

# Linguagem de programação: TypeScript

## Resumo

---

### Referências Bibliográficas

- COSTA, Marcelo Soares da. **TypeScript – Gerando o JavaScript do futuro**. Rio de Janeiro: Ed. Ciência Moderna LTDA, 2021.
- ABREU, Luís. **Typescript. O Javascript Moderno Para Criação de Aplicações**. São Paulo: FCA, 1ª Ed., 2017.

### TypeScript

TypeScript é uma linguagem compilada fortemente tipada e orientada a objetos. Ela foi projetada por Anders Hejlsberg (designer de C#) na Microsoft. TypeScript é uma linguagem e um conjunto de ferramentas. TypeScript é um superconjunto tipado de JavaScript compilado para JavaScript. Em outras palavras, **TypeScript é JavaScript** mais alguns recursos adicionais.

### Recursos do TypeScript

TypeScript é apenas JavaScript. TypeScript começa com JavaScript e termina com JavaScript. O Typescript adota os blocos de construção básicos do seu programa a partir do JavaScript. Portanto, você só precisa saber JavaScript para usar o TypeScript. Todo o código TypeScript é convertido em seu equivalente em JavaScript para fins de execução.

TypeScript oferece suporte a outras bibliotecas JS. O TypeScript compilado pode ser resultado de qualquer código JavaScript. O JavaScript gerado pelo TypeScript pode reutilizar todas as estruturas, ferramentas e bibliotecas JavaScript existentes.

JavaScript é TypeScript. Isso significa que qualquer arquivo .js válido pode ser renomeado para .ts e compilado com outros arquivos TypeScript.

TypeScript é portátil. TypeScript é portátil em navegadores, dispositivos e sistemas operacionais. Ele pode ser executado em qualquer ambiente em que o JavaScript seja executado. Ao contrário de suas contrapartes, o TypeScript não precisa de uma VM dedicada ou de um ambiente de tempo de execução específico para ser executado.

### Por que usar o TypeScript?

#### Os benefícios do TypeScript incluem:

- **Compilação** - JavaScript é uma linguagem interpretada. Portanto, ele precisa ser executado para testar se é válido. Significa que você escreve todos os códigos apenas para não encontrar saída, caso haja um erro. Consequentemente, você terá que passar horas tentando encontrar bugs no código. O transpiler TypeScript fornece o recurso de verificação de erros. TypeScript irá compilar o código e gerar erros de compilação se encontrar algum tipo de erro de sintaxe. Isso ajuda a destacar os erros antes da execução do script.

- **Strong Static Typing** - JavaScript não é fortemente tipado. O TypeScript vem com um sistema opcional de digitação estática e inferência de tipos por meio do TLS (*TypeScript Language Service*). O tipo de uma variável, declarada sem tipo, pode ser inferido pelo TLS com base em seu valor.
- TypeScript oferece **suporte a definições de tipo** para bibliotecas JavaScript existentes. O arquivo de definição TypeScript (com extensão `.d.ts`) fornece definição para bibliotecas JavaScript externas. Consequentemente, o código TypeScript pode conter essas bibliotecas.
- TypeScript oferece **suporte a conceitos de Programação Orientada a Objetos**, como classes, interfaces, herança etc.

## Componentes do TypeScript

Em sua essência, o TypeScript tem os três componentes:

- **Linguagem** - Compreende a sintaxe, palavras-chave e anotações de tipo.
- O **compilador TypeScript** - O compilador TypeScript (`tsc`) converte as instruções escritas em TypeScript em seu equivalente em JavaScript.
- O **TypeScript Language Service** - O "*Language Service*" expõe uma camada adicional em torno do pipeline do compilador principal, que são aplicativos semelhantes a editores. O serviço de linguagem suporta o conjunto comum de operações típicas de um editor, como conclusão de instruções, ajuda de assinatura, formatação e delimitação de código, colorização etc.

## Arquivos de declaração

Quando um script TypeScript é compilado, há uma opção para gerar um **arquivo de declaração** (com a extensão `.d.ts`) que funciona como uma interface para os componentes no JavaScript compilado. O conceito de arquivos de declaração é análogo ao conceito de arquivos de cabeçalho encontrados em C/C ++. Os arquivos de declaração (arquivos com extensão `.d.ts`) fornecem intellisense para tipos, chamadas de função e suporte de variável para bibliotecas JavaScript como jQuery, MooTools etc.

## Sintaxe Básica

A sintaxe define um conjunto de regras para escrever programas. Cada especificação de linguagem define sua própria sintaxe. Um programa TypeScript é composto por:

- Módulos
- Funções
- Variáveis
- Declarações e Expressões
- Comentários

## Identificadores em TypeScript

Identificadores são nomes dados a elementos em um programa, como variáveis, funções etc. As regras para identificadores são:

- Os identificadores podem incluir caracteres e dígitos. No entanto, o identificador não pode começar com um dígito.
- Os identificadores não podem incluir símbolos especiais, exceto sublinhado (`_`) ou um cifrão (`$`).
- Os identificadores não podem ser palavras-chave.
- Eles devem ser únicos.
- Os identificadores são case sensitive a maiúsculas e minúsculas.
- Os identificadores não podem conter espaços.

## Declaração de variável em TypeScript

A sintaxe de tipo para declarar uma variável no TypeScript é incluir dois pontos (:) após o nome da variável, seguido por seu tipo. Assim como no JavaScript, usamos a palavra-chave **var** para declarar uma variável.

Ao declarar uma variável, você tem quatro opções:

- **Declare seu tipo e valor em uma instrução:**  
Var [identificador] : [declaração de tipo] = valor;
- **Declare seu tipo, mas nenhum valor. Nesse caso, a variável será definida como indefinida.**  
Var [identificador] : [declaração de tipo];
- **Declare seu valor, mas nenhum tipo. O tipo de variável será definido para o tipo de dados do valor atribuído.**  
Var [identificador] = valor;
- **Não declare nem o valor nem o tipo. Nesse caso, o tipo de dados da variável será qualquer um e será inicializado como indefinido.**  
Var [identificador];

### Exemplos:

- Var nome:string = "Ana"; a variável armazena um valor do tipo string.
- Var nome: string"; a variável é do tipo string e seu valor será indefinido (undefined) por padrão.
- Var nome = "Ana"; o tipo da variável será definido por seu valor, nesse caso a variável será do tipo string.

## Asserção de tipo em TypeScript

O TypeScript permite alterar uma variável de um tipo para outro. O TypeScript se refere a esse processo como **Asserção de Tipo**. A sintaxe é colocar o tipo de destino entre <> símbolos e colocá-lo na frente da variável ou expressão. O exemplo a seguir explica este conceito:

```
var estring = '1'  
var estring02:number = <number> <any> estring //estring passa a ser numérica  
console.log(typeof(estring02))
```

## Exercícios

---

1. Após estudar o que é o TypeScript, sua função e características, assinale o item correto abaixo:
  - a) É uma linguagem interpretada, fortemente tipada e orientada a eventos.
  - b) É um conjunto tipado de Java compilado para JavaScript.
  - c) É Java com vários recursos adicionais.
  - d) É convertido para Java para fins de execução.
  - e) Quando compilado pode ser resultado de qualquer código JavaScript.
  
2. Ainda sobre o que você aprendeu sobre o TypeScript, assinale o item correto abaixo:
  - a) Arquivos .js e .c válidos podem ser renomeados para .ts e compilados como TypeScript.
  - b) O TypeScript necessita de uma VM dedicada ou de um ambiente de tempo de execução específico para ser executado.
  - c) É portátil, podendo ser executado em vários navegadores, dispositivos e sistemas operacionais diferentes.
  - d) Assim como o JavaScript, o TypeScript é fortemente tipado.
  - e) Oferece suporte aos conceitos de programação orientada a eventos.
  
3. Sobre a formação dos identificadores em TypeScript, assinale o item correto:
  - a) Podem incluir dígitos, podendo inclusive iniciar por um dígito qualquer.
  - b) Podem incluir símbolos especiais de qualquer tipo.
  - c) Podem ser utilizadas palavras chaves como identificadores.
  - d) Não podem conter espaços em branco.
  - e) Não são case sensitive.
  
4. De acordo com o que você estudou sobre as formas de declaração de variáveis em TypeScript, assinale o item abaixo que exibe uma declaração correta de variável em TypeScript:
  - a) nome = Ana;
  - b) var nome : string = "Ana";
  - c) var = "Ana";
  - d) var nome : string == "Ana";
  - e) string nome = "Ana";

5. Observe a porção de código abaixo:

```
1 var nome:string = "Ana";
2 var valor1:number = 50;
3 var valor2:number = 42.50
4 var soma = valor1 + valor2
5 console.log("Nome"+nome)
6 console.log("Primeiro valor: "+valor1)
7 console.log("Segundo valor: "+valor2)
8 console.log("Soma dos valores: "+soma)
```

Assinale o item correto:

- a) A variável nome está declarada incorretamente.
- b) As variáveis valor1 e valor2 são de tipos diferentes.
- c) O conteúdo da variável soma é 92.50.
- d) O resultado da soma apresentada será de 92 pois o tipo da variável valor1 é inteiro.
- e) o resultado apresentado da variável soma será 92,50.

6. Observe a porção de código abaixo:

```
1 var i:number = 1
2 while(i<=10) {
3     if (i % 5 == 0) {
4         console.log ("O primeiro multiplo de 5 entre 1 e 10 é : "+i)
5         break
6     }
7     i++
8 }
```

Assinale o item correto:

- a) Serão exibidos os números de 1 a 10.
- b) Serão exibidos todos os múltiplos de 5.
- c) Serão exibidos apenas os números 5 e 10 como resultado.
- d) Será exibido o número 5 como resultado.
- e) O laço não será executado.

## Gabarito

---

- 1. E**  
A letra E é a alternativa correta, pois TypeScript torna qualquer código JavaScript executável após a sua compilação.
- 2. C**  
A letra C é a alternativa correta, pois uma das principais características do TypeScript é a sua portabilidade, independentemente de ambientes e sistemas.
- 3. D**  
A letra D é a alternativa correta, pois espaços em branco na formação dos identificadores são proibidos.
- 4. B**  
A letra B é a alternativa correta, pois a formação da sintaxe está utilizando a maneira certa de declaração e definição do tipo da variável.
- 5. C**  
A letra C é a alternativa correta, pois o conteúdo armazenado na variável soma é o resultado da adição das variáveis valor1 e valor2.
- 6. D**  
A letra D é a alternativa correta, pois pela lógica do teste no laço, apenas o primeiro múltiplo de 5 será exibido, o próprio número 5.