

Introdução a Python

IPL 2021



O que é um programa?

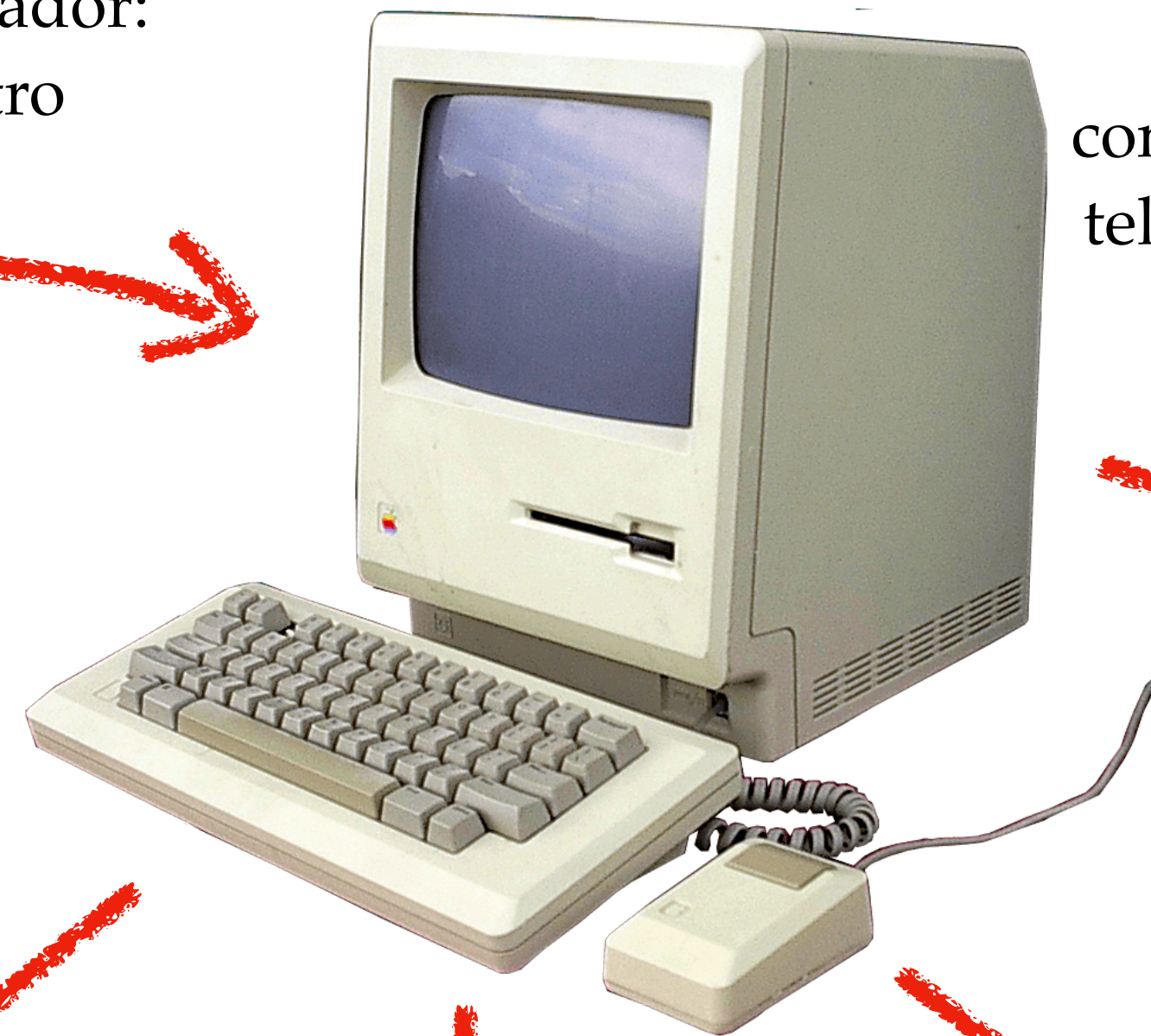
Conjunto de instruções de como realizar uma computação (geralmente para resolver um problema)

Entrada (input)

Dados fornecidos ao computador:
teclado, arquivo, rede, outro
dispositivo

Saída (output)

Resultado da
computação: mostrar na
tela, salvar em arquivo,
enviar por rede



Matemática

Operações em objetos
representados como bits

Repetição

Possivelmente realizar
operações semelhantes
múltiplas vezes

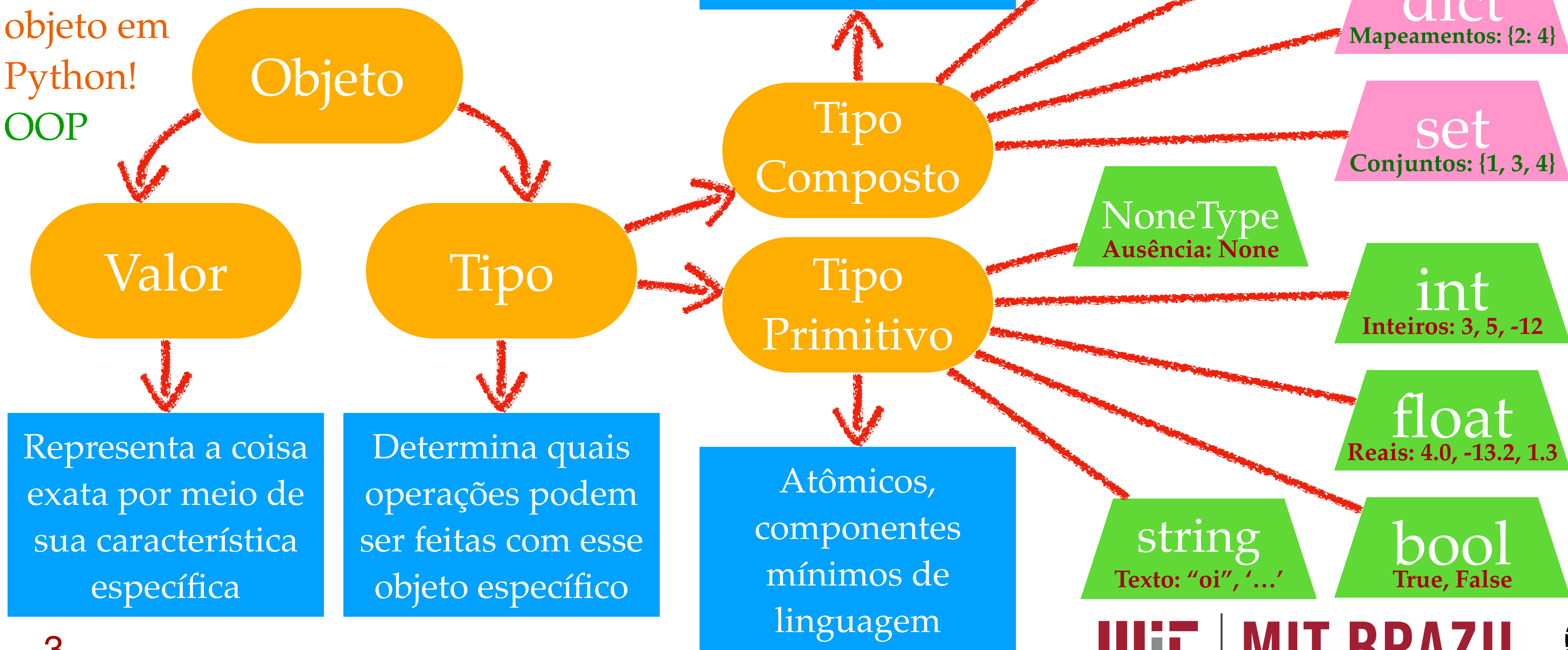
Execução Condicional

Decisões sobre quais
computações realizar

Objetos, valores e tipos primitivos / compostos

PYTHON VERSION

Tudo é um
objeto em
Python!
OOP



Operadores

Algébricos

+
Adição

-
Subtração

Multiplicação

/
Divisão

Exponenciação

→ Não é ^!!!

//
Divisão inteira

%
Módulo

Booleanos

==
Igual a

True se iguais

>
Maior que

True se esq.
estritamente
maior que dir.

>=
Maior ou igual

True se esq.
maior ou igual
a dir.

!=
Não igual a

False se iguais

<
Menor que

True se esq.
estritamente
menor que dir.

<=
Menor ou igual

True se esq.
menor ou igual
a dir.

and

True se ambos
são True

AND	True	False
True	True	False
False	False	False

or

True se pelo
menos um é True

OR	True	False
True	True	True
False	True	False

not

Oposto do valor
booleano

NOT	
True	False
False	True

Variáveis

Atribuição

```
1 dentes = 28
2 cabeca = 1
3 dedos = 20
4 joelhos = 2
5 |
```



Comentários começam com #, vão até o final da linha

```
1 # ruim: o que é r? Não significa nada fora de contexto
2 r = 10
3
4 # ruim: muito longo! Magina escrever isso 10 vezes
5 raio_do_circulo_que_eu_desenhei = 10
6
7 # 👍
8 raio = 10
9
```

Nomes reservados (keywords)

Não podem ser usados para nomear variáveis

and	as	assert	async	await	break	class	continue
def	del	elif	else	except	finally	for	from
global	if	import	in	is	lambda	nonlocal	not
or	pass	raise	return	try	while	with	yield

Escolhendo nomes:

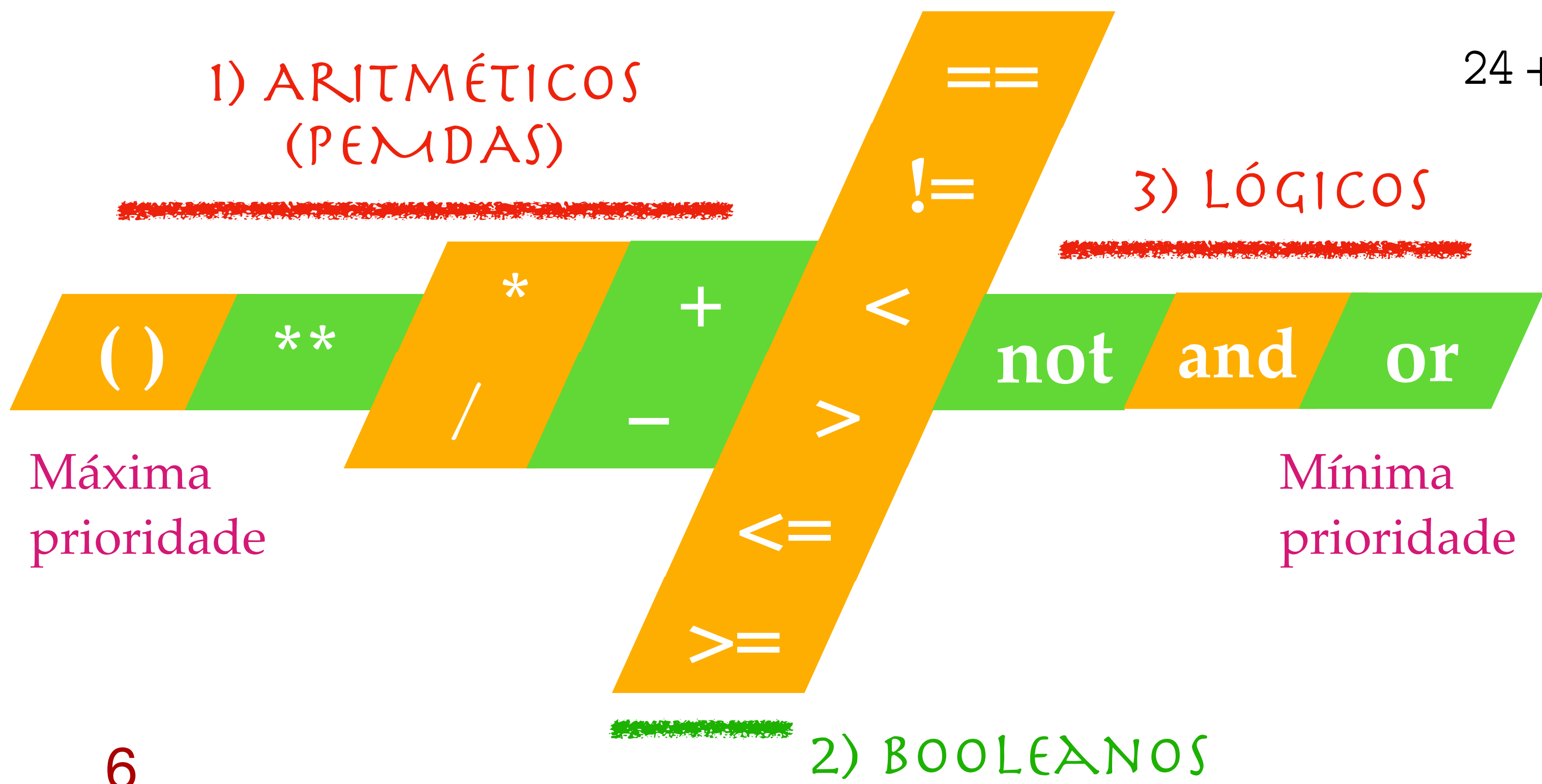
equilíbrio entre legibilidade e tamanho

True, False, None também são keywords

Modelo de Substituição

- Python trabalha na ordem das operações
- Avalia cada subexpressão
- Substitui cada uma por seu resultado

Ordem de Operações



Exemplo

`24 + 1 * 3 ** 2 > 30 and (4 * 2) ** 2 - 20 > 80 or 10 != 10.0`

`24 + 1 * 3 ** 2 > 30 and 8 ** 2 - 20 > 80 or 10 != 10.0`

`24 + 1 * 9 > 30 and 64 - 20 > 80 or 10 != 10.0`

`24 + 9 > 30 and 64 - 20 > 80 or 10 != 10.0`

`33 > 30 and 44 > 80 or 10 != 10.0`

`True and False or False`

`False or False`

`False`

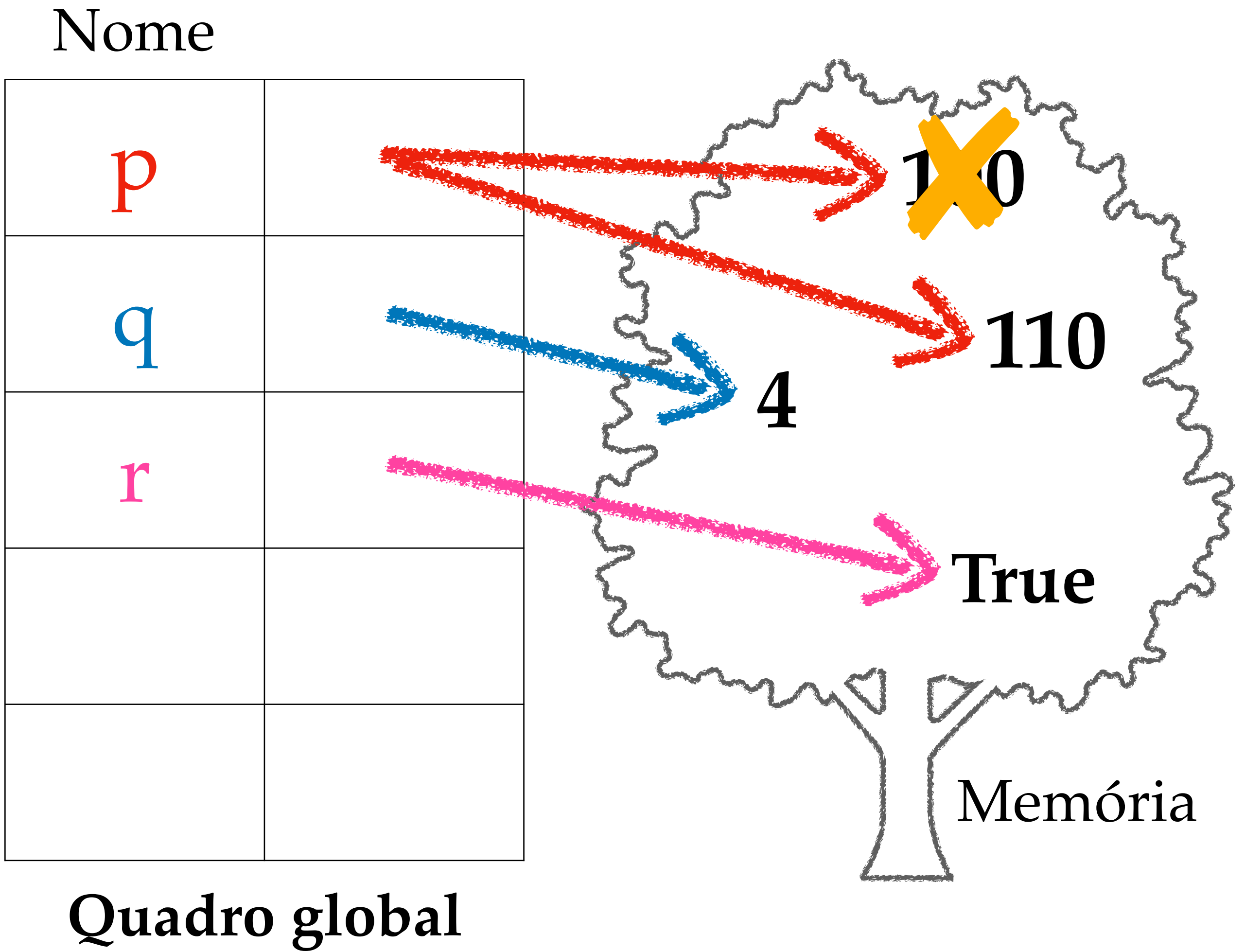
Diagramas de Ambiente

Exemplo

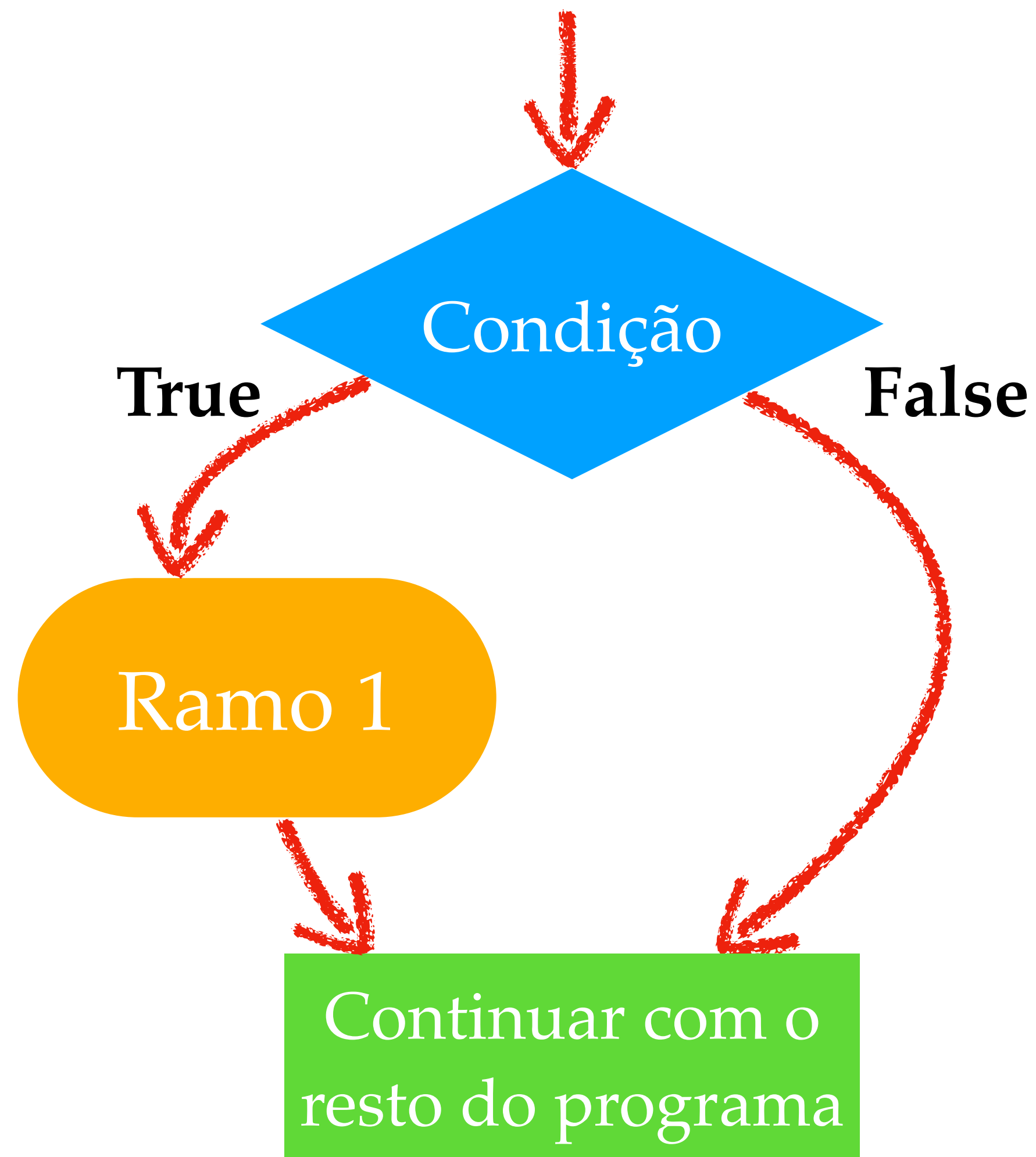
```
1 p = 100
2 print(p ** 2)
3 q = p // 23
4 r = 10 > 20 or 1 ** 4 > 1 or 100
5 p += 10
6 print(r)
7
```

0 tem valor **False** se usado como bool, todos os outros números são **True**

$x \blacksquare = y$ é idêntico a $x = x \blacksquare y$ para qualquer operador em Python

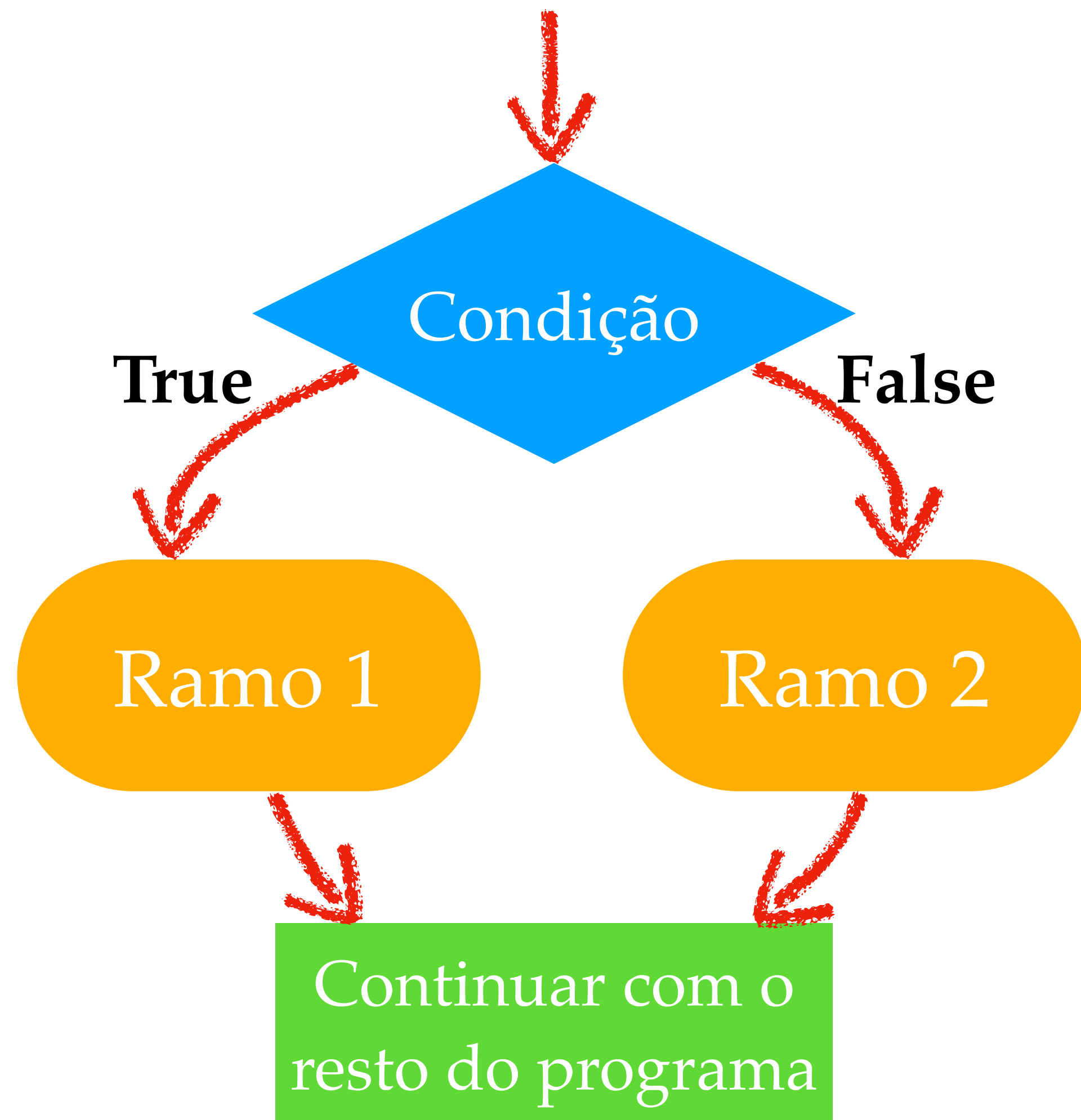


Execução Condicional: if



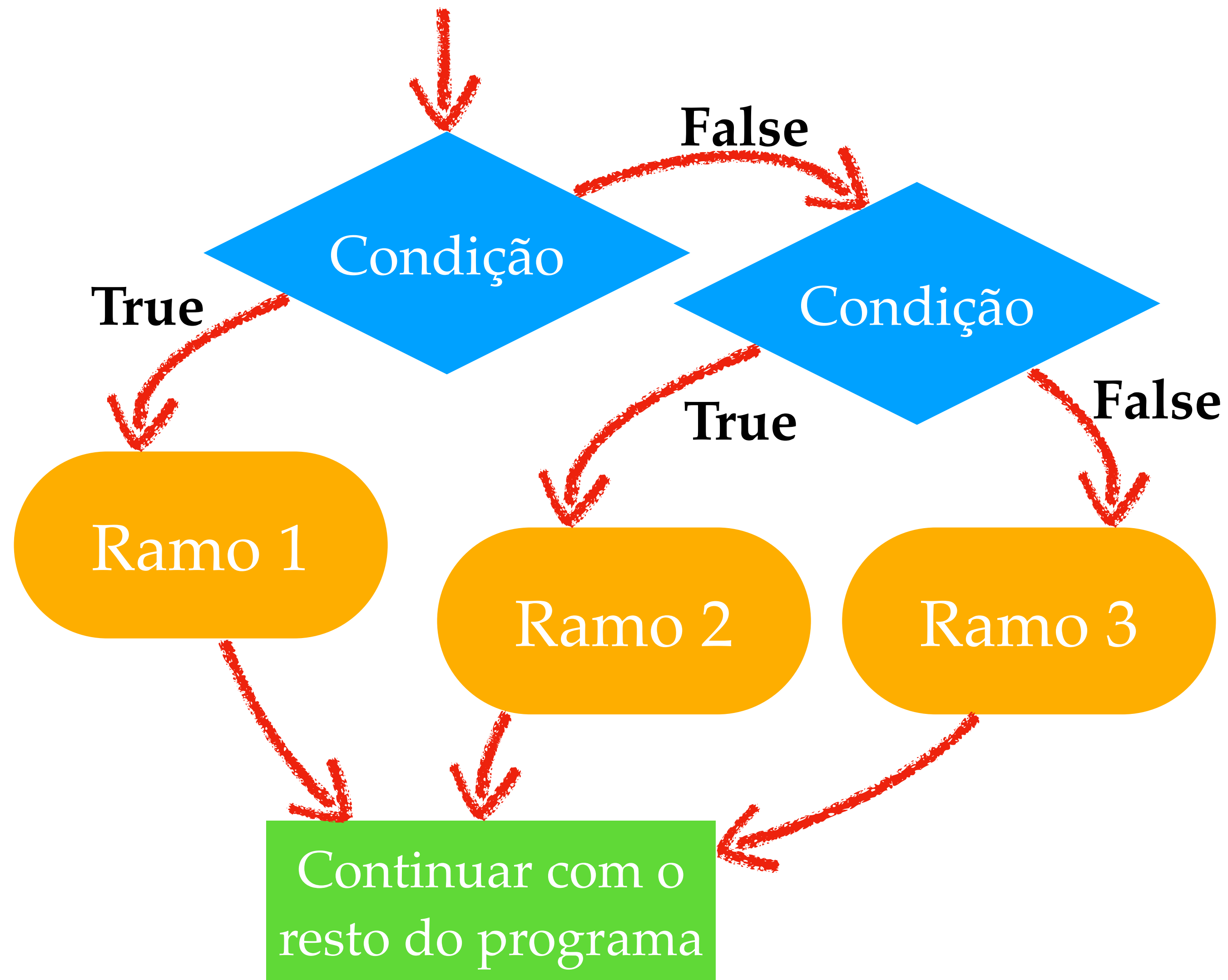
```
1  x = 12
2
3  if x >= 12:
4      print("Hooray!")  # ramo 1
5
6  print(x)
7  |
```


Execução Condicional: if/else



```
1  x = 12
2
3  if x >= 12:
4      print("Hooray!") # ramo 1
5  else:
6      print("Oof") # ramo 2
7
8  print(x)
9  |
```

Execução Condicional: if / elif / else



```
1  x = 12
2
3  if x > 12:
4      print("Hooray!") # ramo 1
5  elif x == 12:
6      print("Bingo!") # ramo 2
7  else:
8      print("Oof") # ramo 3
9
10 print(x)
11
```