



# Typescript

Guilherme Augusto Ferreira  
@guilheeeeeerme



# Ênfase

- Dar noções de typescript
- Recordar tópicos de POO
- Introduzir ferramentas
  - VS Code
  - NodeJS
  - Typescript
  - Angular

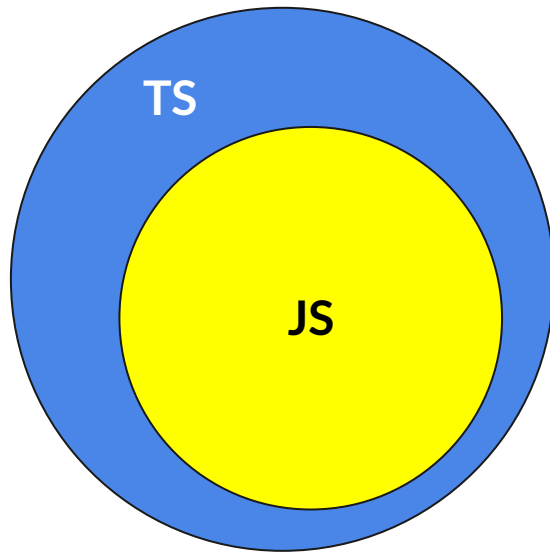


# O que é Typescript?

- TypeScript é um **superconjunto** de JavaScript desenvolvido pela Microsoft que **adiciona recursos a linguagem**
- É uma ferramenta de **desenvolvimento!**
  - O produto final ainda vai ser Javascript e vai rodar no **Node JS** ou no **Browser**

## A ideia de super conjunto

- Todo código Javascript é um código Typescript válido, mas *nem todo código Typescript* é um código Javascript válido.





## Exemplo

- Válido em ambos
- Válido somente no Typescript

```
function FalaOi(nome) {  
    return "Hello, " + nome;  
}  
  
// somente typescript  
function FalaOi(nome: string): string {  
    return "Hello, " + nome;  
}
```

# O que é Typescript?



Arquivos Typescript  
extensão \*.ts



Traduzir / Transpilar

Escolher Target



Arquivos de  
desenvolvimento

Produto final



## Mas qual é a vantagem?

- O Javascript é perigosamente flexível
- Muito parecido com o C#
- Facilita manutenção de projetos grandes
  - Case XPD
- É relativamente fácil migrar do Javascript para o Typescript



## Mas qual é a vantagem?

- Leva conceitos de POO para o Javascript e garante compatibilidade
  - es2015
- Facilita tipagem e intellisense
  - Igual o netbeans faz com o Java
  - As IDEs trabalham muito bem com typescript, principalmente o Visual Code





## O caso do angular

- O Angular 1 (ou AngularJS) foi feito em **Javascript**
- O **Angular2+** foi o AngularJS completamente reescrito com **Typescript** justamente para aproveitar a robustez e os novos recursos.
- É possível (apesar de difícil e não recomendado) converter uma aplicação construída com AngularJS para algo parecido com o Angular2+



# Instalando o TypeScript

`npm install -g typescript`





## O que precisa ter?

- NodeJS - <https://nodejs.org/en/>
- Abrir o terminal e garantir que os comandos **node** e **npm** estão funcionando
- Typescript instalado
  - `npm install -g typescript`
- [Opcional] Visual Studio Code
- [Opcional] http-server
  - `npm install -g http-server`

---

# Brincando no Browser



## Somente para fins didáticos

- Dificilmente isso vai acontecer na prática
- Criar os arquivos **index.ts** e **index.html**
- Vamos para o código



## index.ts

```
function FalaOi(nome) {  
    return "Hello, " + nome;  
}  
  
let nome = 'Fulano'  
  
document.body.textContent = FalaOi(nome);
```



## No terminal

```
tsc index.ts --target es2015
```

```
http-server .
```

*Para quem instalou o http-server*



## apareceu um index.js

```
function FalaOi(nome) {  
    return "Hello, " + nome;  
}  
let nome = 'Fulano';  
document.body.textContent = FalaOi(nome);
```





## É possível mudar o target

```
tsc index.ts --target es6
```



# index.html

```
<!DOCTYPE html>
<html>
  <head><title>TypeScript Greeter</title></head>
  <body>
    <script src="index.js"></script>
  </body>
</html>
```



# Explorando os recursos adicionados

- Tipagem
  - boolean, number, string, array, any, void
- Interface
- Classes
- Herança

---

Tudo de novo, só que no NodeJS



## index.ts

```
function FalaOi(nome) {  
    return "Hello, " + nome;  
}  
  
let nome = 'Fulano'  
  
console.log(FalaOi(nome));
```



## No terminal

```
tsc index.ts --target es5
```

```
node index.js
```



# Conhecendo a linguagem

- Tipagem
  - boolean, number, string, array, any, void
- Interface
- Classes
- Herança
- E múltiplos arquivos
  - imports e exports

---

O tsconfig.json





## tsc --init

- Vai gerar um arquivo chamado tsconfig.json
- O comando **tsc** busca o arquivo tsconfig.json
- Tudo que era passado como parametro pode ir no tsconfig.json
  - Logo, basta executar **tsc index.ts**
- **Outras configurações interessantes do tsconfig.json**
  - **include**
  - **outDir**



## sourceMap (debugger)

- Rodar `npm init -y`
  - Colocar o caminho do “main” com o js principal
- `Ctrl + Shift + P`
  - default build task
    - `tsc build`
- Aperte `F5`

---

# Implementando um design pattern



# Factory Pattern

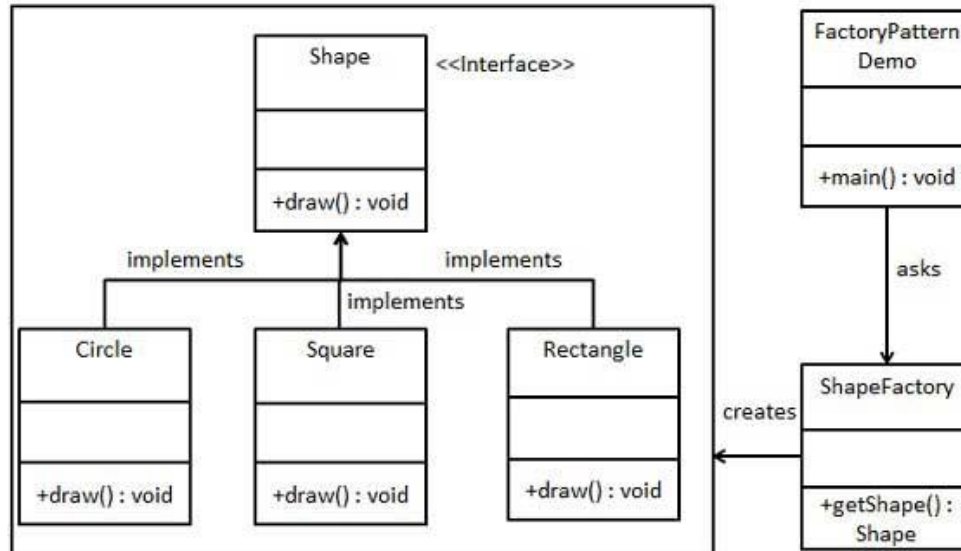
- A ideia do Factory evitar a replicação de código na criação de objetos
- Vamos implementar **Fábrica de Figuras Geométricas**
- Precisamos de um método que receba um *Enum* informando que figura quero criar
- O método deve retornar uma instância da figura geométrica inicializada com parâmetros aleatórios



# Requisitos

- Uma interface para definir o que uma figura geométrica pode fazer
  - Ex: Gira, Desenha
- Uma classe genérica para guardar atributos da figura
  - Ex: Nome, Cor
- Um enum para os tipos que você criar
- Uma classe Factory com uma método *geraFigura* com os atributos comuns
- Testar

# Factory Pattern UML



---

E se eu quiseser usar o Typescript no  
front end?

---

E se eu quiser usar o Typescript no front end? É mais ou menos o que o Angular faz !



---

Obrigado,

Github: @guilheeeeeeeerme