

Tratamento de Exceções


Prof. Alberto Costa Neto
Programação em Python

A estrutura `try` / `except`

- Você pode colocar um bloco de comandos dentro um `try/except`
- Se o código dentro do `try` funcionar – o que está em `except` é desconsiderado
- Se o código dentro do `try` gerar um `erro` – pula-se para os comandos dentro da seção `except`

```
$ cat notry.py
str = 'Ola'
inteiro = int(str)
print('Primeiro', inteiro)
str = '123'
inteiro = int(str)
print('Segundo', inteiro)
```

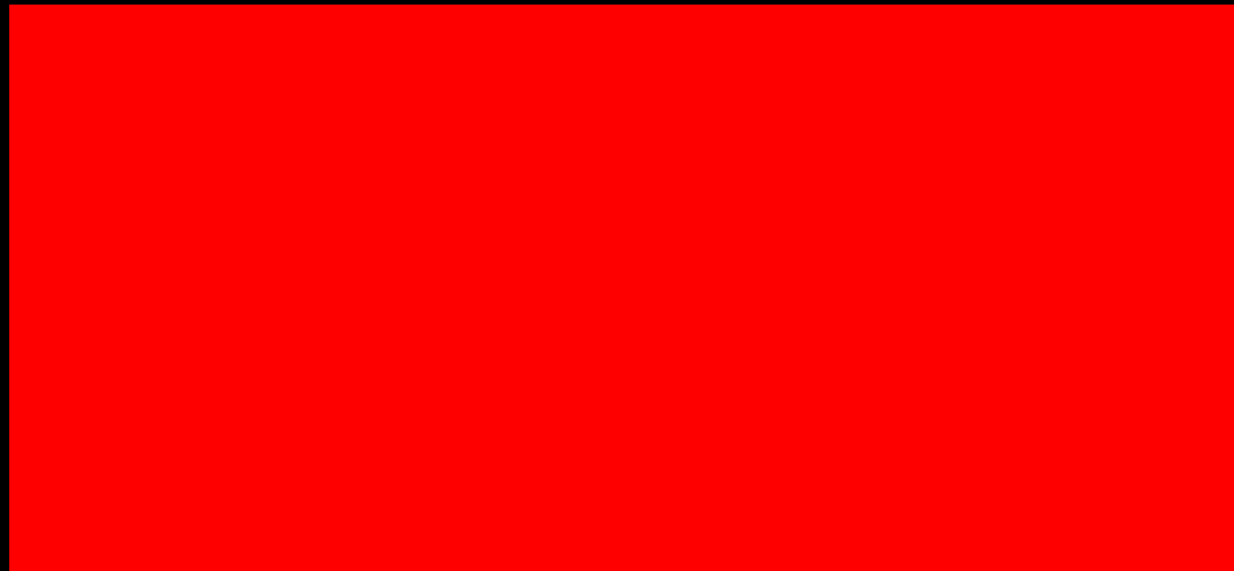
```
$ python3 notry.py
Traceback (most recent call
last):
File "notry.py", line 2, in
<module>    inteiro = int(str)
ValueError: invalid literal for
int() with base 10: 'Olá'
```

 Fim

O programa
para aqui



```
$ cat notry.py  
str = 'Ola'  
inteiro = int(str)
```



```
$ python3 notry.py  
Traceback (most recent call  
last):  
File "notry.py", line 2, in  
<module>    inteiro = int(str)  
ValueError: invalid literal for  
int() with base 10: 'Olá'
```



Fim

```
$ cat tryexcept.py
```

```
str = 'Ola Bob'
```

```
try:
```

```
→ inteiro = int(str)
```

```
except:
```

```
    inteiro = -1 ←
```

```
print('Primeiro', inteiro)
```

```
str = '123'
```

```
try:
```

```
→ inteiro = int(str)
```

```
except:
```

```
    inteiro = -1
```

```
print('Segundo', inteiro)
```

Quando a primeira conversão falha:
- pula-se para dentro da cláusula
except: e o programa continua

```
$ python3 tryexcept.py
```

```
Primeiro -1
```

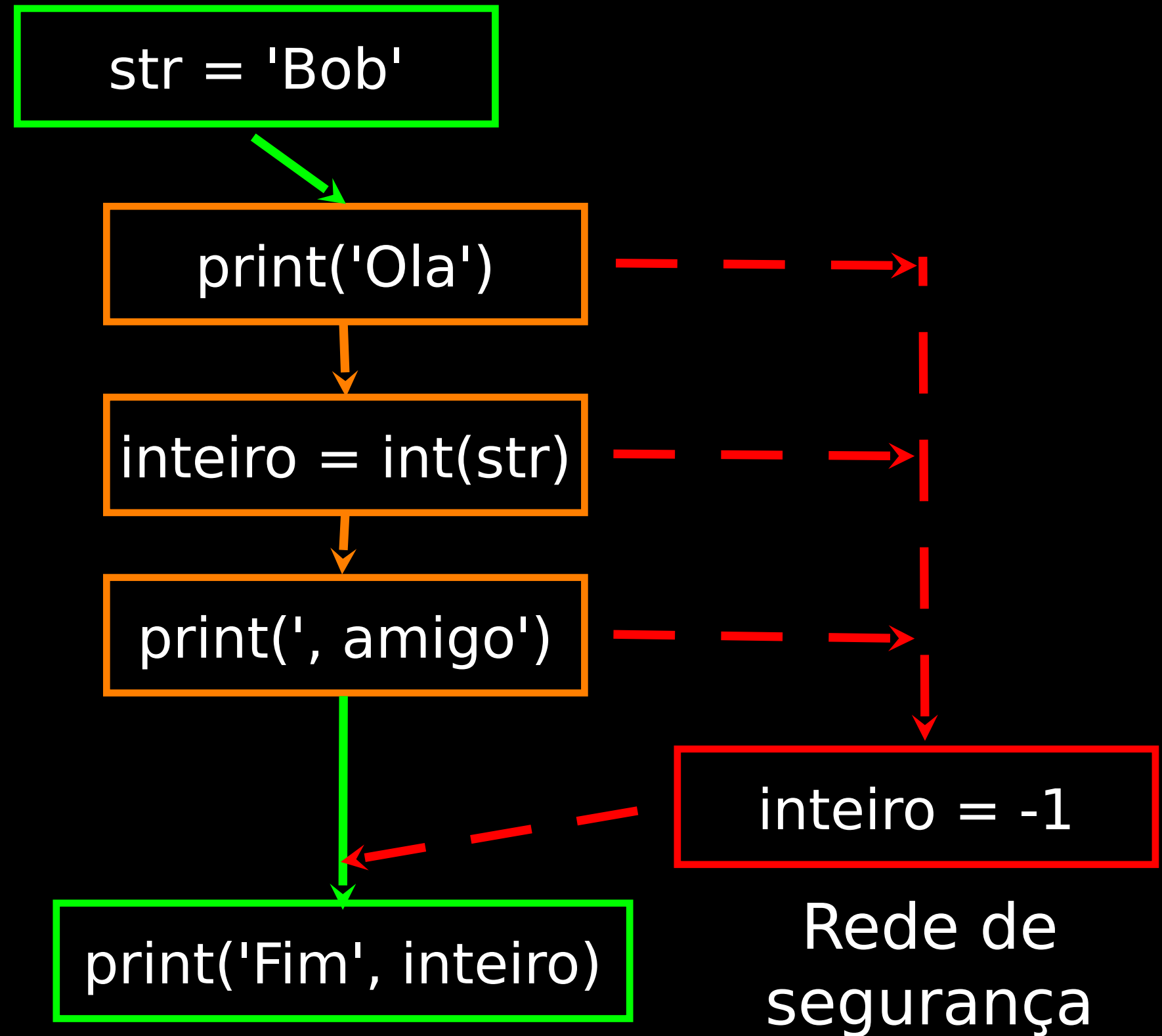
```
Segundo 123
```

Quando a segunda conversão
tem sucesso:

- pula-se a cláusula except: e
o programa continua

try / except

```
str = 'Bob'
try:
    print('Ola')
    inteiro = int(str)
    print(', amigo')
except:
    inteiro = -1
print('Fim', inteiro)
```



Exemplo de try / except

```
valstr = input('Numero:')
try:
    ival = int(valstr)
except:
    ival = -1

if ival > 0 :
    print('Bom trabalho')
else:
    print('Nao e numero')
```

```
$ python3 trynum.py
Numero:42
Bom trabalho
```

```
$ python3 trynum.py
Numero: trinta
Nao e numero
```

Exemplo de try / except

```
valstr = input('Numero:')  
num_str = input('Numerador: ')  
den_str = input('Denominador: ')  
res = None
```

```
try:  
    num, den = int(num_str), int(den_str)  
    res = num / den  
except ZeroDivisionError:  
    print('Erro de divisão por zero')  
except ValueError as e:  
    print('Numerador e/ou denominador não inteiros')  
    print(e)
```

```
if res is None:  
    print('Impossível calcular')  
else:  
    print('Resultado é:', res)
```

```
$ python3 trynum.py  
Numerador: 30  
Denominador: 5  
Resultado é 6.0
```

```
$ python3 trynum.py  
Numerador: 42  
Denominador: 0  
Erro de divisão por zero  
Impossível calcular
```

```
$ python3 trynum.py  
Numerador: trinta  
Denominador: 5  
Invalid literal for int()  
with base 10: 'trinta'  
Impossível calcular
```