

Programação para Design

01 - Introdução ao TypeScript
Prof. Jefferson de Carvalho Silva

Introdução

- JavaScript(JS) foi criada como uma linguagem para executar do lado do cliente (navegador);
- O desenvolvimento do Node.js marcou o uso do JS como uma linguagem que também pode ser usada do lado do servidor;
- No entanto, JS falha ao tentar emular as funcionalidades das linguagens orientadas a objetos, checagem forte de tipos e erros de tempo de compilação;
- O TypeScript veio como uma linguagem para suprir essas “falhas”.

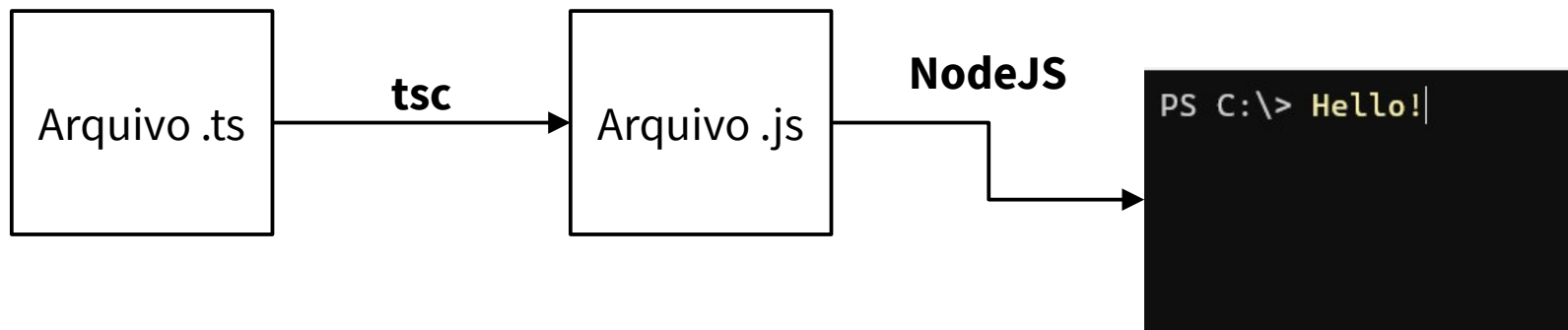
Mas o que seria o TypeScript?

- Por definição: “TypeScript é JavaScript para o desenvolvimento em escala de aplicações”.
- Características:
 - Fortemente tipada;
 - Totalmente orientada a objetos;
 - Linguagem compilada;
 - Projetada por Anders Heljsberg, o mesmo do C#, da Microsoft;
 - É considerada um **superconjunto** JavaScript!

Características do TypeScript

- **TypeScript também É JavaScript.**
 - Você só precisa saber um pouco de JavaScript para aprender TypeScript!
- **TypeScript suporta outras bibliotecas compatíveis com JavaScript.**
 - Código TS compilado na verdade é um código JS, logo...
- **É portátil entre navegadores , dispositivos e sistemas operacionais.**
- **ECMAScript 6**
 - Ou ECMAScript 2015 é a segunda maior revisão sobre a linguagem JS.
 - É especificação oficial do JS.
 - As funcionalidades de OO de TS obedecem essas padronizações e acrescentam mais algumas.
 - <https://262.ecma-international.org/6.0/>

Compilação do TypeScript (nosso caso)



Preparando o ambiente

- **Instalar o Node.js:**

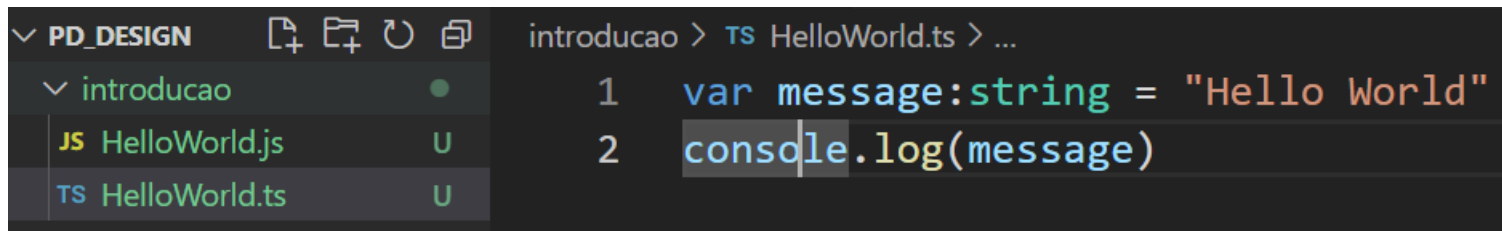
- <https://nodejs.org/en/download/>
- Verifique se instalou com sucesso executando o comando no terminal:
 - `node --version`

- **Instalar o compilador do TypeScript:**

- Execute o comando no terminal:
 - `npm install -g typescript`
 - `tsc --version`

Primeiro programa (fonte)

- Crie uma pasta para a disciplina e abra o VSCode na pasta;
- Crie uma pasta chamada “introducao” dentro da pasta anterior;
- Agora crie o arquivo HelloWorld.js dentro de “introducao”;
- No arquivo, escreva o seguinte código:



```
introducao > TS HelloWorld.ts > ...  
1  var message:string = "Hello World"  
2  console.log(message)
```

Primeiro programa (compilação)

- Salve o arquivo e abra o terminal na pasta do arquivo.
- Compile o código .ts com o seguinte comando:
 - `tsc HelloWorld.ts`
- Perceba que foi gerado o código “objeto” `HelloWorld.js`
- Execute o código JS com:
 - `node HelloWorld.js`

```
$ node HelloWorld.js  
Hello World
```


HelloWorld com OO

HelloWorldOO.ts

```
introducao > TS HelloWorldOO.ts > ...  
1  class HelloWorldOO {  
2      |    sayHello():void {  
3      |        console.log("Hello World OO!")  
4      |    }  
5  }  
6  
7  var hello = new HelloWorldOO();  
8  hello.sayHello()
```

Exercícios

1) Crie um programa .ts que mostre seus dados:

- Nome Completo
- Curso
- Universidade

2) Crie um programa com a mesma lógica do anterior só que agora “orientado a objetos”.

Sistema de Tipos

- O Sistema de Tipos engloba todos os tipos suportados pela linguagem;
- Ele também checa a validade dos tipos supridos como valores;
- O tipo “Any” → é o supertipo de todos os tipos em TypeScript
- Tipos “bult-in”
 - number (ponto flutuante de 64 bits)
 - string (sequencia de caracteres)
 - boolean (valores lógicos, true e false)
 - void → vazio
 - null (ausência intencional de um objeto)
 - undefined (valores não inicializados)

Sistema de Tipos

- **null vs undefined**

- Não são tipos de dados de variáveis e sim valores
- Uma variável inicializada com undefined quer dizer que não existe nenhum objeto ou valor atribuído a ela.
- Uma variável inicializada com null quer dizer que existe um objeto com valor undefined atribuído a ela.

- **Tipos definidos pelo usuário:**

- Enumerations (enums), classes, interfaces, arrays e tuplas.

Variáveis

- Declaração de variáveis

```
var nome:string = "Jefferson"  
var curso:string //undefined  
var universidade = "UFC" //inferido em tempo de execução  
var turno; //tipo any
```

Variáveis

- Declaração de variáveis

```
var nome:string = "John";  
var valor1:number = 50;  
var valor2:number = 42.50  
var soma = valor1 + valor2  
console.log("nome " + nome)  
console.log("first score: " + valor1)  
console.log("second score: " + valor2)  
console.log("sum of the scores: " + soma)
```

Variáveis

- Mudança de tipo de variável (Type Assertion)

```
var code:any = 1234  
var myCode = <number> code  
console.log(typeof(myCode))
```

Variáveis

- Mudança de tipo de variável (Type Assertion) – usando “as”

```
var code:any = 1234  
var myCode = code as number  
console.log(typeof(myCode))
```


Variáveis

- Inferência de tipos

```
var num = 2;    // nun é um number!  
console.log("value of num "+num);  
num = "12";  
console.log(num);
```

Variáveis

- Escopo Global, de Classe e Local

```
1  var global_num = 12           //global variable
2  class Numbers {
3      num_val = 13;             //class variable
4      static sval = 10;         //static field
5
6      storeNum(): void {
7          var local_num = 14;    //local variable
8      }
9  }
10 console.log("Global num: " + global_num)
11 console.log(Numbers.sval)      //static variable
12 var obj = new Numbers();
13 console.log("Global num: " + obj.num_val)
```

Operadores

- **Define uma função que sera executada em dados. Os dados são chamados de “operandos”.**
 - Operadores Aritméticos
 - Operadores Lógicos
 - Operadores Relacionais
 - Operadores Bitwise
 - Operadores de Atribuição
 - Operador Ternário
 - Operador String
 - Operador de Tipo

Referências

- <https://www.tutorialspoint.com/typescript/index.htm>