

*Estruturas de Linguagem*

*Interpretação de Programas  
(com programação funcional)*

***Verificação***

**Francisco Sant'Anna**

**francisco@ime.uerj.br**

**<http://github.com/fsantanna-uerj/EDL>**

# Verificação Estática

- Nem todo programa com sintaxe ok é válido:
  - `"ola" + 10;` // tipos incompatíveis
  - `x = a + b;` // var não declarada
  - `strlen(s, 10);` // param inexistente

# Verificação Estática

- Linguagem com declaração de variáveis:
  - `verificaCmd :: Cmd -> Bool`
  - `verificaCmd :: [String] -> Cmd -> ([String], Bool)`

```
data Cmd = Atr String Exp  
         | Seq [Cmd]  
         | Dcl String
```

```
c1 = Atr "x" (Num 1)  
c2 = Seq (Dcl "x") c1
```

# O Tipo “Maybe”

- Usado quando houver possibilidade de falha:
  - `avalíaProg :: Cmd -> Maybe Int`
- Verificação do programa pode falhar
  - Em caso de falha, retornar `Nothing`
  - Em caso de sucess, retornar `Just Int`
- `Maybe a = Nothing | Just a`
- Uso do Maybe sempre exige considerar a falha:
  - ```
case avalíaProg <algum-cmd> of
  Nothing  -> trata o erro
  Just ret -> trata o sucesso
```