Sermão (2019)
Castello branco

Live de Python

#105 - Estruturas de decisão

Obrigado!

apoia.se/livedepython

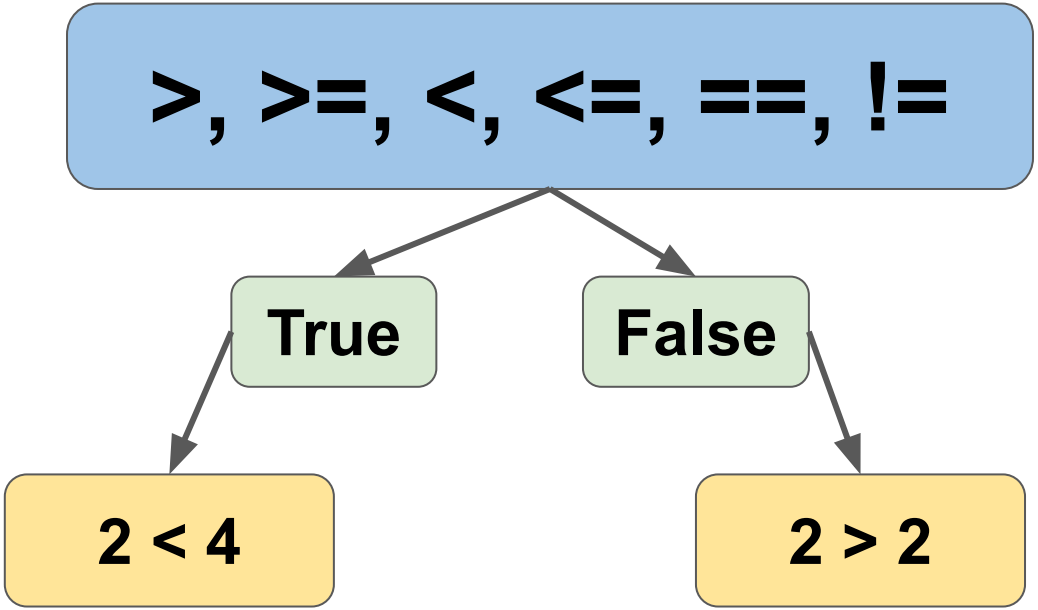
```
dunossauro at bouman in ~/git/apoiase on master*
$ python apoiadores.py
Alexandre Possebom   Alysson Oliveira     Amaziles Carvalho    Andre Machado
Andre Rodrigues      Bernardo Fontes      Bruno Guizi          Carlos Augusto
Cleber Santos        Cleiton Mittmann     David Reis           Dayham Soares
Diego Ubirajara      Edimar Fardim        Eliabe Silva         Eliakim Morais
Elias Soares         Emerson Lara         Eugenio Mazzini      Fabiano Silos
Fabiano Teichmann    Fabiano Gomes        Fernando Furtado     Franklin Silva
Fábio Serrão         Gleison Oliveira     Guilherme Ramos      Hemilio Lauro
Humberto Rocha       Hélio Neto           JOAO COELHO          JONATHAN DOMINGOS
Jean Vetorello       Johnny Tardin        Jonatas Oliveira     Jonatas Simões
João Lugão           Jucélio Silva        Júlia Kastrup        Leon Teixeira
Lucas Nascimento     Magno Malkut         Marcus Salgues       Maria Boladona
Matheus Francisco    Nilo Pereira         Pablo Henrique       Paulo Tadei
Pedro Alves          Rafael Galleani      Regis Santos         Renan Moura
Renato Santos        Rennan Almeida       Rodrigo Ferreira     Rodrigo Vaccari
Sérgio Passos        Thiago Araujo        Tiago Cordeiro       Tyrone Damasceno
Vergil Valverde      Vicente Marcal       Wander Silva         Welington Carlos
Wellington Camargo   Welton Souza         William Oliveira     Willian Gl
Yros Aguiar          Falta você           Falta você           Falta você
```

Roteiro

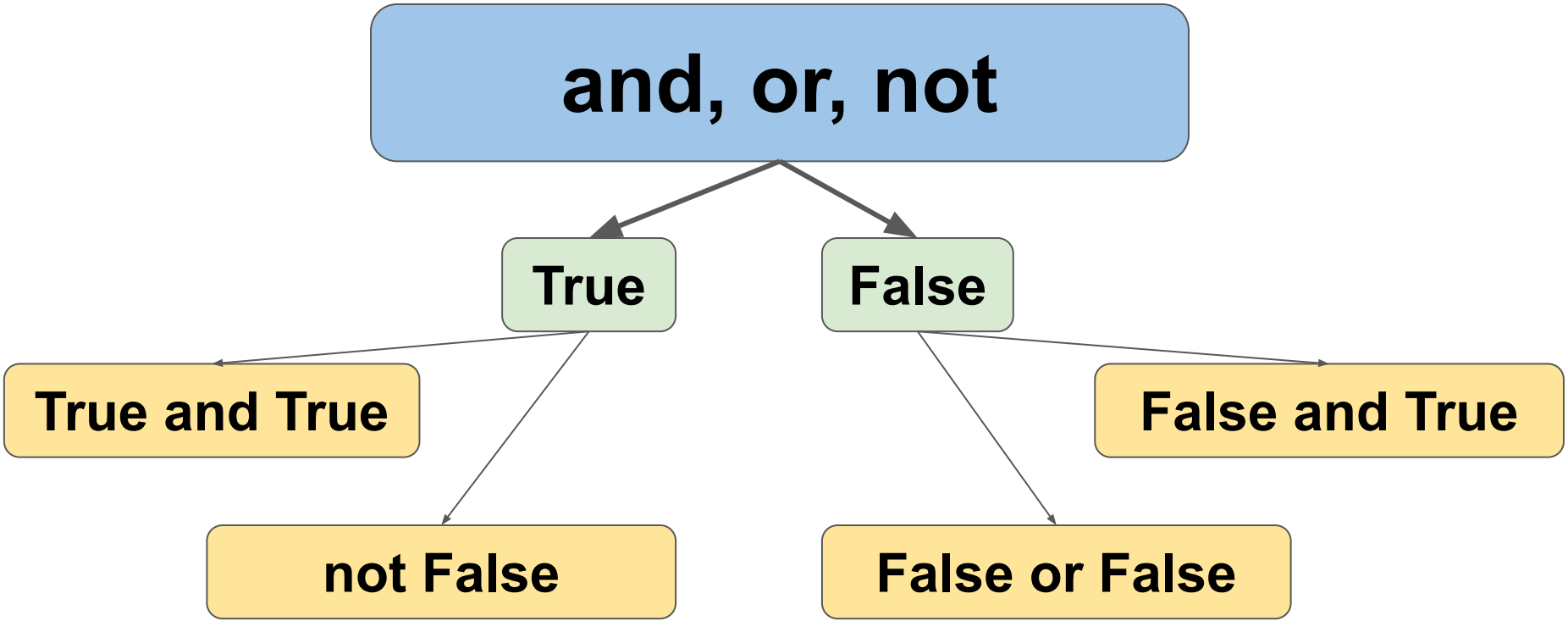
- Operadores de comparação
- Operadores booleanos
- Operadores de sequências
- A função bool()
- Deu curto aqui
- if / elif / else
- Como tudo isso funciona por trás?
- Dá pra interagir com tudo mesmo?

Referência: https://docs.python.org/3.8/library/stdtypes.html
Exercícios: https://wiki.python.org.br/EstruturaDeDecisao

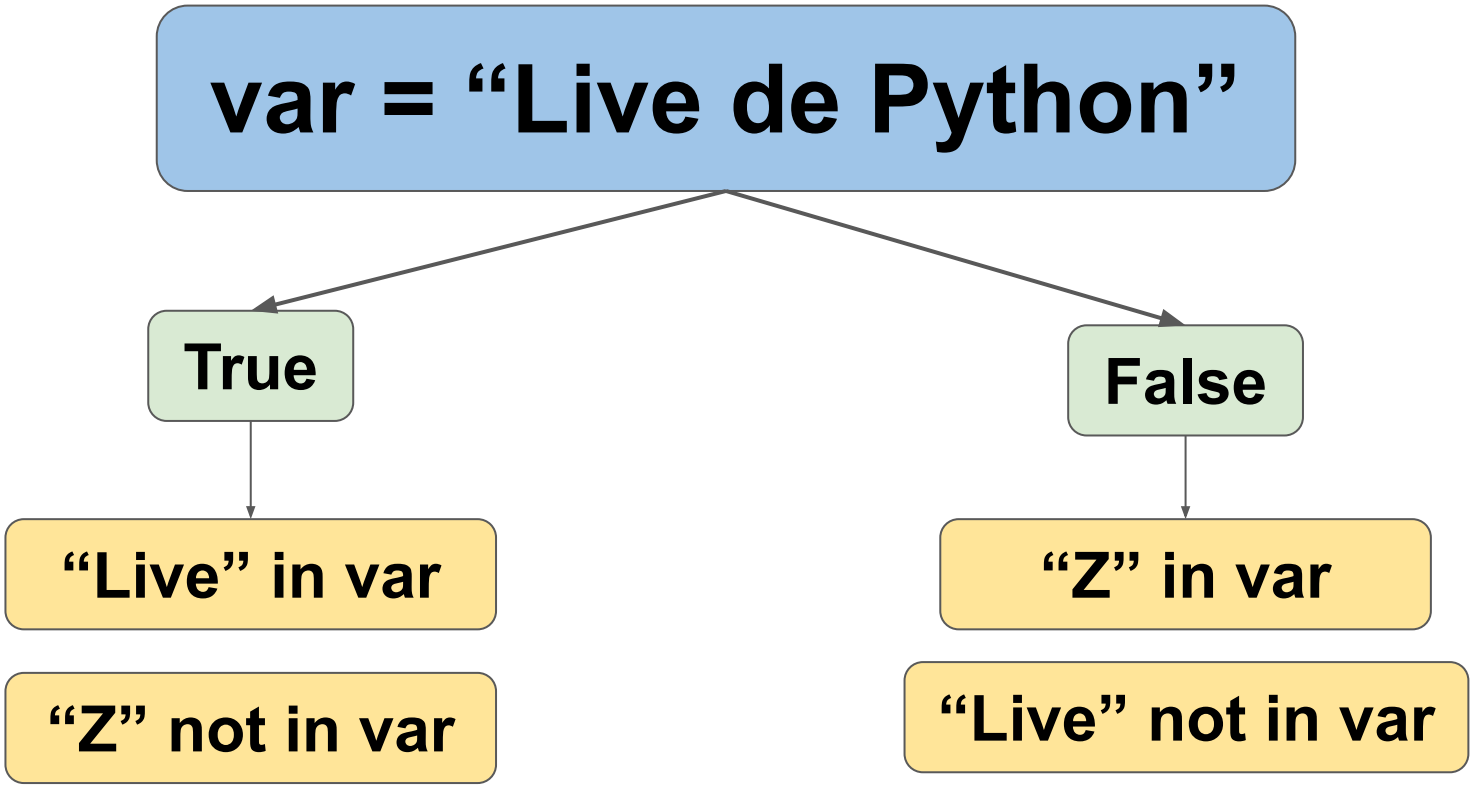
Operações de comparação - Bool



Operadores booleanos - Bool

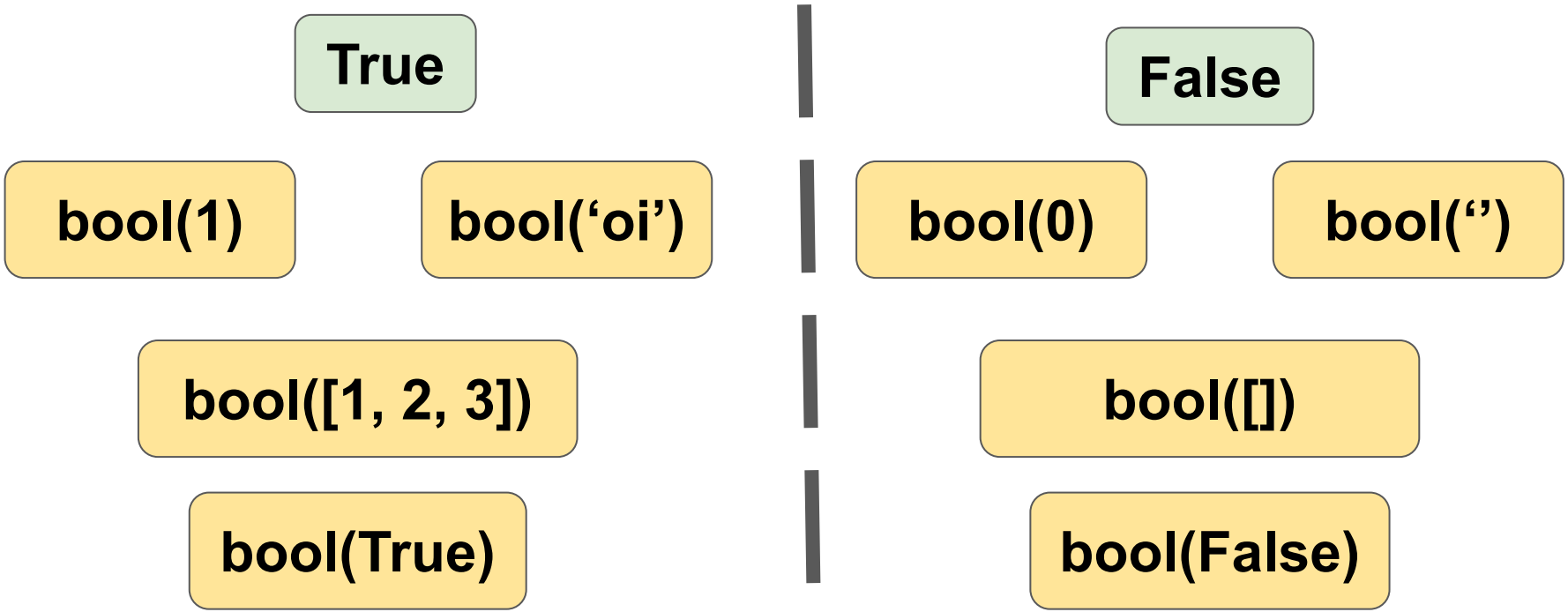


Operadores de sequência - Bool



A função bool()

A função **bool()** força o objeto a retornar um valor **True** ou **False**



Condições Falsas - Bool

- 0 (zero)
- False
- None
- Sequências vazias
  - ‘’ - string vazia
  - []
  - {}

Deu curto aqui

O python trabalha com avaliação de curto circuito em casos onde existem conectivos booleanos.

Vamos pensar no caso do operador **or**:

<Codar aqui, pfv>

Estruturas de decisão - if,elif,else

```
if <condição A>:
    print('Entramos na condição A')

elif <condição B>:
    print('Entramos na condição B')

else:
    print('Não acertamos a condição')
```

Estruturas

Parenteses não são necessários, nem mesmo com conectivos, ex: if x>y or x<y

```
>>> x = 7
>>> y = 6
>>> if x == y:
        print('Mesmo valor')
    elif x > y:
        print('x é maior que y')
    else:
        print(f'Não sei resolver {x} e {y}')
```

Fixação

Problema 1

Faça uma função que receba dois número e retorne o maior deles

Problema 2

Faça uma função que receba uma letra e retorne se é uma vogal ou uma consoante.

Problema 3

Faça uma função que recebe uma string e retorne se ela tem de:

- 0 a 5 caracteres

- 6 - 11 caracteres

- mais de 11

Problema 4

Tá, mas como o python faz isso?

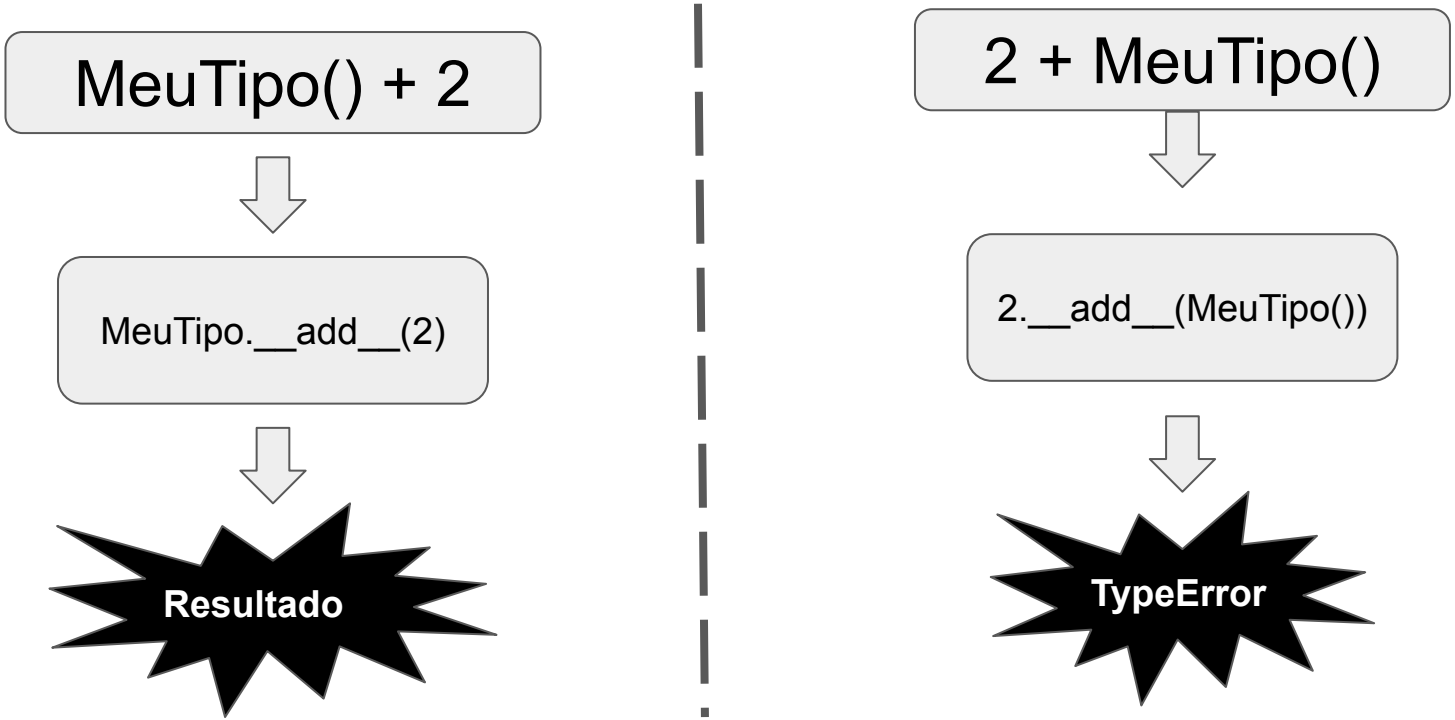
Operadores nos objetos

Operador	Método
==	__eq__
!=	__ne__
<=	__le__
>=	__ge__
<	__lt__
>	__gt__

Meus objetos conseguem interagir com isso?

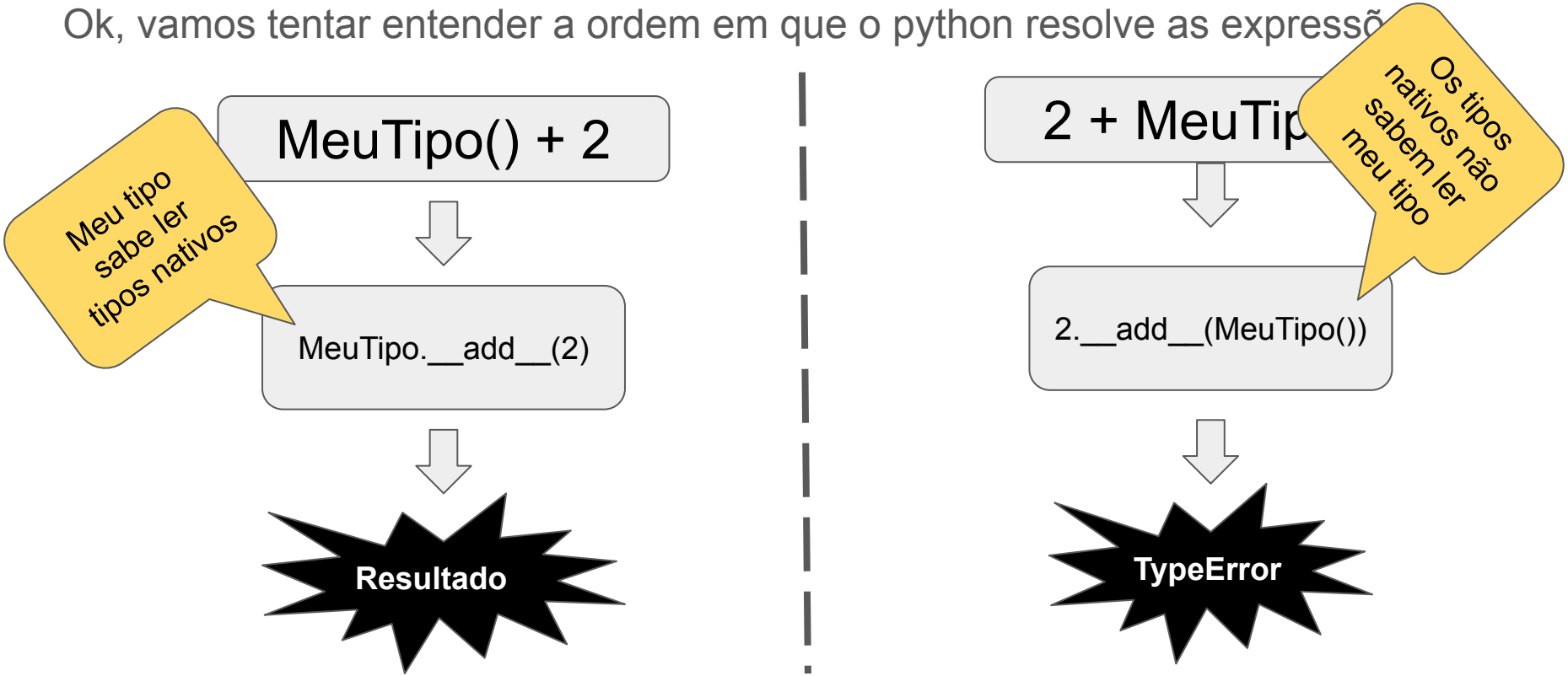
Operadores infixos

Ok, vamos tentar entender a ordem em que o python resolve as expressões

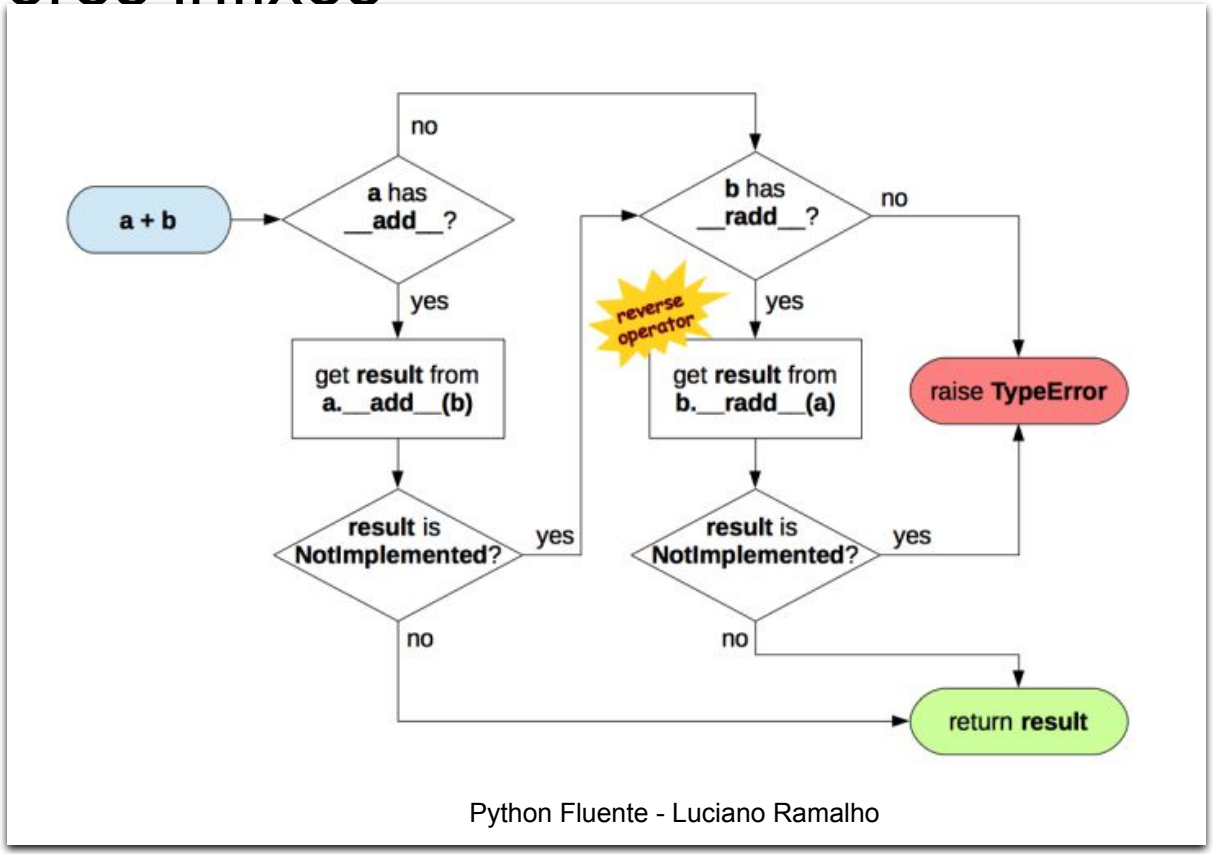


Operadores infixos

Ok, vamos tentar entender a ordem em que o python resolve as expressõ

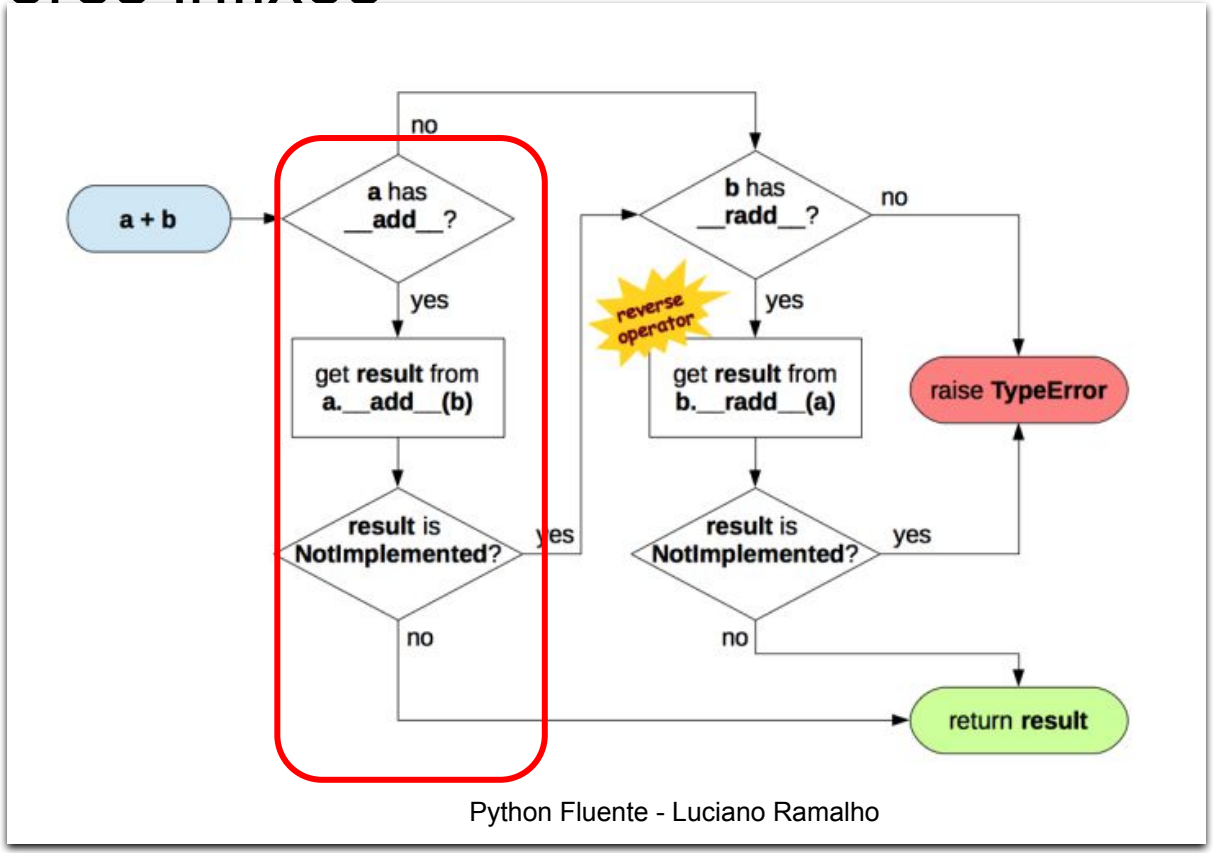


Operadores infixos



Python Fluente - Luciano Ramalho

Operadores infixos



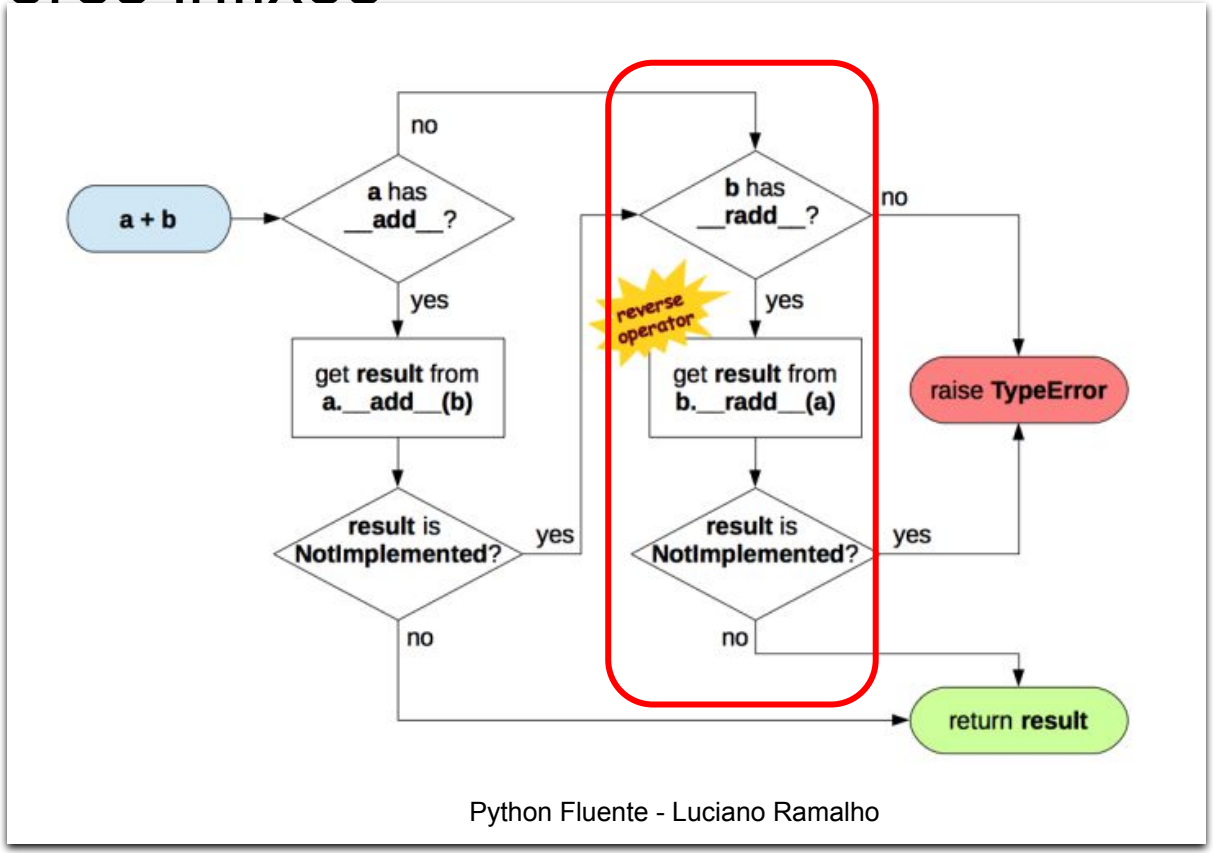
Python Fluente - Luciano Ramalho

Operadores infixos

E pra cada um desses operadores temos um **dunder** específico e um reverso

Operador	Método	Reverso
+	__add__	__radd__
-	__sub__	__rsub__
*	__mul__	__rmul__
/	__floordiv__	__rfloordiv__
//	__truediv__	__rtruediv__
<<	__lshift__	__rlshift__
>>	__rshift__	__rrshift__

Operadores infixos



Python Fluente - Luciano Ramalho

Operador infixo com operação reversa

```
In [15]: class Somável:
    ...:     def __add__(self, OUTRO_VALOR):
    ...:         print('Olha só, eu sei somar')
    ...:         return Somável()
    ...:     def __radd__(self, OUTRO_VALOR):
    ...:         print('Olhá, já sei somar reverso')
    ...:         return Somável()
    ...:

In [16]: Somável() + 2
Olha só, eu sei somar
Out[16]: <__main__.Somável at 0x7f6e1e53e3c8>

In [17]: 2 + Somável()
Olhá, já sei somar reverso
Out[17]: <__main__.Somável at 0x7f6e1e567518>
```