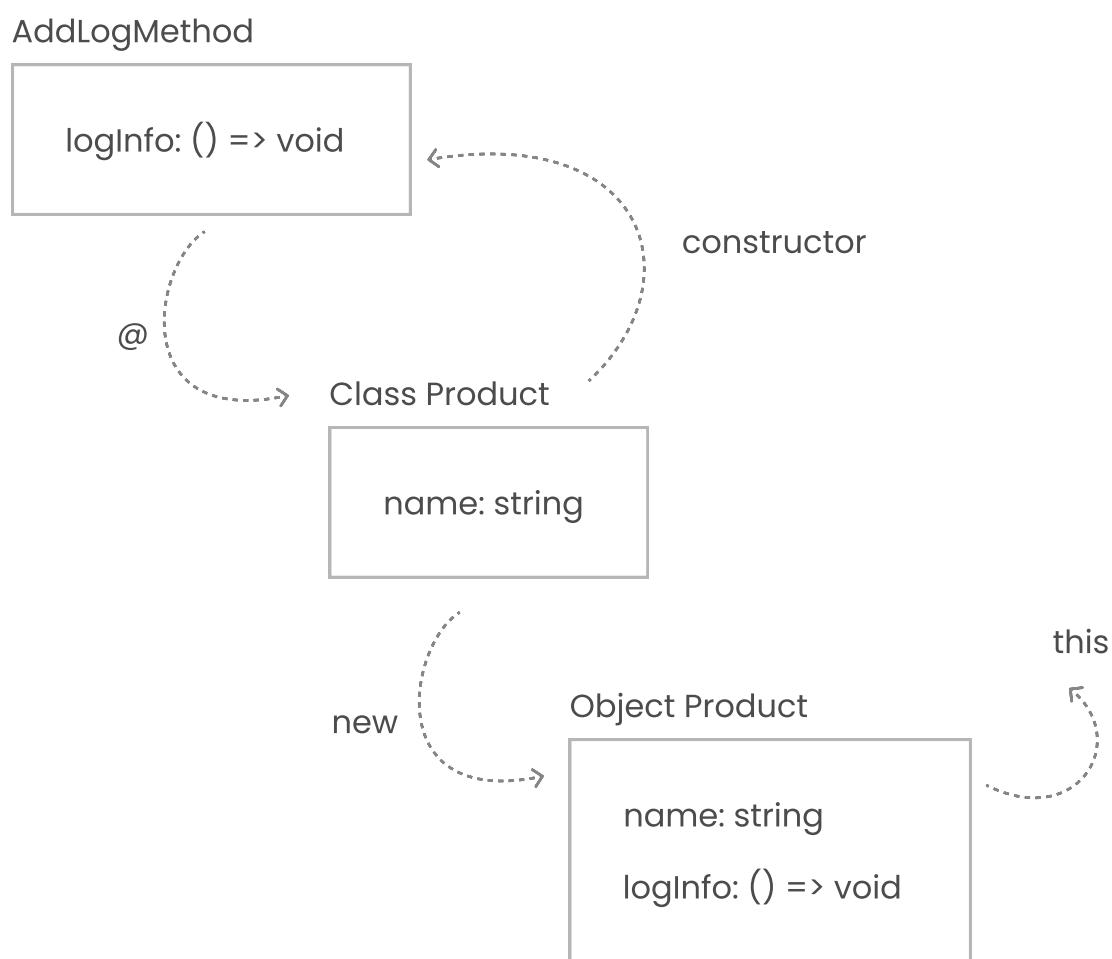
- ▶ TypeScript - Decorators

Introdução



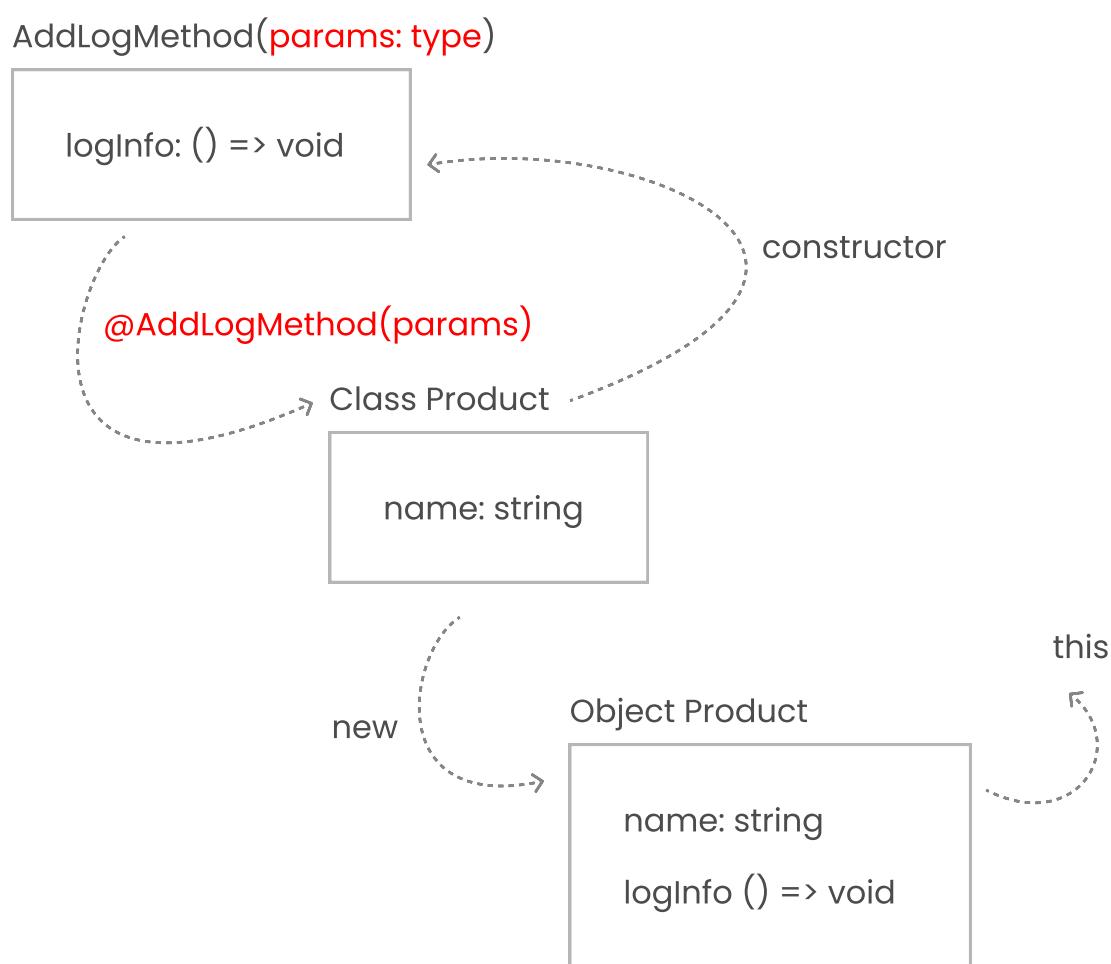
- ▶ TypeScript - Decorators

Decorando Classes (Class Decorator) - Parte 1



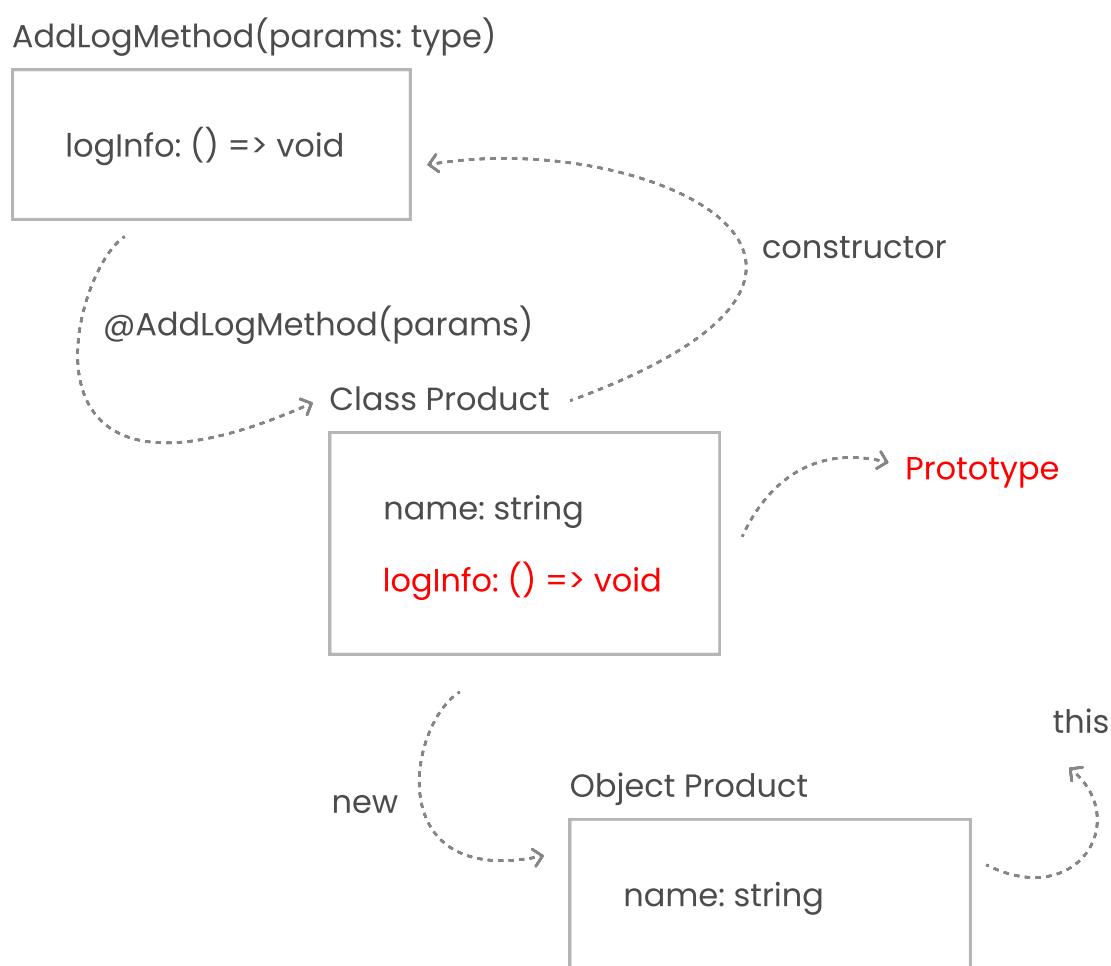
- ▶ TypeScript - Decorators

Encaminhando Parâmetros para Decoradores



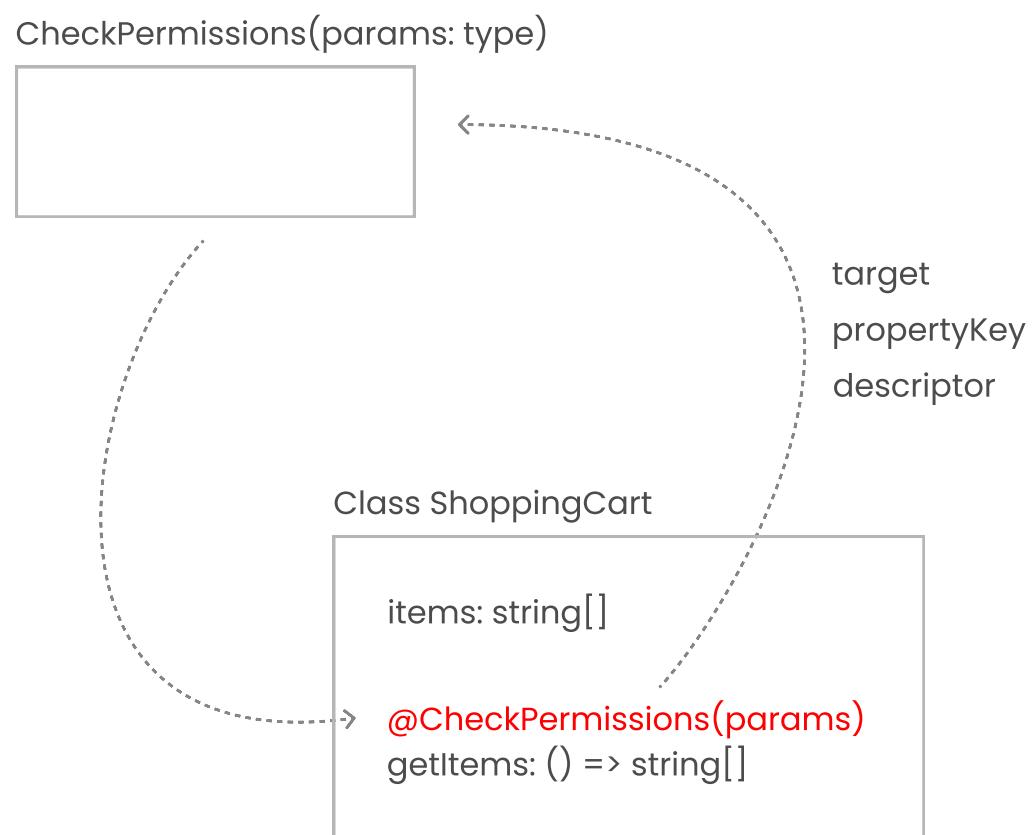
- ▶ TypeScript - Decorators

Decorando Classes (Class Decorator) - Parte 2



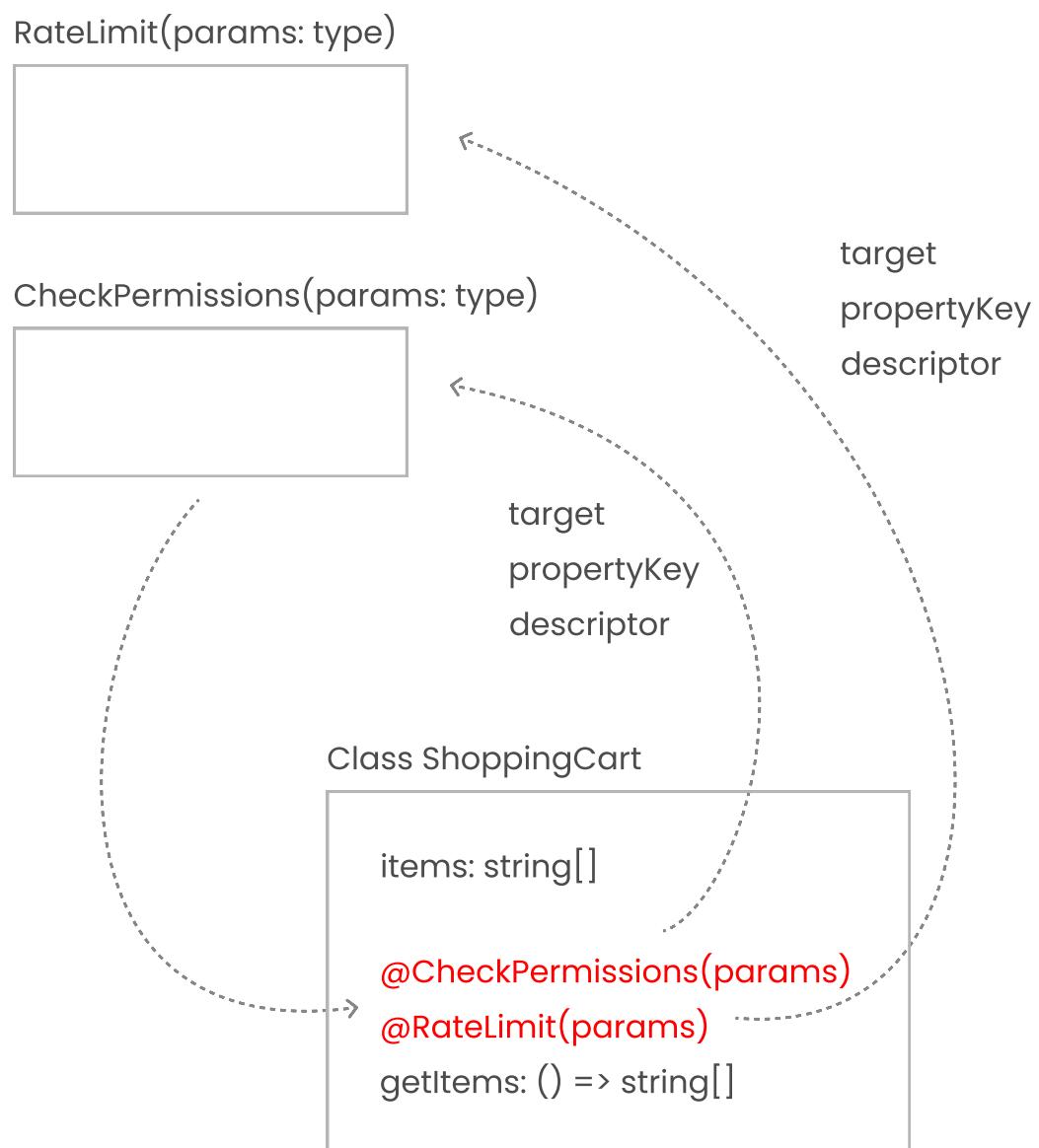
- ▶ TypeScript - Decorators

Decorando Métodos (Method Decorator)



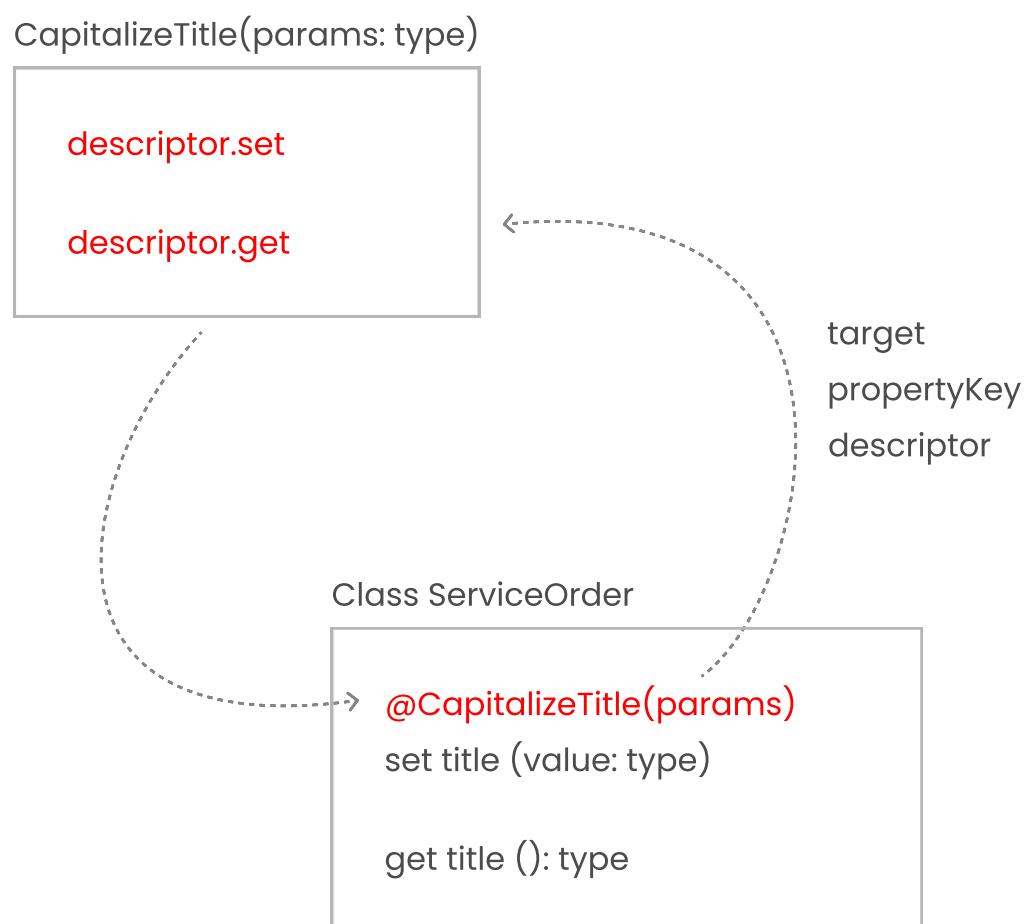
- TypeScript - Decorators

Múltiplos Decoradores



- ▶ TypeScript - Decorators

Decorando Métodos Assessores (Assessor Decorator)



- ▶ TypeScript - Decorators

Decorando Propriedades (Property Decorator)

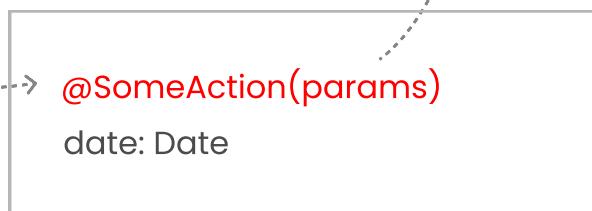
SomeAction(params: type)



←

target
propertyKey
descriptor X

Class Event



- ▶ TypeScript - Decorators

Decorando Parâmetros (Parameter Decorator)

AnotherAction(params: type)



target
propertyKey
parameterIndex
descriptor

Class Event

checkAvailability(@AnotherAction param: type)

ARGUS ACADEMY
www.argus-academy.com