Entendendo o Sistema de Tipos de TypeScript

Marcelo “Haskell” Camargo
github.com/haskellcamargo
marcelocamargo@linuxmail.org

Sumário

1) Introdução a TypeScript
2) Mergulhando no Sistema de Tipos

1 · Introdução a TypeScript

1) O que é TypeScript?
2) Quais as diferenças de TypeScript para JavaScript?
3) Quais as vantagens de TypeScript?

1 · 1 – O que é TypeScript?

- JavaScript como deveria ser
- Superset de ECMAScript
- Sistema de tipos decente
- Criado pela Microsoft
- 100% open-source

1 · 2 – Quais as diferenças de TS para JS?

- Sistema de tipos com checagem em tempo de compilação!

1 · 3 – Quais as vantagens de TypeScript?

- Coerência de tipos e de dados
- Captura de erros em tempo de compilação
- Features de ECMAScript 6 (código retrocompatível!)
- Linter com *code-completion*
- Sistema de módulos padrão
- Abstração e expressividade

1 · 3 · A – Coerência em tipos de dados

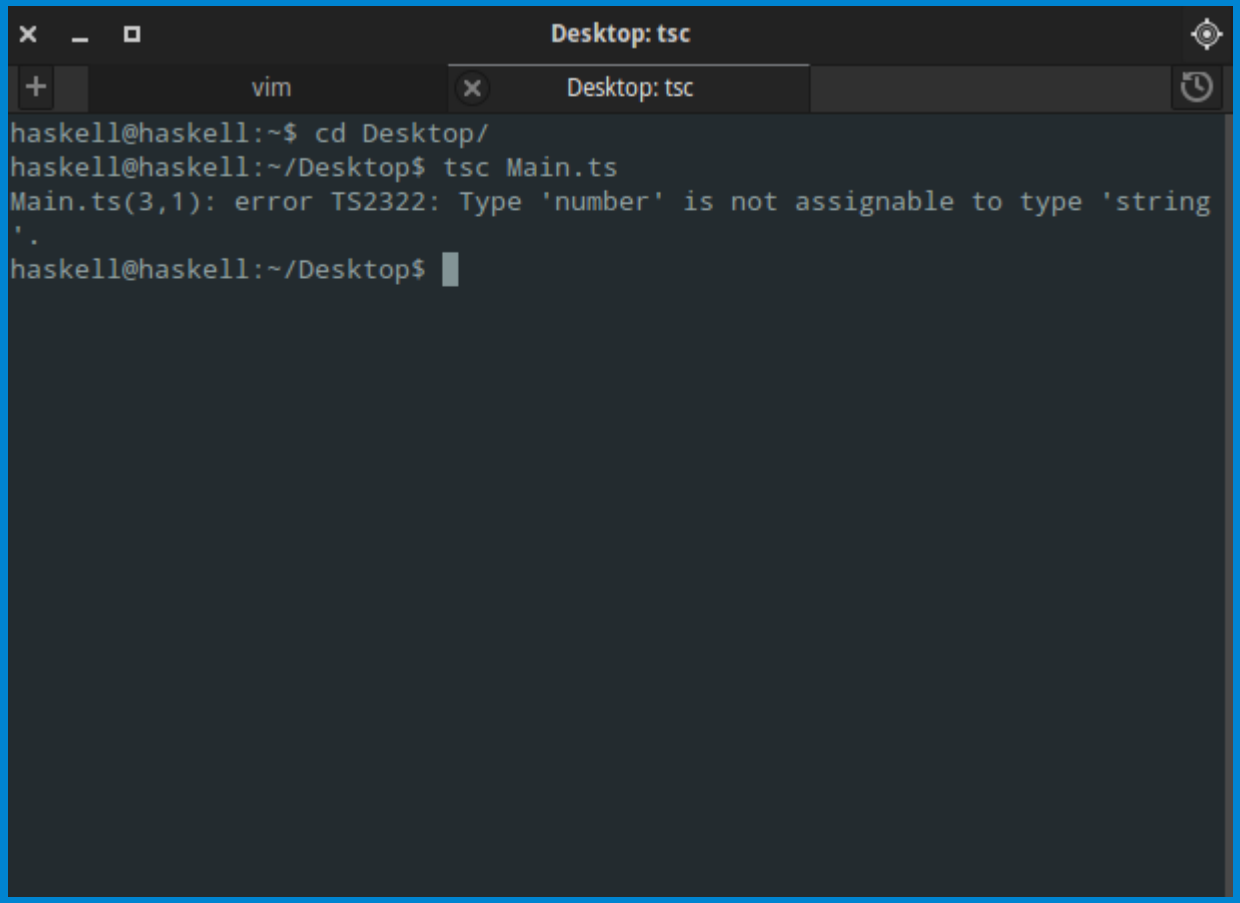
File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano

```
1
2 var x: string = "A";
3 x = 10;
4
```

Type 'number' is not assignable to type 'string'., WakaTime active 12:59 AM, Line 3, Column 2 Spaces: 2 TypeScript

1 · 3 · B — Capturas de erros em tempo de compilação



1 · 3 · C — Features de ECMAScript 6

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano

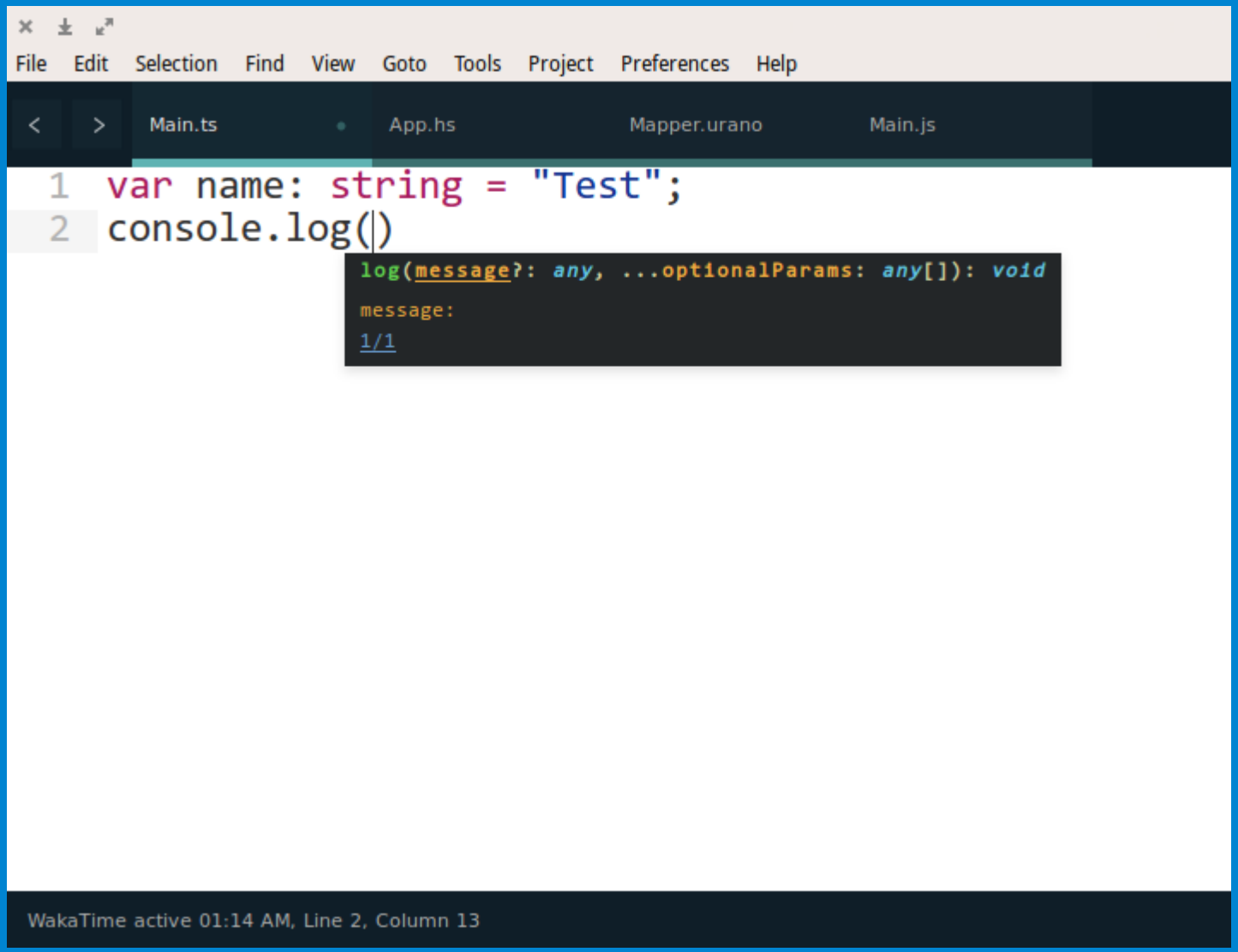
```typescript
class Animal {
  private specie: string;
  private habitat: string;

  constructor(
    specie: string,
    habitat: string
  ) {
    this.specie = specie;
    this.habitat = habitat;
  }
}
```

Main.js

```js
var Animal = (function () {
    function Animal(specie, habitat) {
        this.specie = specie;
        this.habitat = habitat;
    }
    return Animal;
})();
```

WakaTime active 01:11 AM, Line 7, Column 20 Spaces: 2 TypeScript

1 · 3 · D – Linter com *code-completion*



1 · 3 · E – Sistema de módulos padrão

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano

```ts
module Prelude {
  function map<T>(fn: (x: T) => T, xs: T[]): T[] {
    return xs.length
      ? map<T>(fn, [fn(xs[0])].concat(xs.slice(1)))
      : xs;
  }
}
```

Main.js

```js
var Prelude;
(function (Prelude) {
    function map(fn, xs) {
        return xs.length
            ? map(fn, [fn(xs[0])].concat(xs.slice(1)))
            : xs;
    }
})(Prelude || (Prelude = {}));
```

WakaTime active 01:21 AM, Line 8, Column 9 Spaces: 2 JavaScript

1 · 3 · F – Abstração e expressividade

WTF!?

2 · Mergulhando no Sistema de Tipos

1) Tipos básicos
2) Interfaces
3) Function types
4) Array types
5) Class types
6) Extensão de interfaces
7) Tipos híbridos
8) Type-aliases
9) Extensão de tipos nativos
10) Tipos genéricos
11) Misturando tipos

2 · 1 – Tipos básicos

Main.ts App.hs Mapper.urano Main

```
1  var isDone: boolean = false;
2  var height: number = 6;
3  var name: string = "Bob";
4  var ages: number[] = [1, 2, 3];
5  enum Status {
6    Play,
7    Pause
8  }
9  var yourBelly: any = "¯\_(ツ)_/¯";
10 var x: void; // Why!?
11
```

2 · 2 – Interfaces

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano

```
1  interface Person {
2    name: string;
3    age: number;
4  }
5
6  var marcelo: Person = {
7    name: "Marcelo",
8    age: 18
9  };
10
```

2 · 3 – Function types

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano Main.js

```
1
2
3
4 function add(x: number): (y: number) => number {
5   return (y: number) => {
6     return x + y;
7   }
8 }
9
```

2 · 4 – Array types

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano

```
1
2  interface Lang {
3    name: string;
4  }
5
6  var persons: Lang[] = [
7    { name: "Haskell" },
8    { name: "Java" },
9    { name: "Erlang" }
10 ];
11
```

2 · 5 – Class types

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano Main.js

```
1  interface IDog {
2    bark(): void;
3  }
4
5  class Dog implements IDog {
6    public bark(): void {
7      alert("Wow such code much nice!");
8    }
9  }
10
11 var dog: Dog = new Dog;
12 dog.bark();
13
```

2 · 6 – Extensão de interfaces

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano Main.js

```typescript
interface IAnimal {
  die(): void; // Sem retorno, entende!?
}

interface IDog extends IAnimal {
  bark(): void;
}

class Dog implements IDog {
  public bark(): void {
    alert("Wow such code much nice!");
  }

  public die(): void {
    alert("Queria estar morta!");
  }
}

var dog: Dog = new Dog;
dog.bark();
```

WakaTime active 02:20 AM, Line 15, Column 34 Spaces: 2 TypeScript

2 · 7 – Tipos híbridos

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano Main.js

```typescript
interface Counter {
  (start: number): string;
  interval: number;
  reset(): void;
}

var c: Counter;
c(10);
c.reset();
c.interval = 5.0;
```

WakaTime active 02:25 AM, Line 6, Column 1 Spaces: 2 TypeScript

2 · 8 – *Type-alias*

File Edit Selection Find View Goto Tools Project Preferences Help

Main.ts App.hs Mapper.urano Main.js

```typescript
type Sentence = string[];
var s: Sentence = [
  "Ah, como eu amo capivaras",
  "e de maneira proporcional...",
  "a como eu odeio Java!",
  "Porque a Oracle é bundona!"
];
```

WakaTime active 02:27 AM, Line 8, Column 1 Spaces: 2 TypeScript

2 · 9 – Extensão de tipos nativos

File Edit Selection Find View Goto Tools Project Preferences Help

App.hs Main.ts Mapper.urano Main.js

```
1  interface String {
2    withDot(): string;
3  }
4
5  String.prototype.withDot = function() {
6    return this + ".";
7  };
8
9  "Hello World!".withDot();
10
```

2 · 10 – Tipos genéricos

File Edit Selection Find View Goto Tools Project Preferences Help

App.hs Main.ts Mapper.urano Main.js

```typescript

function duplicate<T>(x: T): [T, T] {
  return [x, x];
}

function assertHTML<T extends HTMLElement>(x: T): void {}

var div: HTMLDivElement;
var svg: SVGElement;
assertHTML(div);
assertHTML(svg);

```

WakaTime active 02:36 AM, Line 6, Column 20 Spaces: 2 TypeScript

2 · 11 – Misturando tipos

File Edit Selection Find View Goto Tools Project Preferences Help

App.hs | Main.ts | Mapper.urano | Main.js

```typescript
type Literal = string
             | boolean
             | number
             ;
var t: Literal = 1;
t = "...";
t = true;
```

untitled

```typescript
interface X { a: string; }
interface Y { b: number; }

var xy: X & Y = {
  a: "...",
  b: 1
};
```

Line 7, Column 3 Spaces: 2 TypeScript

“*Capivaras são amigas, não comida*”

Perguntas?

2 · 11 – Misturando tipos

File Edit Selection Find View Goto Tools Project Preferences Help

App.hs | Main.ts | Mapper.urano | Main.js

```typescript
type Literal = string
             | boolean
             | number
             ;
var t: Literal = 1;
t = "...";
t = true;
```

untitled

```typescript
interface X { a: string; }
interface Y { b: number; }

var xy: X & Y = {
  a: "...",
  b: 1
};
```

Line 7, Column 3 Spaces: 2 TypeScript