

Por que meus  
testes não são  
bons?

Eduardo Mendes



# Deep, deep Python

Eduardo Mendes

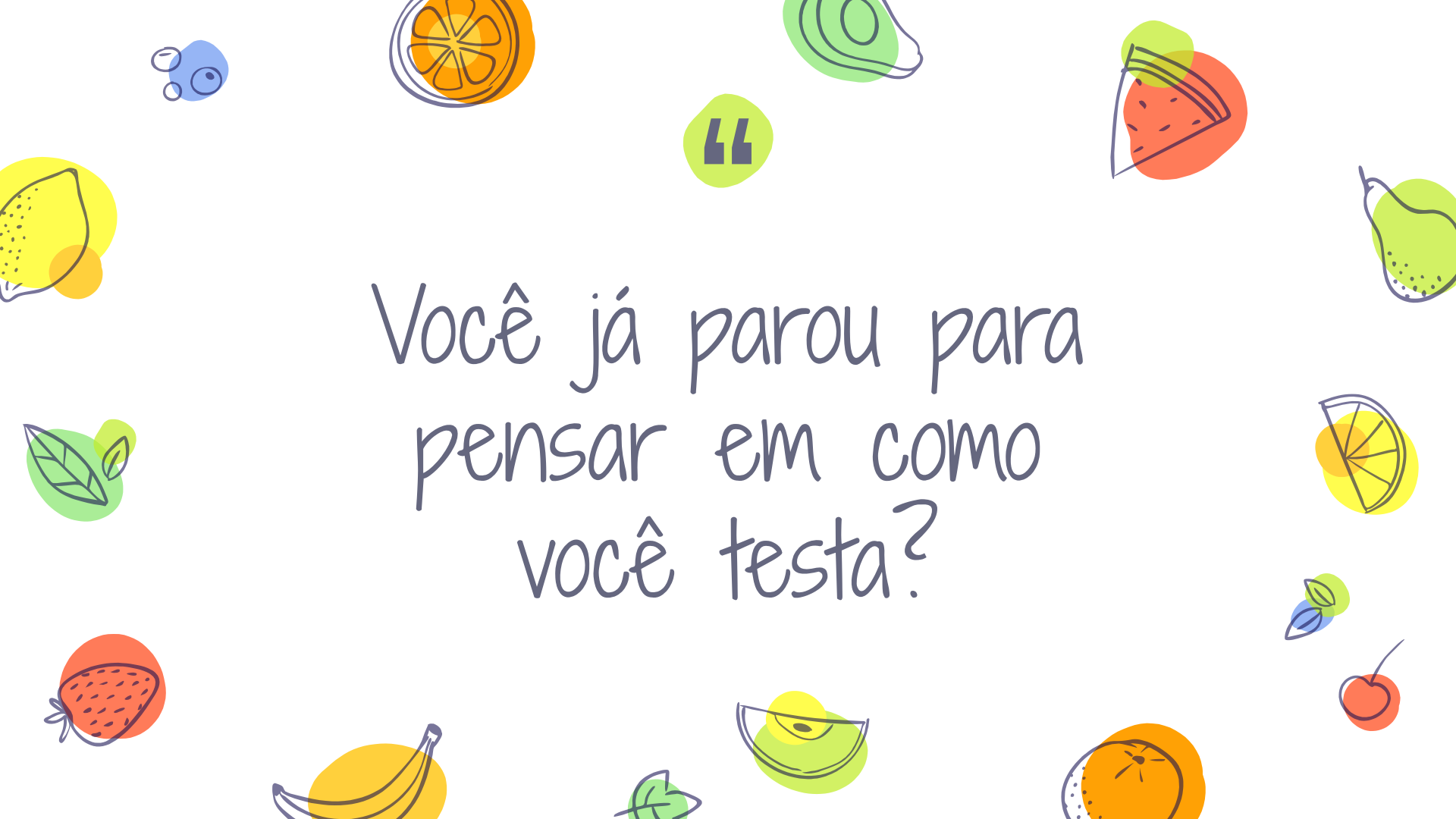




# Olar bebês!

Sou Eduardo Mendes (@dunossauro)

Apto fazedor de lambdas, pythonista, apaixonado por software **livre** e ciência.



Você já parou para  
pensar em como  
você testa?

```
def soma(x, y):  
    """Retorna uma soma de 2 números."""  
    return x + y  
  
def test_soma():  
    assert soma(0, 0) == 0
```

Um exemplo básico de uma função de soma e seu teste

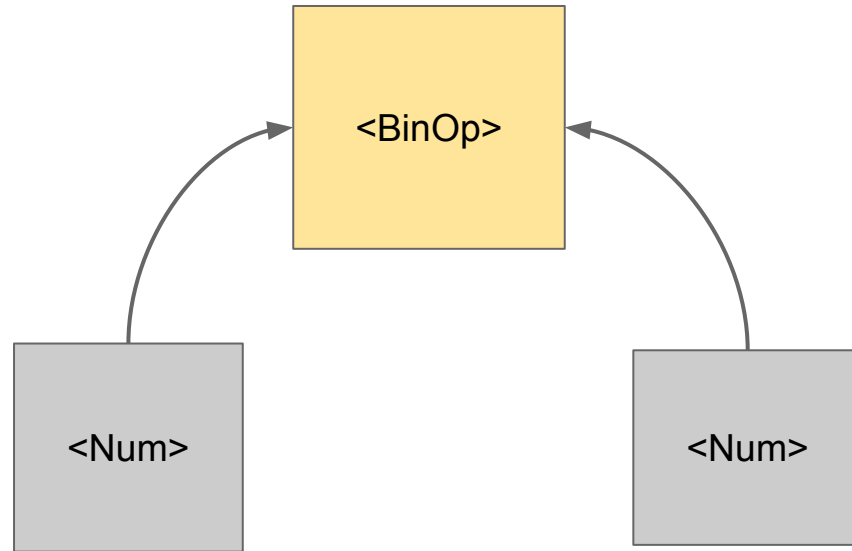
Mas esse teste é bom mesmo?



The background is a solid orange color. It is decorated with various white line-art illustrations of fruits and leaves. At the top, there are three small circles, a slice of citrus, a whole citrus, a slice of watermelon, and a pear. On the left side, there is a lemon and a leaf. On the right side, there is a slice of citrus and a pear. At the bottom, there is a strawberry, a banana, a leaf, a slice of citrus, a cherry, and an orange.

# Por que eu não acho bom?

Simples, meu teste não valida o meu **código**



Vamos refletir um pouco sobre aquela função

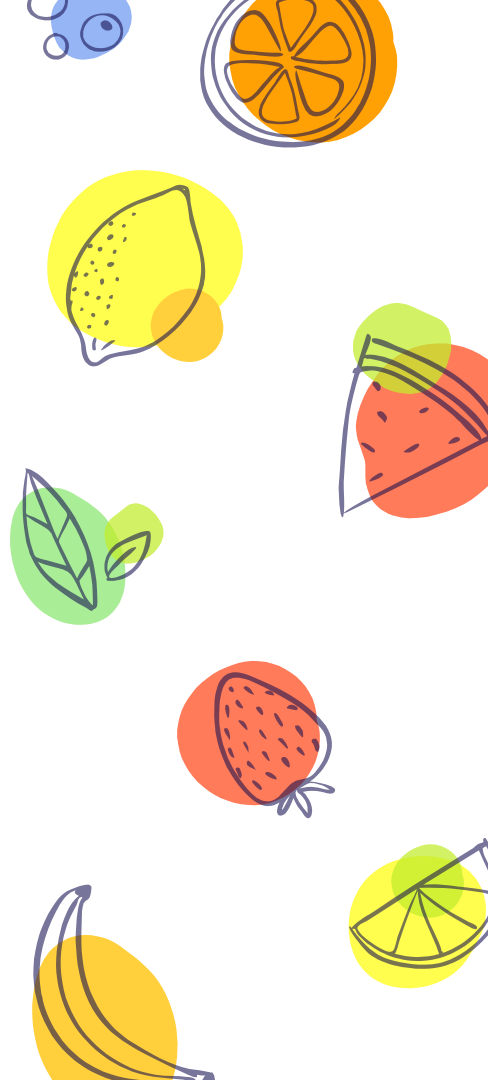


Tá, mas isso não diz como analisar o código!

# Testes de mutação

Existe uma categoria de testes que alterar o comportamento do nosso código no momento do teste.

Baseado em defeitos\*

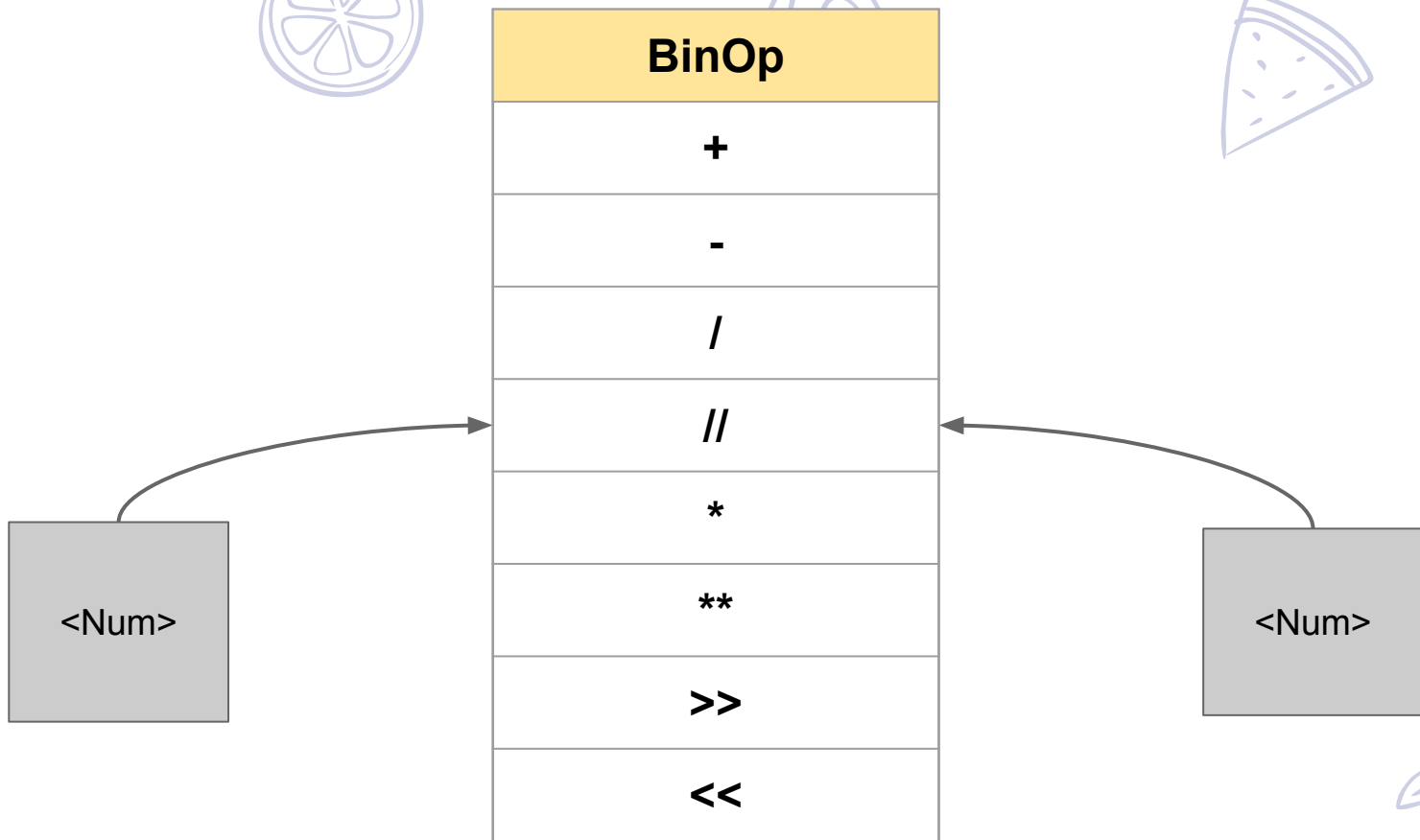


# A ideia principal da mutação

É trocar seus statements por outros

- ✗ AOR - Arithmetic Op Rotation
- ✗ AOI - Arithmetic Op Insertion
- ✗ ASR - Assignment op Replacement
- ✗ ...





**AOR**

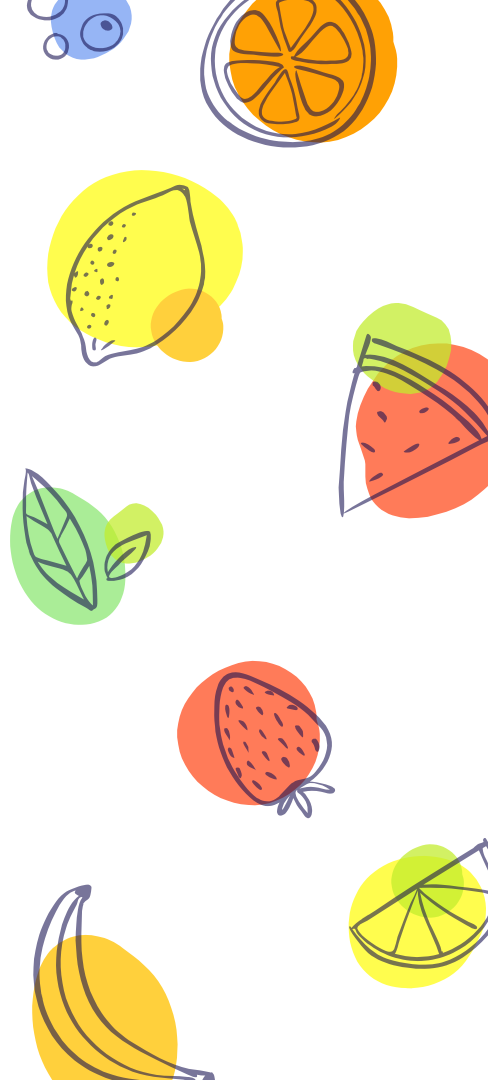
Para o tutorial de hoje você vai precisar de:

- ✗ AST
- ✗ inspec
- ✗ Muita atenção



# Talk baseada nas libs

- ✗ mutpy
- ✗ astmonkey
- ✗ cosmic-ray



Prepara que agora vêm código!