



BACHARELADO EM SISTEMAS DE INFORMAÇÃO
CEFET-RJ NOVA FRIBURGO

Programação de Clientes Web

PROF. THIAGO DELGADO PINTO
thiago.pinto@cefet-rj.br

Introdução ao TypeScript

versão: 2020.11.24



Licença Creative Commons 4

checagem de tipos importa

```
const obj = { width: 10, height: 15 };  
const area = obj.width * obj.heighth;
```

Property 'heigh' does not exist on type '{ width: number; height: number; }'. Did you mean 'height'?

```
function greeter(person: string) {  
  return "Hello, " + person;  
}
```

```
let user = [0, 1, 2];
```

```
document.body.textContent = greeter(user);
```

Argument of type 'number[]' is not assignable to parameter of type 'string'.

TypeScript (TS)

linguagem de programação criada pela Microsoft

licença Apache 2

typescriptlang.org

é um **superconjunto** de JavaScript (JS)/EcmaScript (ES)

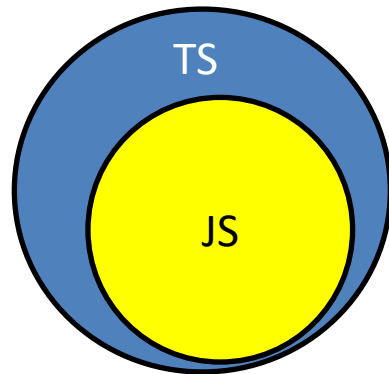
qualquer código JS funciona em TS

adiciona **tipagem estática opcional**

pode ser adotada incrementalmente

pode ser convertido para diferentes versões de JS (3, 5, 6, ...)

conversão é chamada de *transpilação*



TypeScript (TS)

projetada por **Anders Hejlsberg**

também é criador das linguagens C# e Roslyn

criador dos ambientes Delphi e Turbo Pascal

da Borland

criador do .NET Framework e Windows Foundation Classes

da Microsoft

criador do ambiente MS Visual J++



segue padrões e propostas da EcmaScript

muitas vezes as propostas do TS influenciam o ES

histórico resumido

v0.8 (2012) – primeira liberação
após 2 anos de desenvolvimento

v0.9 (2013) – adicionou suporte à *generics*

v1.0 (2014) – desempenho até 4x melhor
saiu do CodePlex para o GitHub

v2.0 (2016) – suporte a mais restrições de sintaxe
ex.: proibir *null*

v3.0 (2018) – suporte a tuplas, operadores de *rest* e *spread*

v4.0 (2020) – fábricas de JSX configuráveis, tuplas variáveis

- Há várias novidades em versões MAJOR e MINOR também.
- Costuma manter compatibilidade retroativa.
- Optou por não adotar versionamento semântico.

algumas funcionalidades

indicação, inferência e verificação de tipos

union types (`|`)

anotações (`@`)

classes – com encapsulamento, facilidades de construtor, *readonly*, etc.

interfaces

definição de tipos e *aliases*

tipos enumerados

tipos genéricos – com diversos operadores

vários tipos utilitários – `Record`, `Partial`, `Pick`, `ReturnType`, ...

namespaces

módulos

tuplas

arrow functions

parâmetros opcionais e *default*

async e *await*

documentação

site tem uma ótima documentação

<https://www.typescriptlang.org/docs>

possui livro (*handbook*) aberto e gratuito

[epub](#) e [pdf](#)

há inúmeros cursos e vídeos

tendência – Google Trends



Google Trends

Pesquisar



Todo o mundo ▼

09/09/2012 – 09/09/2020 ▼

Todas as categorias ▼

Pesquisa na Web ▼

Interesse ao longo do tempo ⓘ



Imagem: <https://trends.google.com/trends/explore?date=2012-09-09%202020-09-09&q=%2Fm%2F0n50hvxv>

tendência – Google Trends

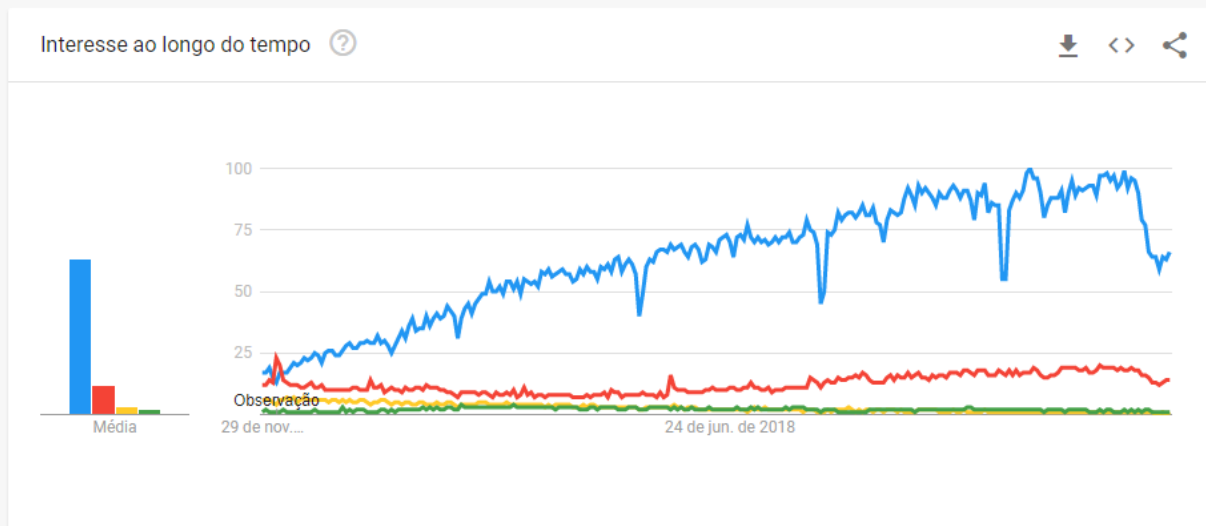
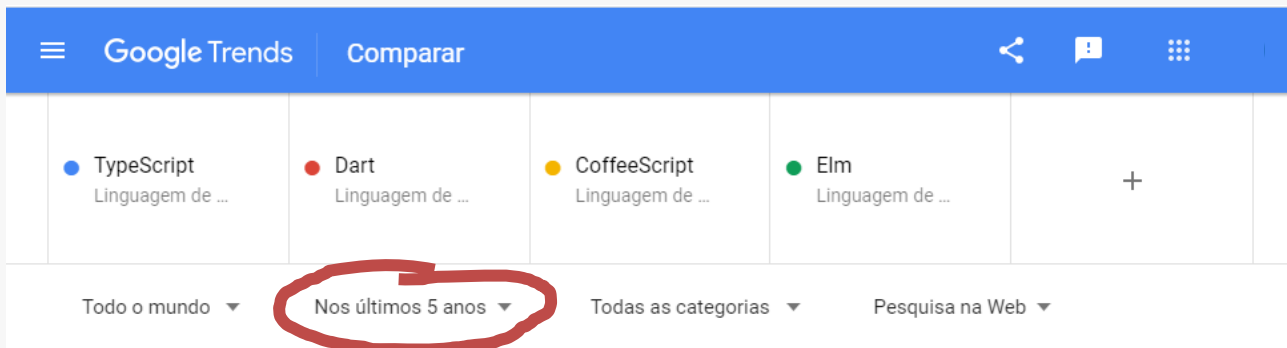


Imagem: <https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0n50hvx,%2Fm%2F0h52xrI,%2Fm%2F0hjc5m0,%2Fm%2F0ncc1sv>

satisfação – State of JS

JavaScript Flavors

Languages that compile to JavaScript

Rankings

Export

Share

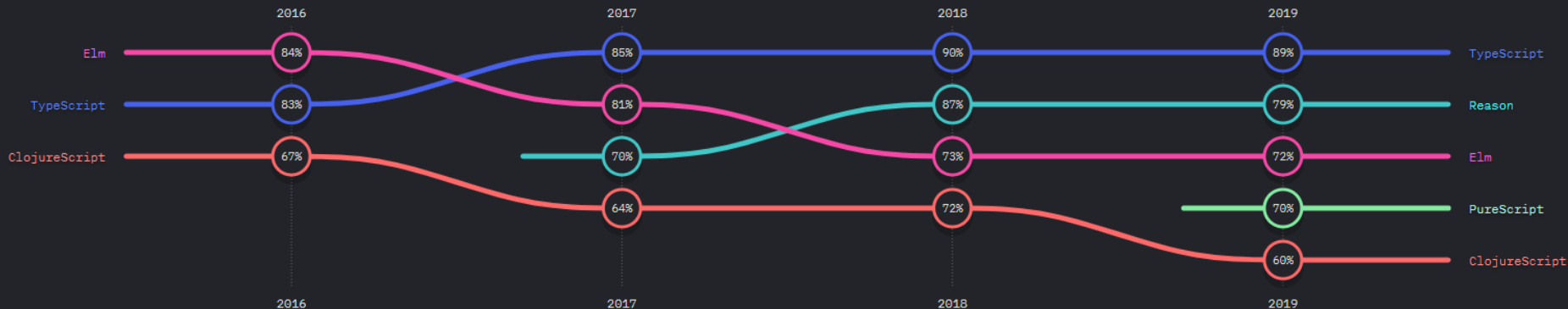


Satisfaction

Interest

Awareness

Awareness, interest, and satisfaction ratio rankings.



há várias **linguagens** que *transpilam* para JS

TypeScript

PureScript

CoffeeScript

Flow

Elm

Dart

até as mais "tradicionais"

C++, Java, Python, PHP, Go, Ruby, ...

veja <https://github.com/jashkenas/coffeescript/wiki/list-of-languages-that-compile-to-js>

instalação

NodeJS

`npm i -D typescript`

crie antes um arquivo `package.json` – com `npm init --yes`

DenoJS

nada a instalar – adota TypeScript por padrão

→ usaremos com [NodeJS](#) e [VSCode](#)

VSCode e o TypeScript

VSCode usa o motor do TypeScript para analisar código JavaScript

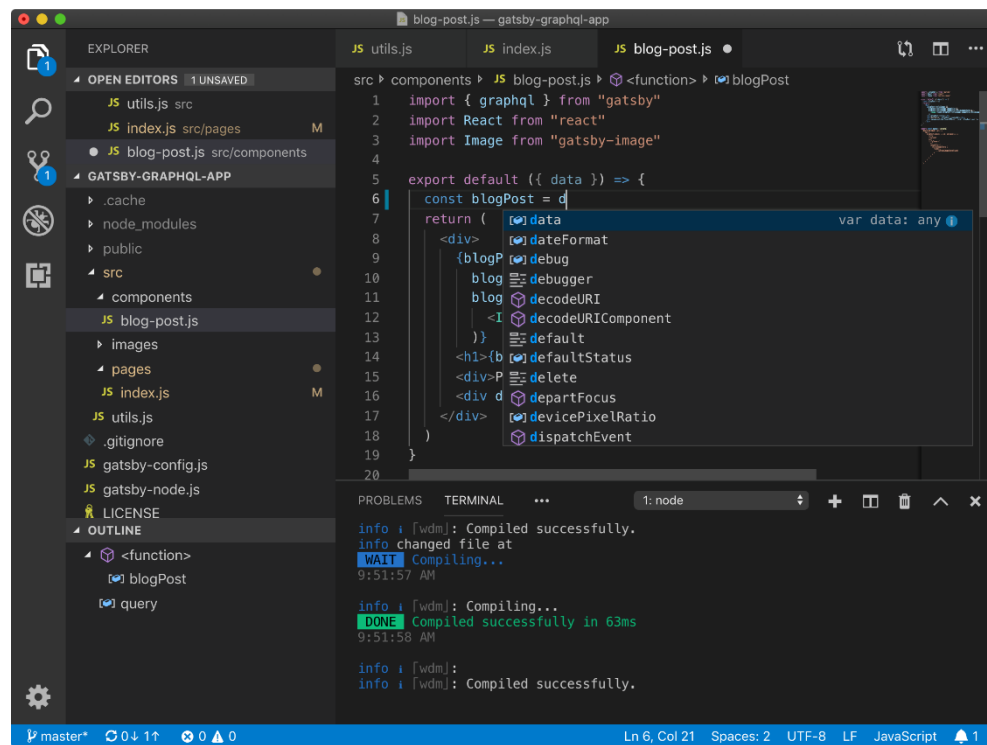
em qualquer arquivo **.js**

code completion

static checking

possui funções de refatoração

há diversos plug-ins para TS



transpilação de arquivo único

ES3 por *default*: `npx tsc exemplo.ts`

use `--target` ou `-t` para definir versão alvo:

ES5: `npx tsc exemplo.ts --target ES5`

ES 2015 (ES6): `npx tsc exemplo.ts -t ES2015`

...

ES 2020: `npx tsc exemplo.ts -t ES2020`

ES Next: `npx tsc exemplo.ts -t ESNext`

use `--outDir` para gerar para um certo diretório:

`npx tsc exemplo.ts -t ES2015 --outDir dist`

definindo arquivo de configuração

tsc procura por **tsconfig.json**

pode ser gerado com **npx tsc --init**

veja detalhes em <https://aka.ms/tsconfig.json>

exemplo

```
{
  "include": [
    "src"
  ],
  "compilerOptions": {
    "incremental": true,
    "target": "es6",
    "module": "umd",
    "lib": [
      "ES2015",
      "DOM"
    ],
    "outDir": "./public/js",
    "strict": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true
  }
}
```


referências

TypeScript Website. **TypeScript Handbook**. Disponível em: <https://www.typescriptlang.org/docs/handbook/intro.html>

Wikipedia. **TypeScript**. <https://en.wikipedia.org/wiki/TypeScript>

fim

2020.11.24: Inicial.



Licença Creative Commons 4

ESTE MATERIAL PERTENCE AO PROFESSOR THIAGO DELGADO PINTO
E ESTÁ DISPONÍVEL SOB A LICENÇA CREATIVE COMMONS VERSÃO 4.
AO SE BASEAR EM QUALQUER CONTEÚDO DELE, POR FAVOR, CITE-O.