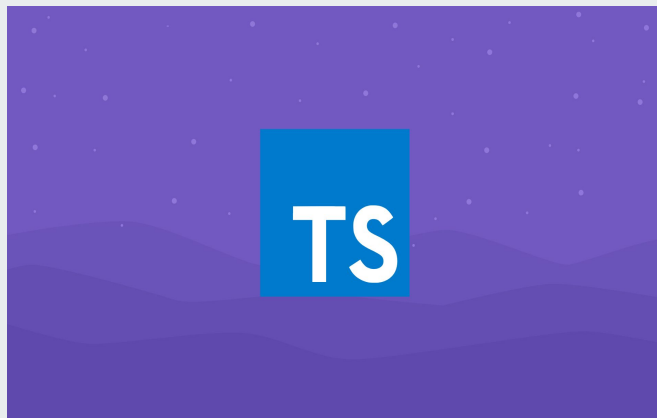


TypeScript

The TypeScript logo, which consists of a horizontal bar divided into two segments: a teal segment on the left and an orange segment on the right.

Lucas Alves da Costa
Pedro Rocha Pinto

História



É uma linguagem que complementa o JavaScript, tendo funcionalidades e recursos que não existem em JavaScript. A principal característica do TypeScript é a tipagem estática.

Ela foi desenvolvida pela Microsoft no fim de 2012, foi influenciada pelas linguagens: JavaScript, Java, C#.

Desenvolvimento



O TypeScript é uma linguagem multi-paradigmas. O paradigma usado com mais frequência é a orientação a objetos.

É uma linguagem de tipagem estática. Seu principal uso é para o desenvolvimento front end.

É executado em qualquer navegador e sistema operacional.

As principais IDEs que suportam TypeScript são Eclipse e Visual Studio Code. Alguns editores de texto também suportam a escrita do código, por exemplo: Vim, Atom e Emacs.

Avaliação



É uma linguagem de fácil aprendizagem para quem tem experiência com as linguagens que ela foi influenciada, C#, Java e JavaScript, pois não traz muitas inovações na sintaxe e semântica do código. Por ser de tipagem estática ela é costuma ser mais legível que JavaScript, mas em relação a outras linguagens como Python ela não é tão legível porque seus comandos têm nomes grandes e menos intuitivos como no Python. A manutenção do código é mais simples que no Javascript, pois alguns erros que ocorrem aparecem na compilação, enquanto que em JavaScript erros parecidos ocorreriam na execução.

Avaliação



Uma das funcionalidades do typescript são os tipos condicionais. Dessa forma, dependendo do valor atribuído a uma variável ela vai ter tipos distintos e isso pode ser feito de um modo simples. Essa funcionalidade também aplica-se às funções, por exemplo, uma função de ordenação, usando essa funcionalidade é possível ordena tanto strings e inteiros sem precisar criar uma função para cada tipo.

Avaliação



O TypeScript também possui o recurso da inferência de tipos, nem todas as linguagens tipadas estaticamente possuem esse recurso, ele também possui tipos genéricos, o tipo ***any*** e ***unknown***, que são tipos que recebem qualquer um dos tipos primitivos da linguagem.

Exemplos de códigos

```
interface Ipessoa{  
    nome: string;  
    idade: number;  
    email: string;  
}
```

```
function exe( Ipessoa ) { // Com interface  
    console.log("Olá %s você tem %d anos e seu email : %s",pessoa.nome,  
pessoa.idade, pessoa.email);  
}
```

```
var pessoa = { nome : "Nome" , idade : 22, email : "email@email.com"};  
exe(pessoa);
```

Exemplo de código



JavaScript

```
function exe(Ipessoa) {  
    console.log("Olá %s você tem %d anos e seu email : %s" ,  
    pessoa.nome, pessoa.idade, pessoa.email);  
}  
  
var pessoa = { nome: "Nome", idade: 22, email: "email@email.com" };  
exe(pessoa);
```

No exemplo acima os códigos executam a mesma coisa e possuem sintaxes semelhantes, mas o Typescript tem o recurso Interface que ajuda a entender como o objeto funciona, tornando o código mais fácil de ser entendido e também mais fácil.

Exemplo de códigos



```
var x,y : number | string;
x = 'Nm';
y = 6;
var z = somaouconc(x,y);
if (typeof z === 'number')
    console.log("%d",z);
else
    console.log("%s", z);
function somaouconc ( x: number | string, y : number | string): number|string{
    if (typeof x === 'number' && typeof y === 'number')
        return x+y;
    else if (typeof x === 'string' && typeof y === 'string')
        return x+y;
    else
        return "As variáveis não possuem o mesmo tipo"; }
```