TypeScript

XP-37

Foto por Todd Quackenbush @ Unsplash

Oi :)

@gabrieluizramos

gabrieluizramos.com.br

Gabriel Ramos

Desenvolvedor @ PlutoTV | Mentor @ Laboratória

Dedicando os últimos anos a trabalhar com React e Node, recentemente estudando mais afundo sobre testes em JavaScript e outros assuntos que envolvem a linguagem.
Fotógrafo por hobby e "o doido dos gatos" (só 3).

Agenda

Introdução (1h)

- O que é?
  - Linguagem?
  - Superset?
  - Onde vive?
  - Do que se alimenta?
  - Compilação
  - Deno
- Quais problemas o TS resolve?
- Quais problemas o TS traz?
- Devo sair tipando tudo?
- O que é inferência?
- Como instalar?

Tipos básicos (1h)

- String
- Number
- Boolean
- Array
- Tupla
- Enum
- Any
- Unknown
- Null e Undefined
- Object

Mais tipos (1h)

- Void
- Never
- Union types
- Intersection types
- Type alias
  - Valores nulos e opcionais
- Type assertion

Um pouco de OO (1h)

- OO
  - Classes
  - Interfaces (e semelhanças com type alias)
  - Herança
  - Modificadores de acesso
- Configurações extras do compilador

Foto por Curtis MacNewton @ Unsplash

Material

Nossa dinâmica vai ser bem "viva" e esses slides servem mais para referência de conteúdo e para que possamos nos guiar.

Todos os exemplos e materiais discutidos aqui estarão disponíveis em um mini-projeto que você pode acessar através do link bit.ly/xp-37-ts, repositório onde encontrará um resumo de tudo que vamos falar.

Mesmo assim, não esqueça de fazer suas anotações também! Caso precise de qualquer coisa, meu usuário é @gabrieluizramos em todas as redes sociais e podemos bater um papo.

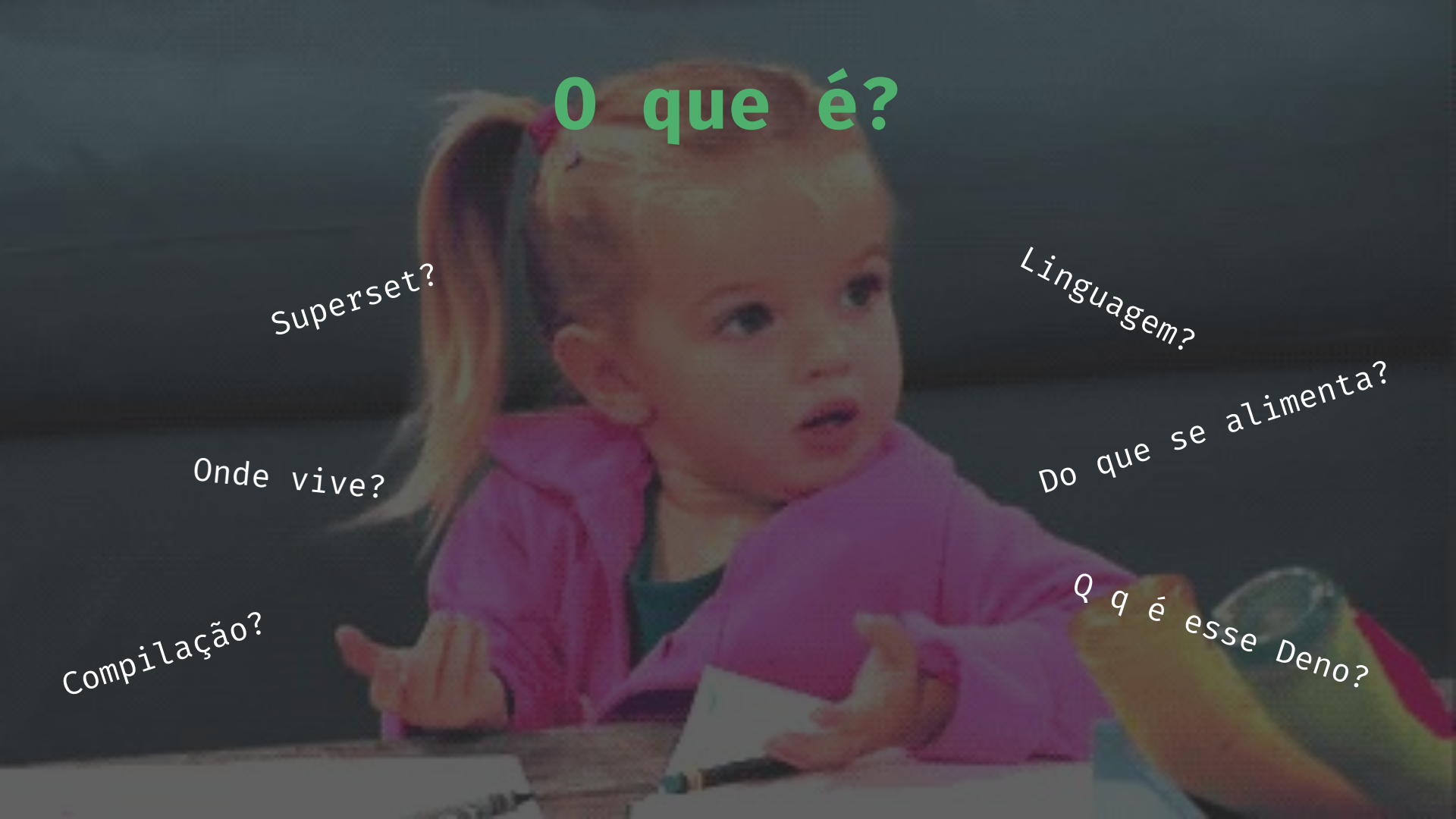
Mesmo assim ...

Não esqueça de fazer suas anotações também!

Caso precise de qualquer coisa, meu usuário é @gabrieluizramos em todas as redes sociais e podemos bater um papo.

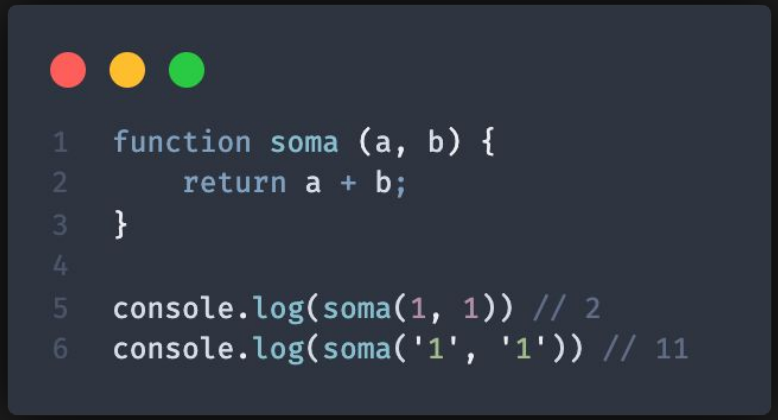
Bora lá!

O que é?


Superset?
Linguagem?
Onde vive?
Do que se alimenta?
Compilação?
Q q é esse Deno?

Quais problemas o TS resolve?

Um exemplo bem curto e fácil de lembrar é esse



```
1  function soma (a, b) {
2      return a + b;
3  }
4
5  console.log(soma(1, 1)) // 2
6  console.log(soma('1', '1')) // 11
```

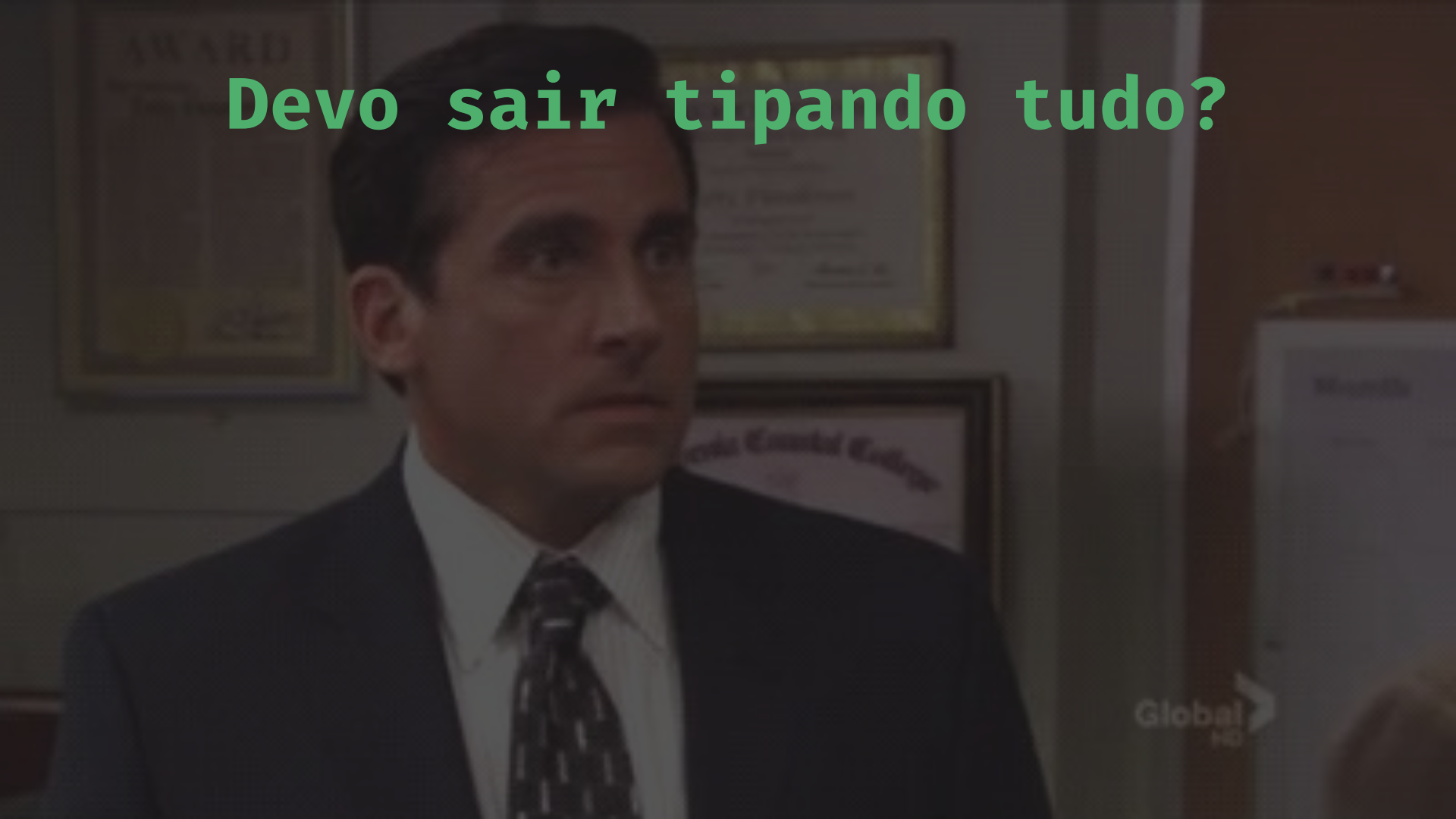
Quais problemas o TS resolve?

Podemos aplicar validação estática de nossa tipagem e garantir que parâmetros, definições e retornos seguem uma determinada regra.

```
~/Development/personal/gama-academy/xp-37/mini-projeto(master*) » npm run build

> gama-xp-37-ts@1.0.0 build
> tsc

src/1-tipos-basicos/index.ts:6:18 - error TS2345: Argument of type 'string' is not assignable to parameter of type 'number'.

6 console.log(soma('1', '1'))
                   ~~~
```

Que problemas o TS traz?

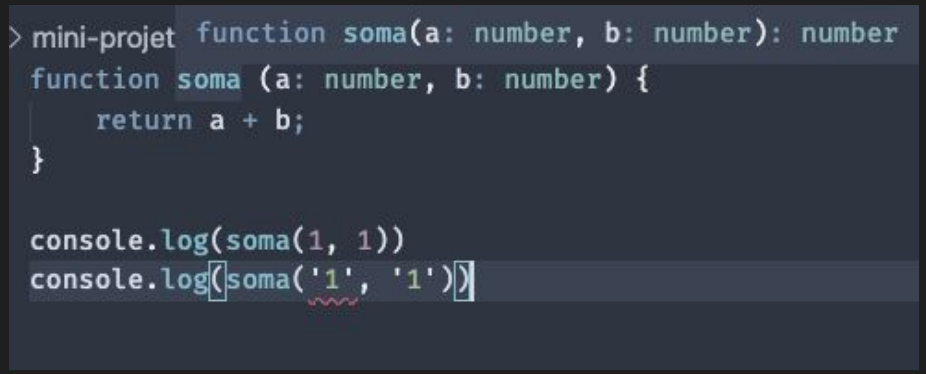
- Necessidade de um processo de build/compilação
- Uma certa burocratização do desenvolvimento
- Mensagens de erros nem sempre muito claras
- Falta de embasamento em JS dependendo de como é estudado


Devo sair tipando tudo?

O que é inferência?

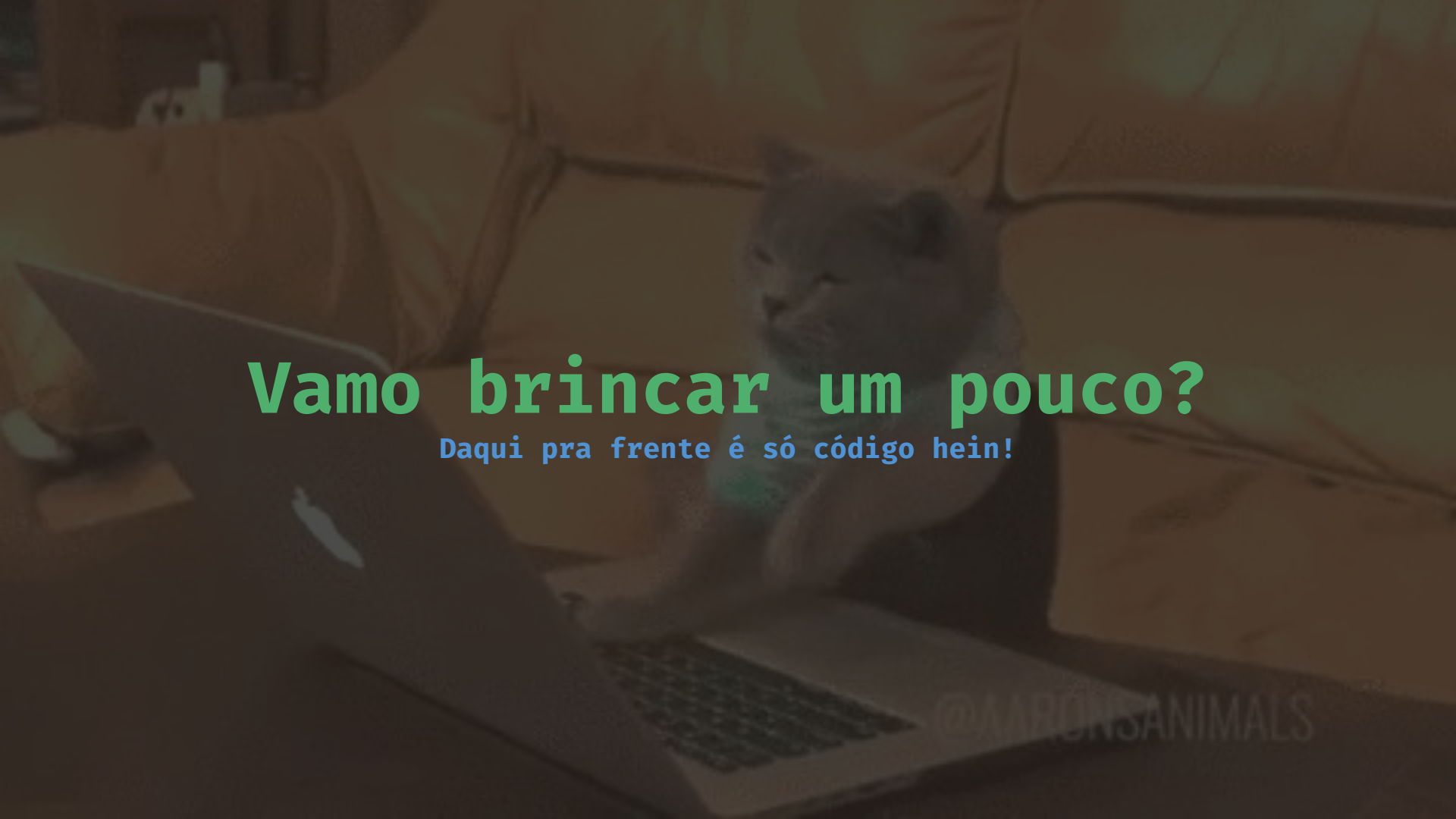
Inferir significa "deduzir" ou "concluir". E o próprio compilador do **TS** é capaz de deduzir muitos tipos, que permitem que o trabalho fique mais prático.

No exemplo abaixo, não deixamos explícito o tipo de retorno da função (apenas de seus parâmetros). Mesmo assim, o retorno é deduzido (ou inferido) pelo próprio compilador.



Como instalar?

Tão simples quanto:



Vamo brincar um pouco?

Daqui pra frente é só código hein!

Obrigado!

@gabrieluizramos

gabrieluizramos.com.br