



PRO_GRAMMING

< OFICINA DE PYTHON >



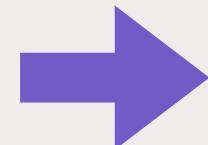
< EXEMPLOS DE LÓGICA >

Betim tinha X reais. Apostou em uma bet e perdeu R\$300,00. Seu amigo Vitão o convidou para assistir a um jogo de futebol. Porém, Beto só poderá ir se tiver mais de R\$50,00 na conta. Descreva a lógica apresentada



< EXEMPLOS DE LÓGICA >

Quanto tem na conta? -> valor
saldo = valor - 300
se saldo é maior que 50:
 mostrar "Beto vai ao jogo"
senão:
 mostrar "Beto não vai ao jogo"



```
Saldo = int(input("Quanto tem na conta:"))
Saldo = Saldo - 300
if Saldo > 50:
    print("Beto vai ao jogo")
else:
    print("Beto não vai ao jogo")
```



< PARADIGMA IMPERATIVO >

OPERAÇÕES MAIS COMUNS

- Ler, transformar, salvar
- Entrada e saída
- Condicionais
- Repetições (Loops)
- Funções



```
Programa.py - □ x
Instrução 1
Instrução 2
Instrução 3
Instrução 4
Instrução 5
:
```



< EXEMPLO >

Nome: _



< EXEMPLO >

Nome: Alfredo
Ano de Nascimento: _



< EXEMPLO >

Nome: Alfredo
Ano de Nascimento: 1947
Alfredo tem 76 anos



< EXEMPLO >

Nome: Alfredo
Ano de Nascimento: 1947
Alfredo tem 77 anos

Nome = Alfredo
Nascimento = 1947
Idade = 2024 - Nascimento
print(f"Alfredo tem {idade} anos")



< EXEMPLO >

```
Nome = Alfredo  
Nascimento = 1947  
Idade = 2024 - Nascimento  
print(f"Alfredo tem {idade} anos")
```

LITERAIS são valores
usados no código



< EXEMPLO >

```
Nome = Alfredo
Nascimento = 1947
Idade = 2024 - Nascimento
print(f"Alfredo tem {idade} anos")
```

A função PRINT é usada
para mostrar informações
na tela do terminal



```
Prompt de Comando
Microsoft Windows [versão 10.0.22621.2134]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\PET Eletrica>77
```



< EXEMPLO >

```
Nome = Alfredo  
Nascimento = 1947  
Idade = 2024 - Nascimento  
print(f"Alfredo tem {idade} anos")
```

NOME, NASCIMENTO
e **IDADE** são
variáveis



< COMENTÁRIOS >

Raio = 10

pi = 3.1415

Circ = 2 * pi * Raio

#raio do circulo

#valor de pi

#circunferência

"""" Este é um comentário de várias
linhas. Nada que estiver aqui será
executado """"

A = A + 10

#Aqui vai!



< TIPOS DE VARIÁVEIS >

**Informações em linguagens de
programação possuem
diferentes tipos. As operações
disponíveis e os
comportamentos destas
operações dependem dos tipos**



< TIPOS DE VARIÁVEIS >

Informações em linguagens de programação possuem diferentes tipos. As operações disponíveis e os comportamentos destas operações dependem dos tipos

int: 1, 10, -15, 0

float: 1.2, -6.25

bool: True, False

string: "palavra", "Flu"



< OPERADORES MATEMÁTICOS >

OPERAÇÕES MAIS COMUNS

- + : soma**
- : subtração**
- * : multiplicação**
- / : divisão real**
- // : divisão inteira**
- % : resto da divisão inteira**
- ** : exponenciação**



< OPERADORES DE COMPARAÇÃO >

OPERAÇÕES MAIS COMUNS

- $==$: igual à**
- $!=$: diferente de**
- $>$: maior que**
- $<$: menor que**
- \geq : maior ou igual à**
- \leq : menor ou igual à**



< OPERADORES DE COMPARAÇÃO >

**EXEMPLOS PARA
PRATICAR**

$5 \geq 5$	→
$3 == 2$	→
$3 != 2$	→
$3 == "3"$	→
$5 < 5$	→
$3.0 == 3$	→



< OPERADORES DE COMPARAÇÃO >

**EXEMPLOS PARA
PRATICAR**

$5 \geq 5$	→	True ✓
$3 == 2$	→	
$3 != 2$	→	
$3 == "3"$	→	
$5 < 5$	→	
$3.0 == 3$	→	



< OPERADORES DE COMPARAÇÃO >

EXEMPLOS PARA PRATICAR

$5 \geq 5$	→	True ✓
$3 == 2$	→	False ✗
$3 != 2$	→	
$3 == "3"$	→	
$5 < 5$	→	
$3.0 == 3$	→	



< OPERADORES DE COMPARAÇÃO >

EXEMPLOS PARA PRATICAR

$5 \geq 5$	→	True ✓
$3 == 2$	→	False ✗
$3 != 2$	→	True ✓
$3 == "3"$	→	
$5 < 5$	→	
$3.0 == 3$	→	



< OPERADORES DE COMPARAÇÃO >

EXEMPLOS PARA PRATICAR

$5 \geq 5$	→	True ✓
$3 == 2$	→	False ✗
$3 != 2$	→	True ✓
$3 == "3"$	→	False ✗
$5 < 5$	→	
$3.0 == 3$	→	



< OPERADORES DE COMPARAÇÃO >

EXEMPLOS PARA PRATICAR

$5 \geq 5$	→	True ✓
$3 == 2$	→	False ✗
$3 != 2$	→	True ✓
$3 == "3"$	→	False ✗
$5 < 5$	→	False ✗
$3.0 == 3$	→	



< OPERADORES DE COMPARAÇÃO >

EXEMPLOS PARA PRATICAR

$5 \geq 5$	→	True ✓
$3 == 2$	→	False ✗
$3 != 2$	→	True ✓
$3 == "3"$	→	False ✗
$5 < 5$	→	False ✗
$3.0 == 3$	→	True ✓



< TIPOS DE VARIÁVEIS >

str: Converte em string
int: Converte em inteiro
float: Converte em float

"3" + str(5)	→	"35"
int("3") + 5	→	8
float("3") + 5	→	8.0
"3" + 5	→	ERRO

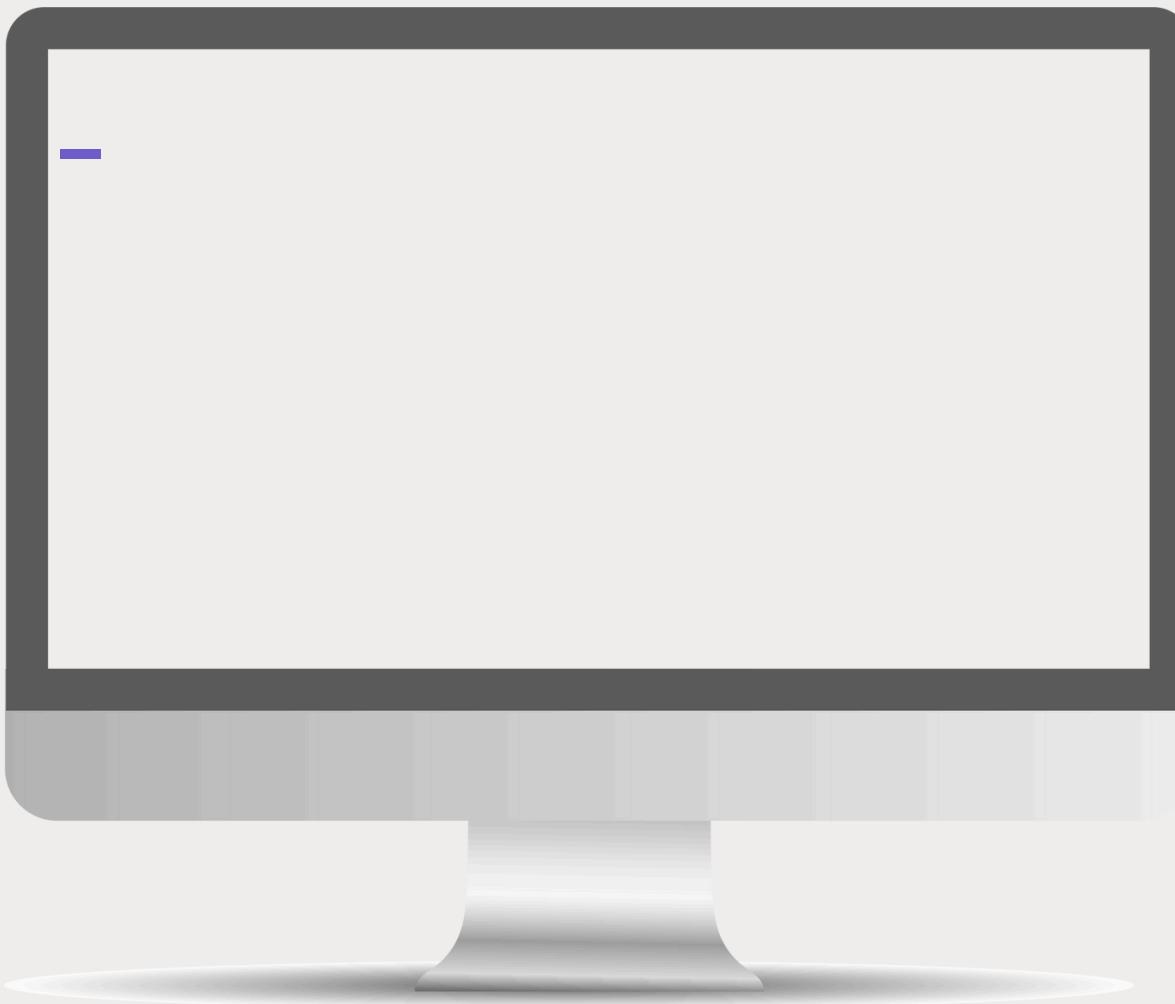


< EXEMPLO >

```
nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - int(nascimento)
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```



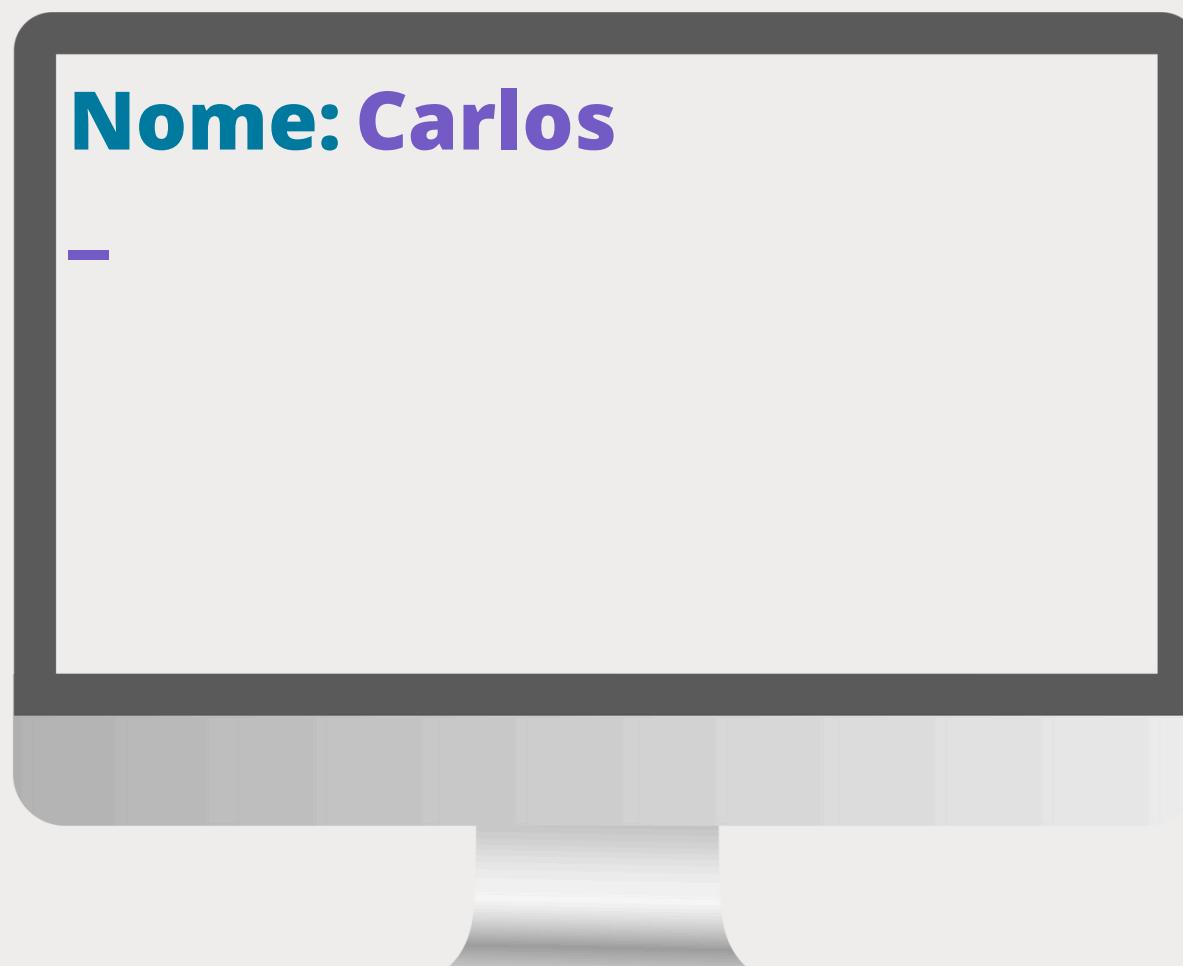
```
nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - int(nascimento)
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```



```
→ nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - int(nascimento)
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```

Nome: _____

```
→ nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - int(nascimento)
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```



```
nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - int(nascimento)
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: _____

```
nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - int(nascimento)
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000

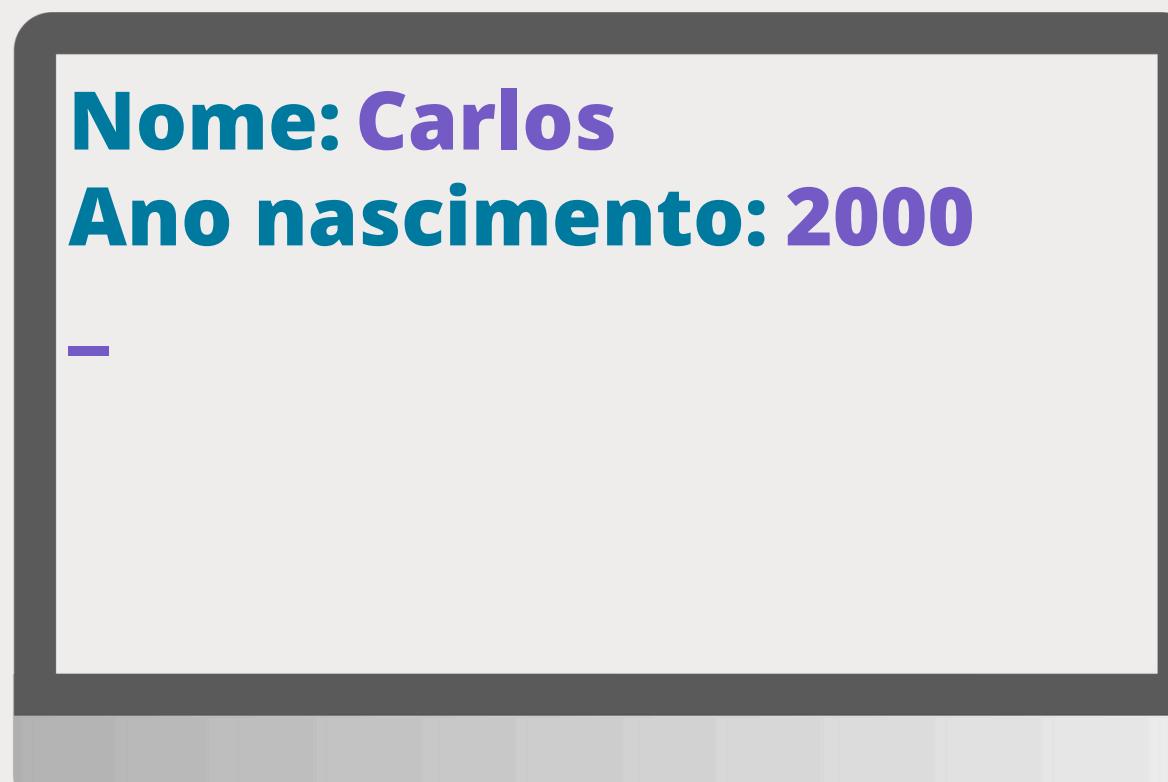
```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 2024 - int(nascimento)  
permissao = idade >= 18  
print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissao}")
```



Nome: Carlos
Ano nascimento: 2000

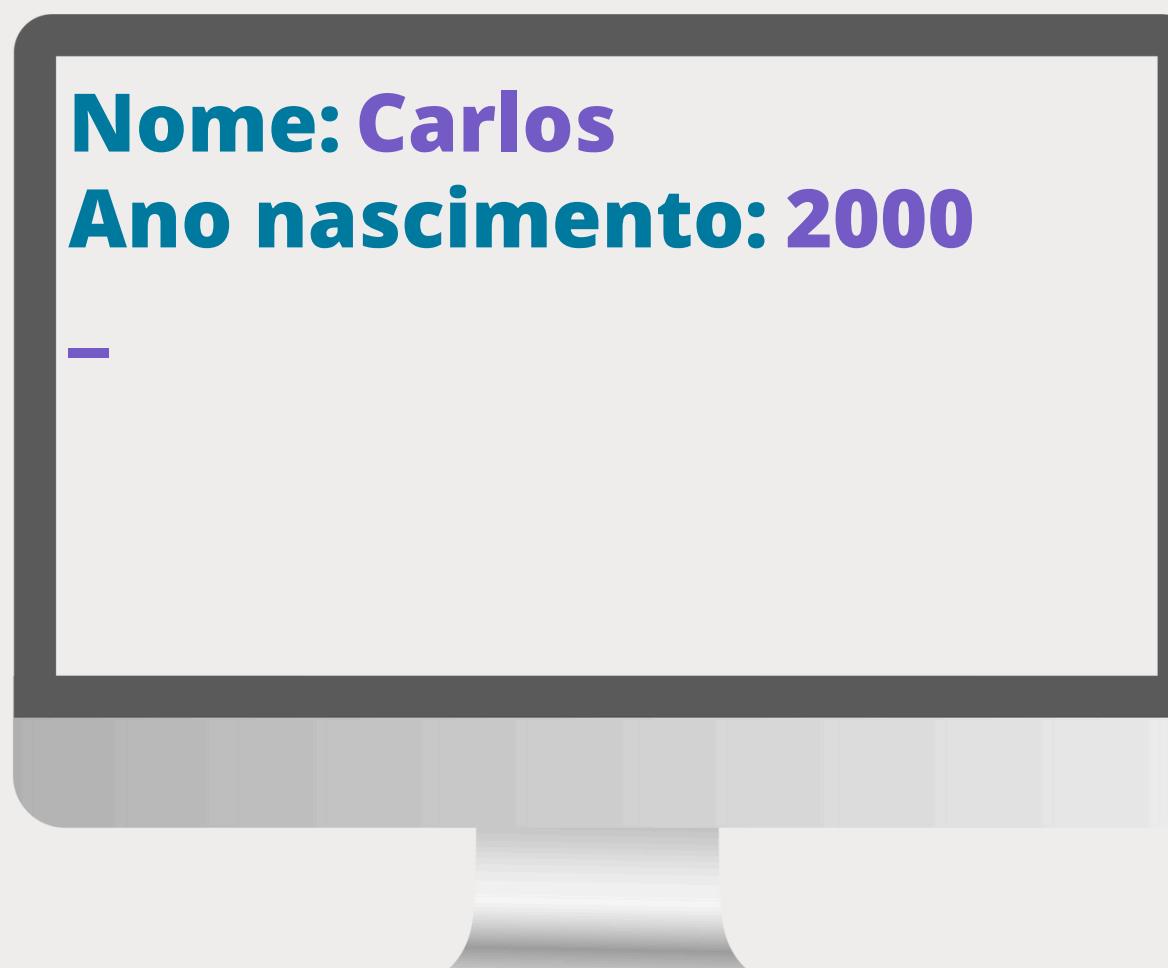
```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 2024 - int("2000")  
permissoao = idade >= 18  
print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissoao}")
```





```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 2024 - int("2000")  
permissoao = idade >= 18  
print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissoao}")
```





```
nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 2024 - 2000
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000

```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
→ idade = 24  
permissao = idade >= 18  
print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24



```
nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 24
permissao = idade >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000

```

nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 24
permissao = 24 >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")

```

24 é maior que 18? SIM (True)

Nome: Carlos
Ano nascimento: 2000

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24



```

nome = input("Nome: ")
nascimento = input("Ano nascimento: ")
idade = 24
permissao = 24 >= 18
print(f"{nome} tem {idade} anos.\n")
print(f"Pode beber: {permissao}")

```

24 é maior que 18? SIM (True)

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24
permissao	bool	True

Nome: Carlos
Ano nascimento: 2000



```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 24  
permissao = True  
→ print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24
permissao	bool	True



```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 24  
permissao = True  
→ print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000
Carlos tem 24 anos.

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24
permissao	bool	True



```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 24  
permissao = True  
→ print(f"{nome} tem {idade} anos.\n")  
print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000
Carlos tem 24 anos.



Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24
permissao	bool	True

```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 24  
permissao = True  
print(f"{nome} tem {idade} anos.\n")  
→ print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000
Carlos tem 24 anos.

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24
permissao	bool	True



```
nome = input("Nome: ")  
nascimento = input("Ano nascimento: ")  
idade = 24  
permissao = True  
print(f"{nome} tem {idade} anos.\n")  
→ print(f"Pode beber: {permissao}")
```

Nome: Carlos
Ano nascimento: 2000
Carlos tem 24 anos.

Pode beber: True

Nome	Tipo	Valor
nome	str	"Carlos"
nascimento	str	"2000"
idade	int	24
permissao	bool	True



<EXERCÍCIO 1>

**Crie um programa que pergunte o nome
do usuário, o time que ele torce, e
mostre na tela:**

“NOME torce para o TIME”



< SOLUÇÃO >

```
nome = input("Nome: ")  
time = input("Time: ")  
print(f"{nome} torce para o {time}")
```



< EXPRESSÕES LÓGICAS >

- **not** : retorna "True" se a condição for falsa, e vice-versa
- **and** : retorna "True" se todas as condições forem verdadeiras
- **or** : retorna "True" se uma ou mais condições forem verdadeiras



< COMANDOS CONDICIONAIS >

**São utilizados para executar trechos
de códigos apenas em alguns casos,
quando uma condição é verdadeira**

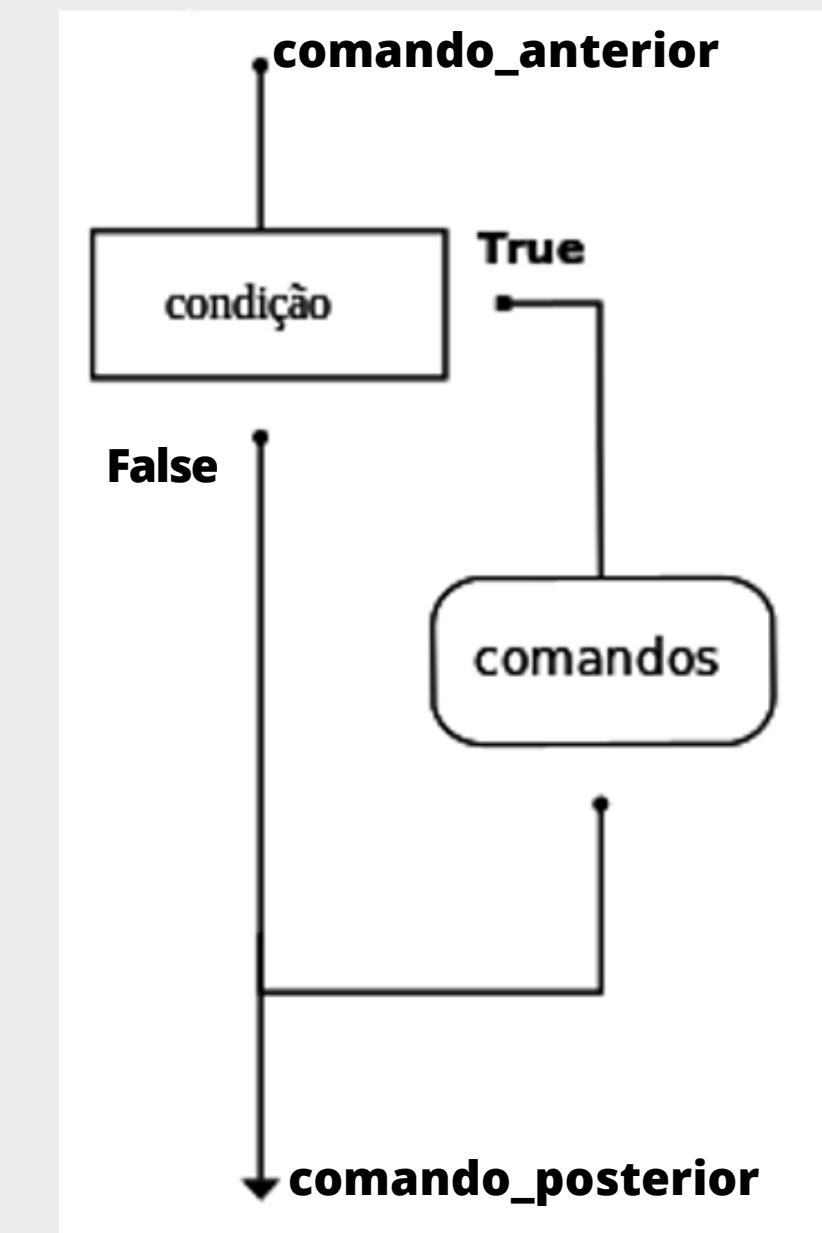


COMANDOS CONDICIONAIS

comando_anterior

if (condição):
comandos

comando_posterior

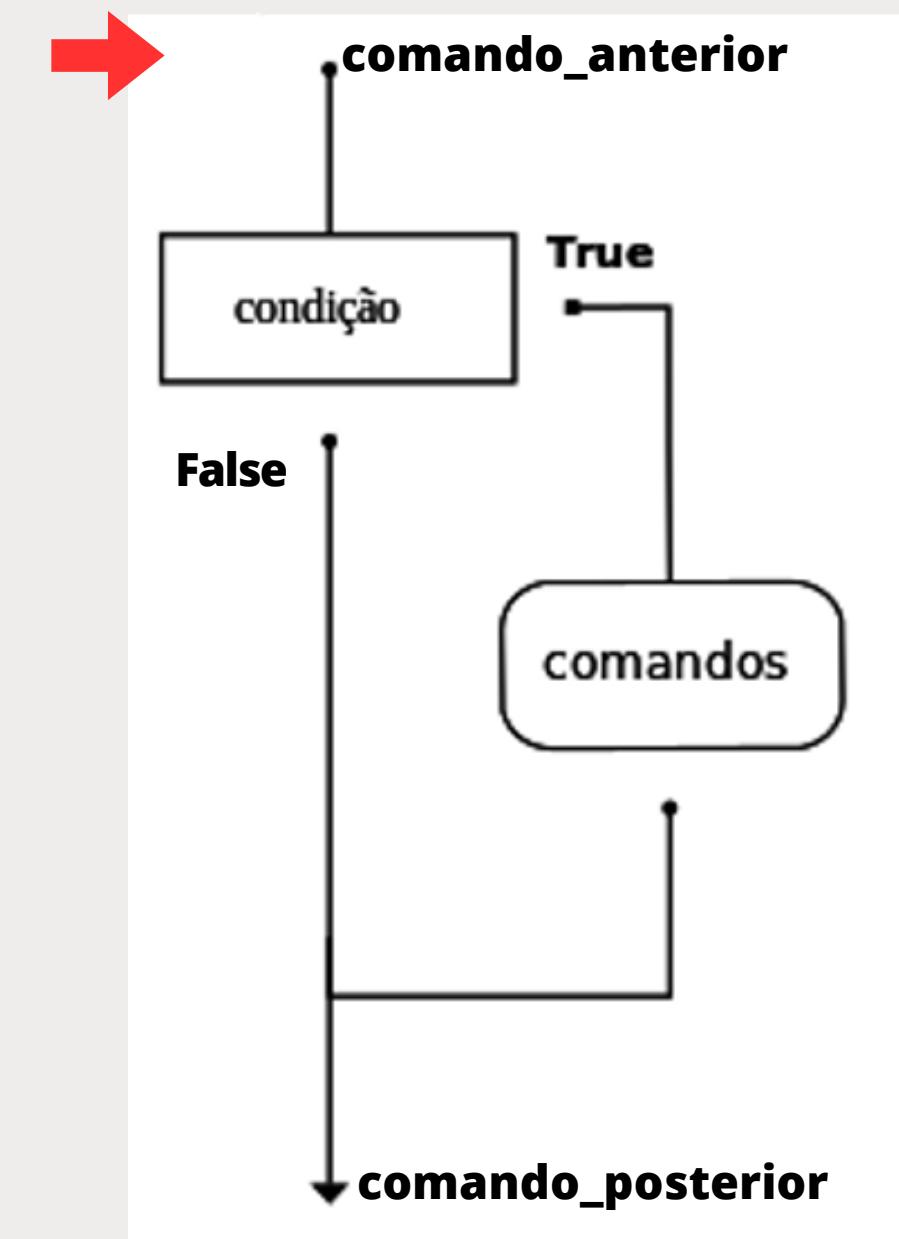


COMANDOS CONDICIONAIS

comando_anterior

```
if (condição):  
    comandos
```

comando_posterior

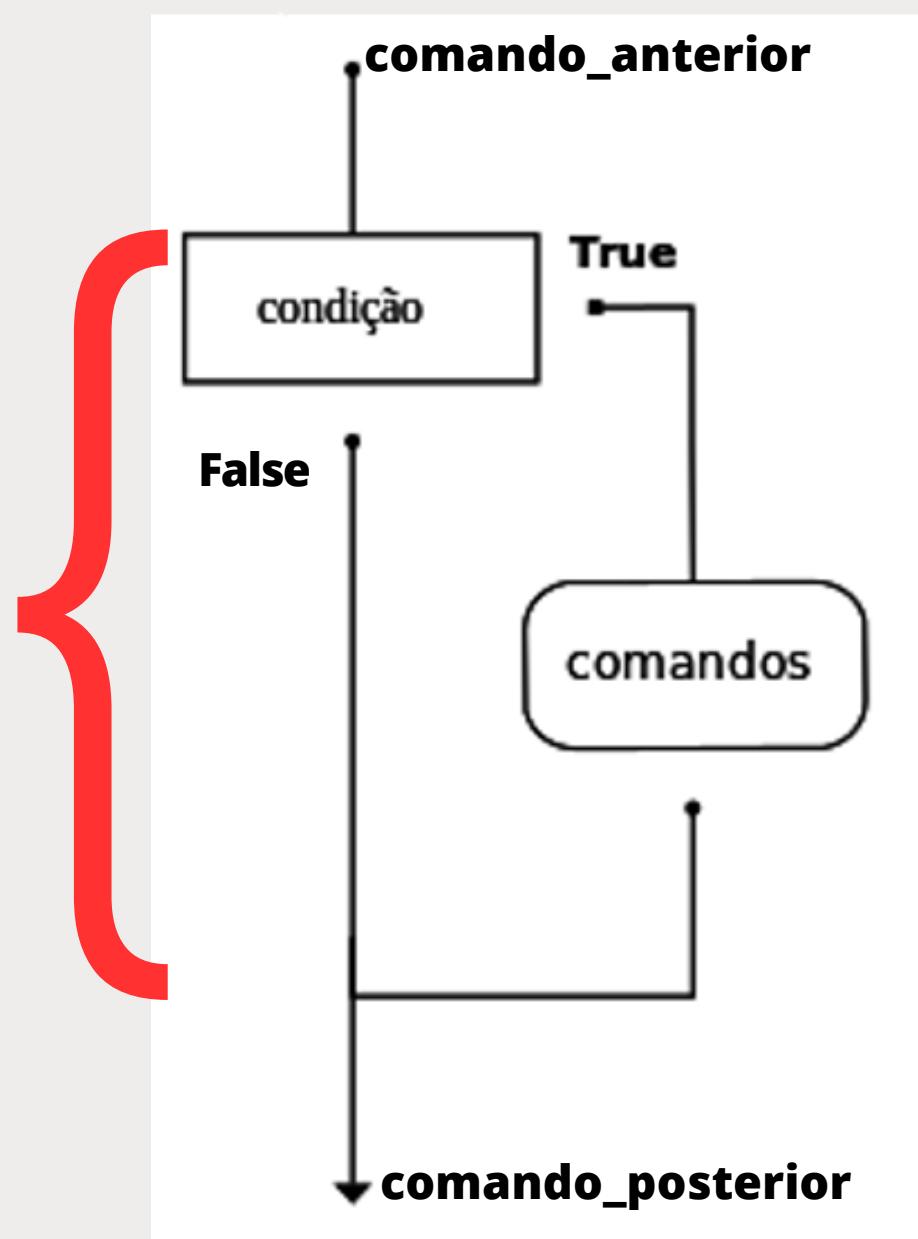


COMANDOS CONDICIONAIS

comando_anterior

```
if (condição):  
    comandos
```

comando_posterior

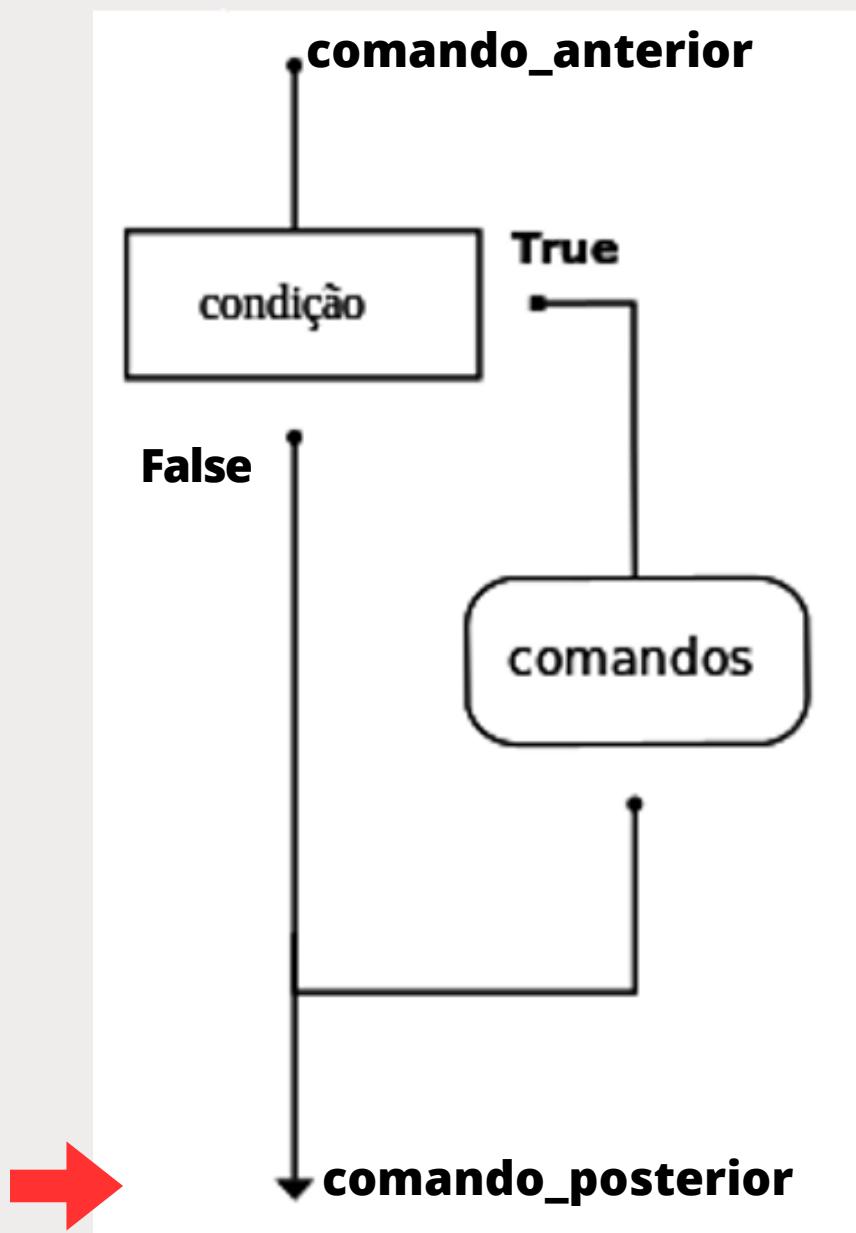


COMANDOS CONDICIONAIS

comando_anterior

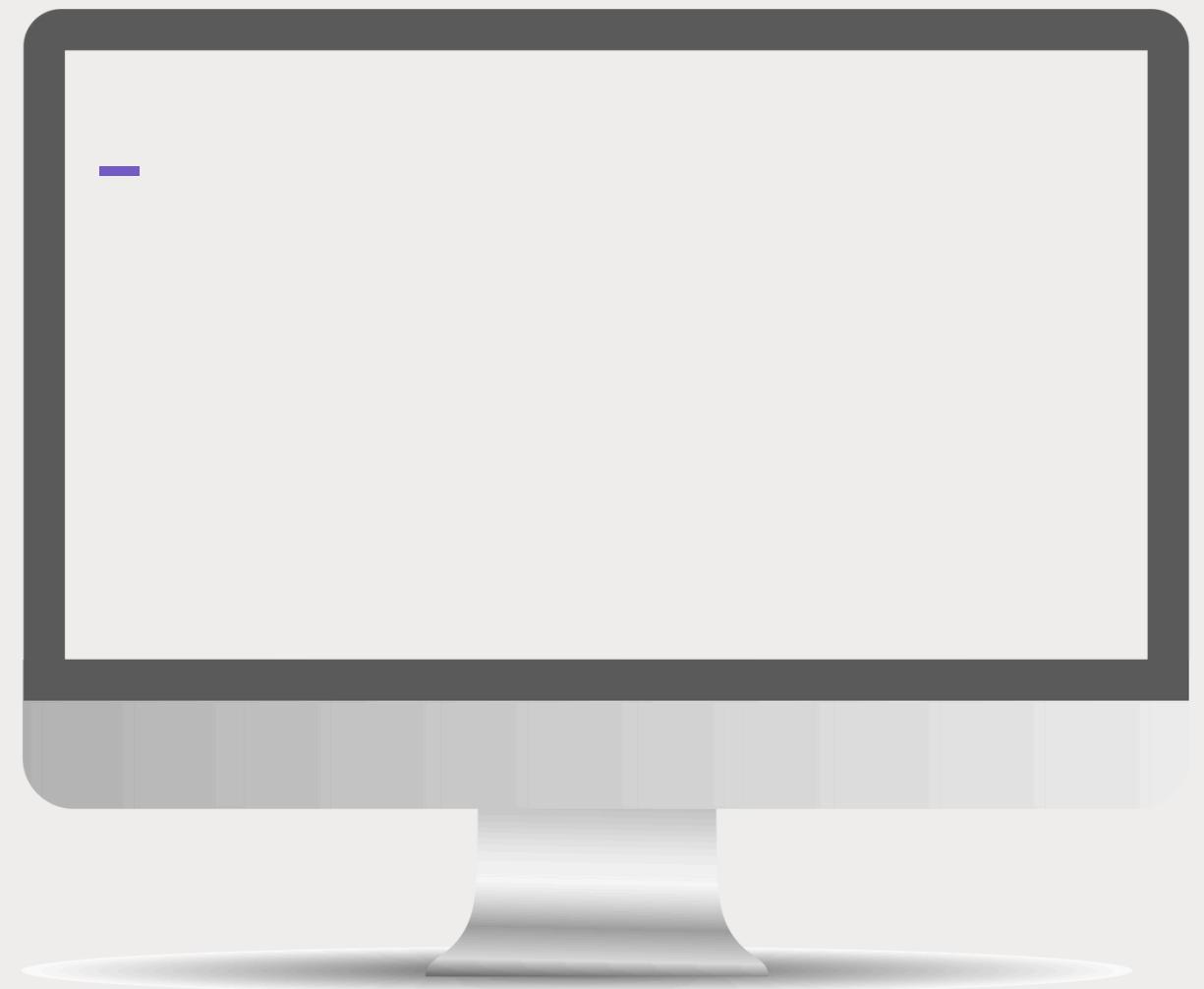
if (condição):
comandos

comando_posterior ←



< EXEMPLO >

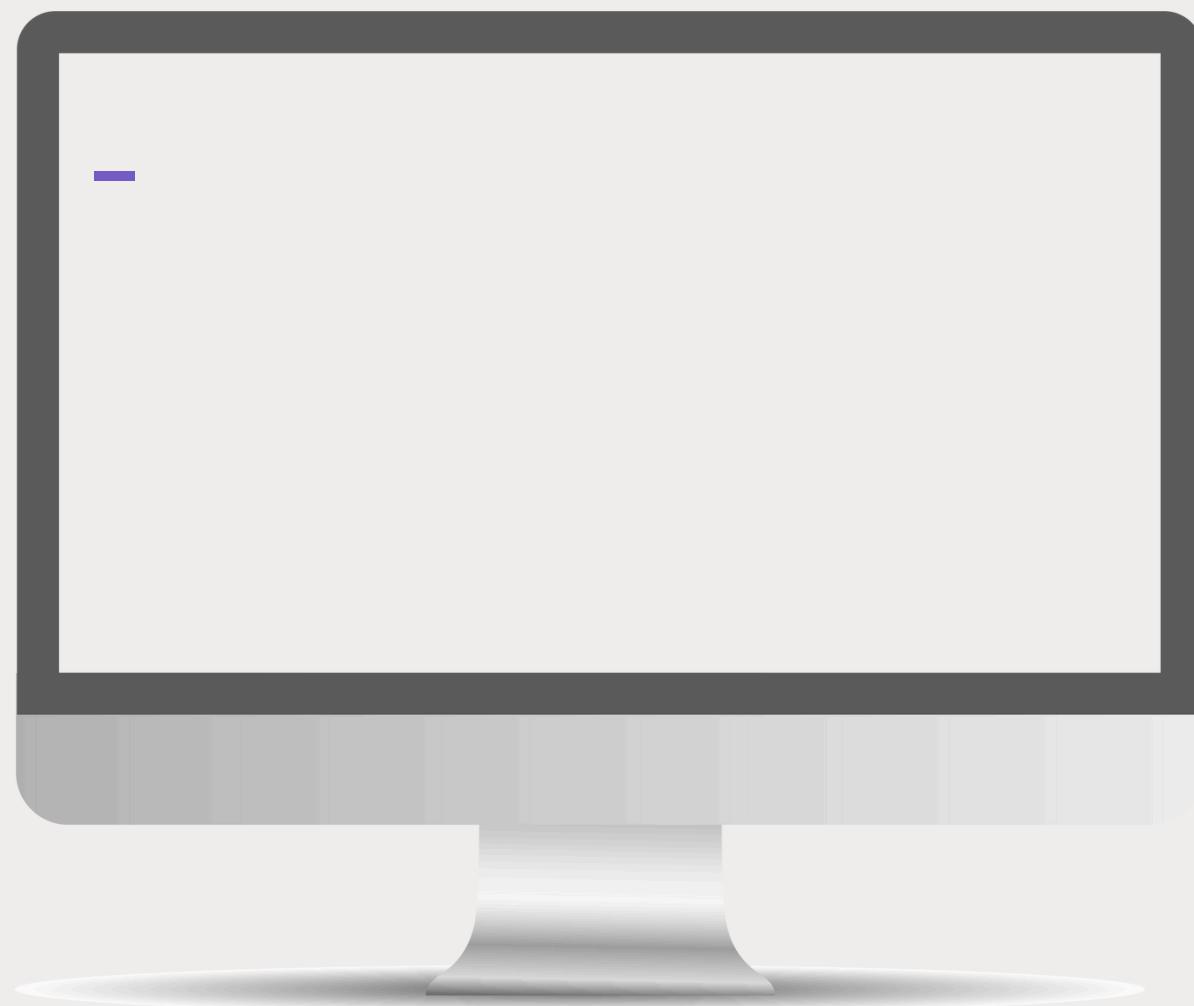
```
valor = 10  
if valor > 5:  
    print('O valor é maior que 5.')
```



< EXEMPLO >

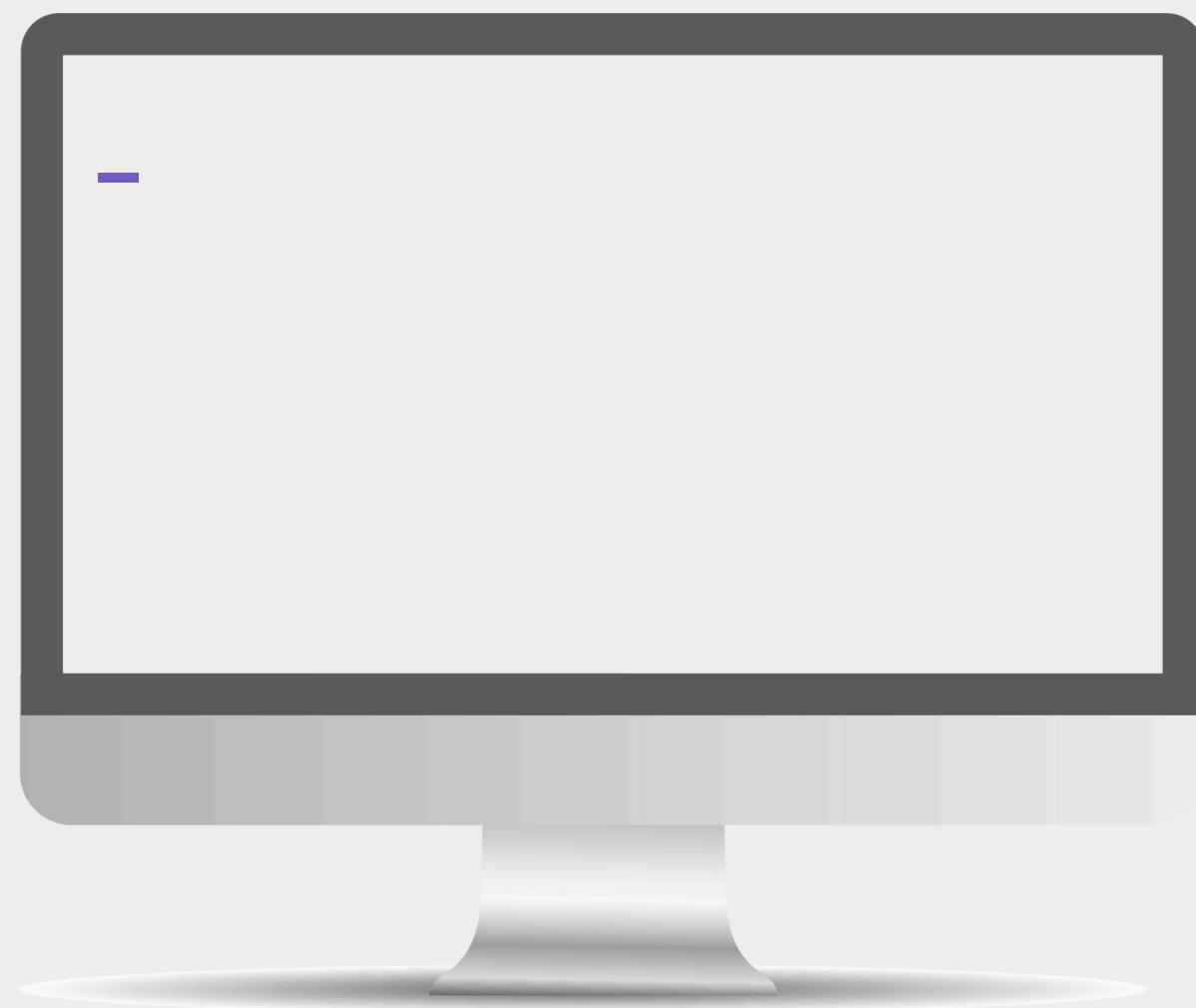
valor é maior que 5?

```
valor = 10
if valor > 5:
    print('O valor é maior que 5.')
```



< EXEMPLO >

```
True  
valor = 10  
if valor > 5:  
    print('O valor é maior que 5.')
```



< EXEMPLO >

```
valor = 10  
if valor > 5:  
    print('O valor é maior que 5.')
```



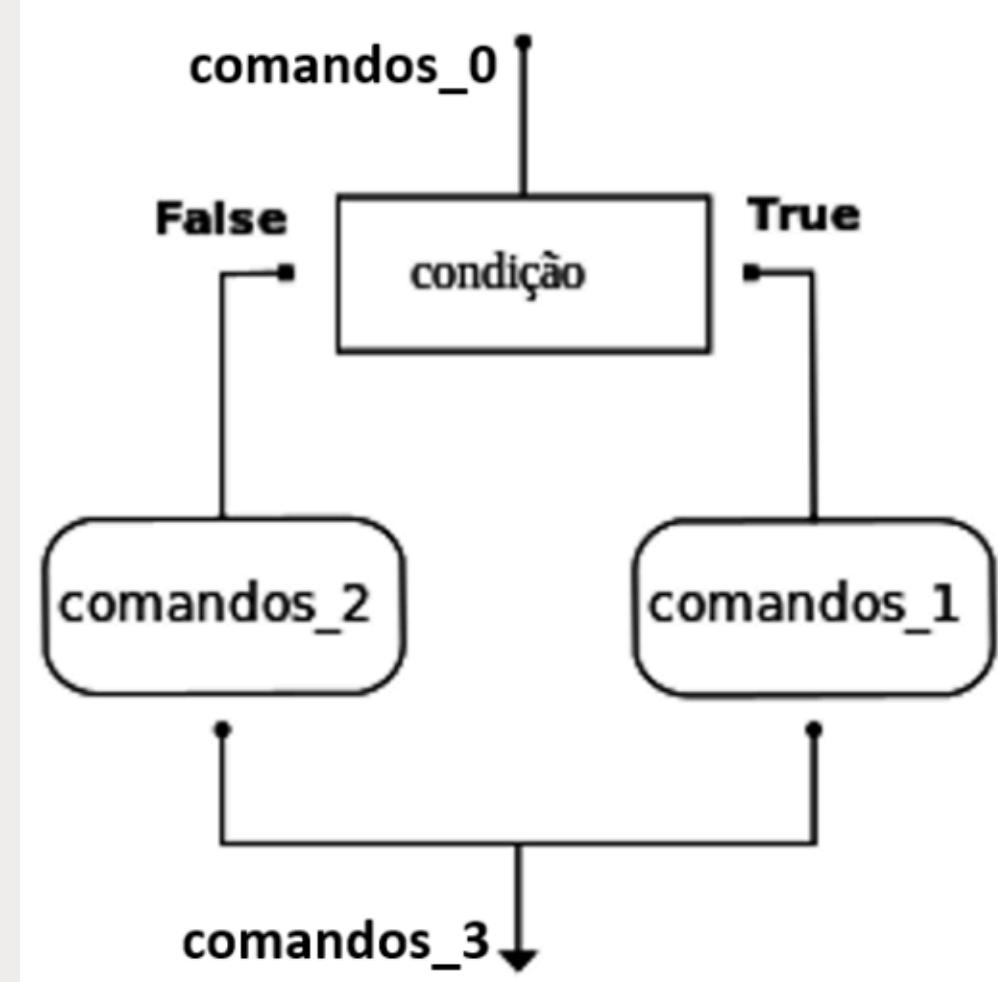
O valor é maior que 5.



COMANDOS CONDICIONAIS

```
comandos_0
if (expressão booleana):
    comandos_1
else:
    comandos_2
comandos_3
```

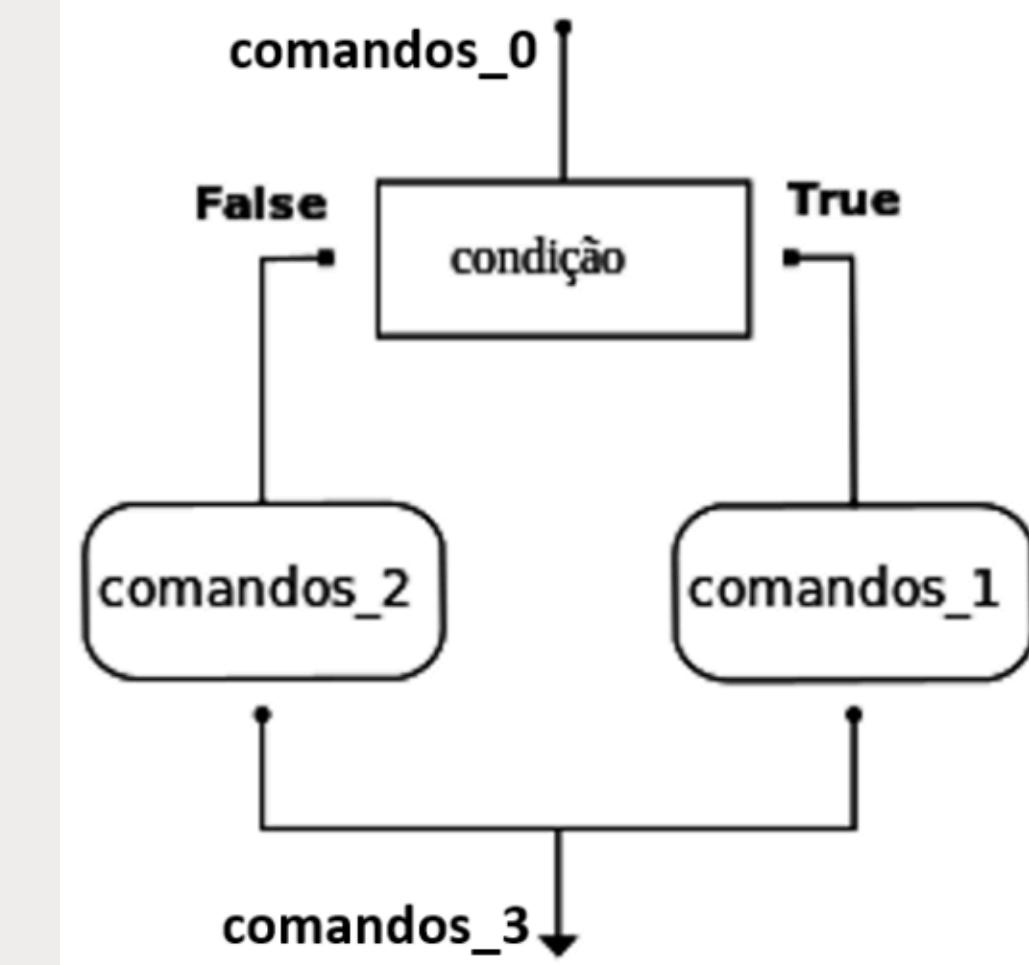
Executado antes
da verificação
condicional



COMANDOS CONDICIONAIS

comandos_0
if (expressão booleana):
 comandos_1
else:
 comandos_2
comandos_3

Condição



COMANDOS CONDICIONAIS

comandos_0

if (expressão booleana):

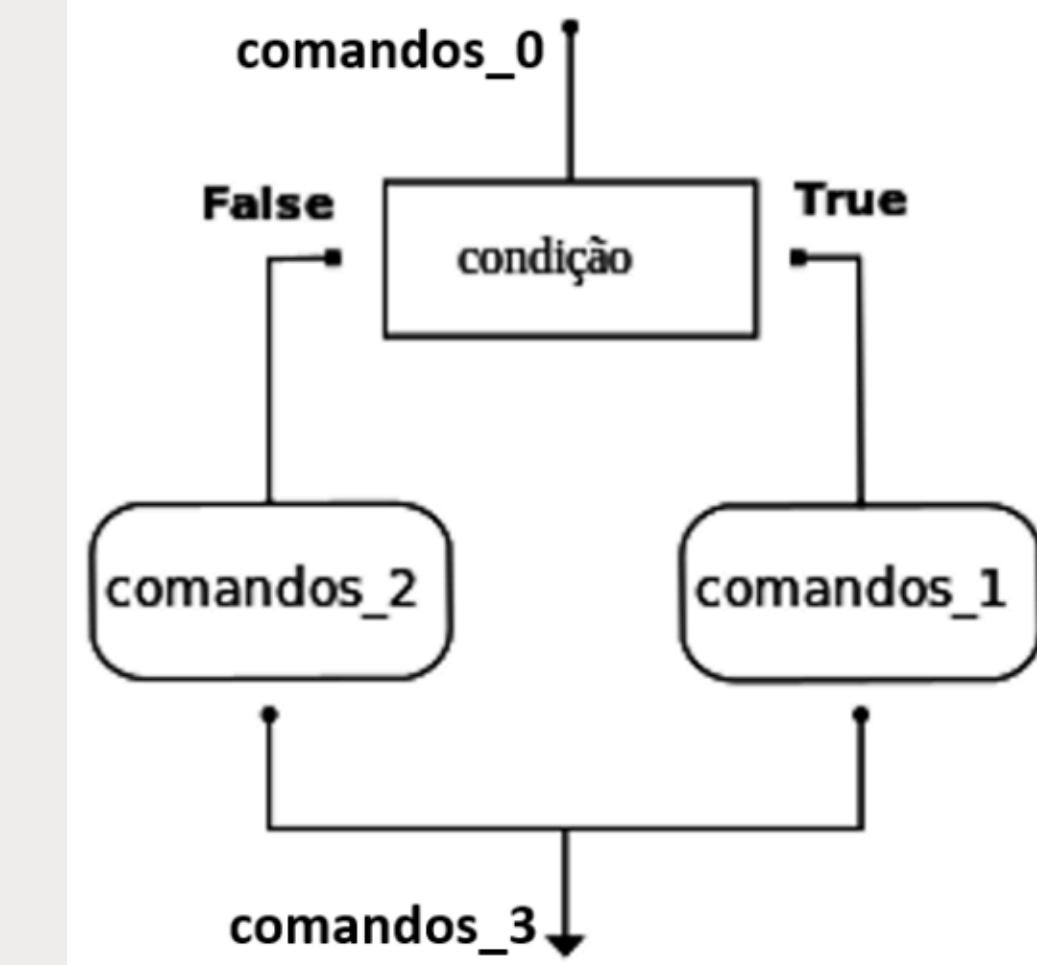
comandos_1

else:

comandos_2

comandos_3

**Executado se
a expressão
booleana for
verdadeira**



COMANDOS CONDICIONAIS

comandos_0

if (expressão booleana):

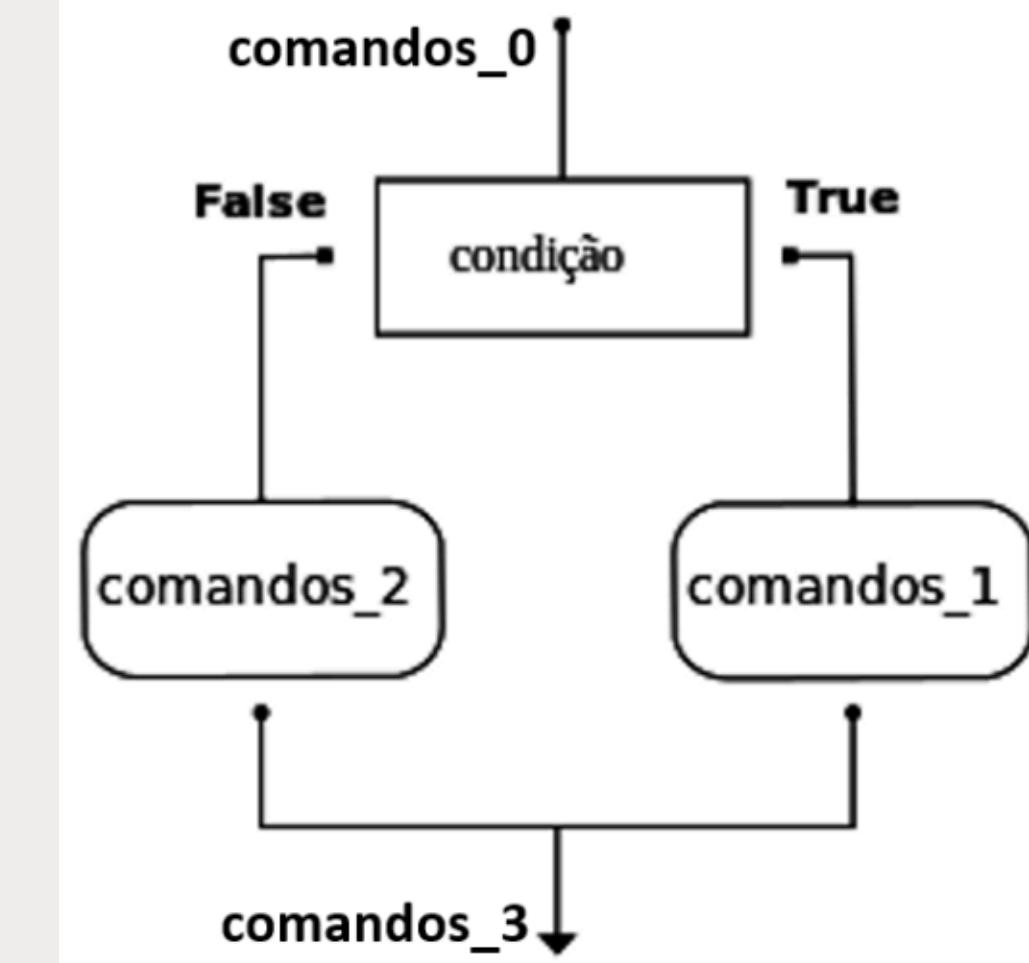
comandos_1

else:

comandos_2

comandos_3

**Executado se
a expressão
booleana NÃO
for verdadeira**



COMANDOS CONDICIONAIS

comandos_0

if (expressão booleana):

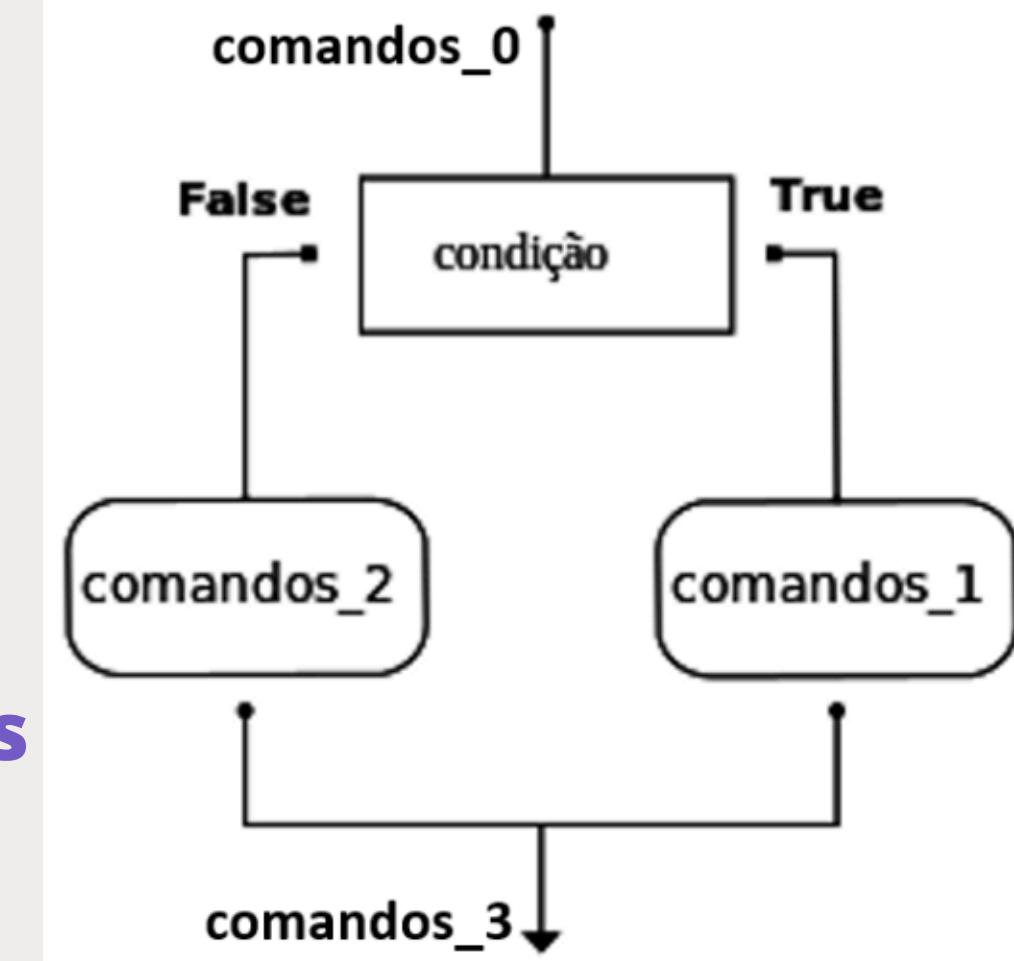
comandos_1

else:

comandos_2

comandos_3

Executado após
a verificação
condicional



< EXEMPLO >

**Leia um número e mostre na tela se
ele é par ou ímpar**

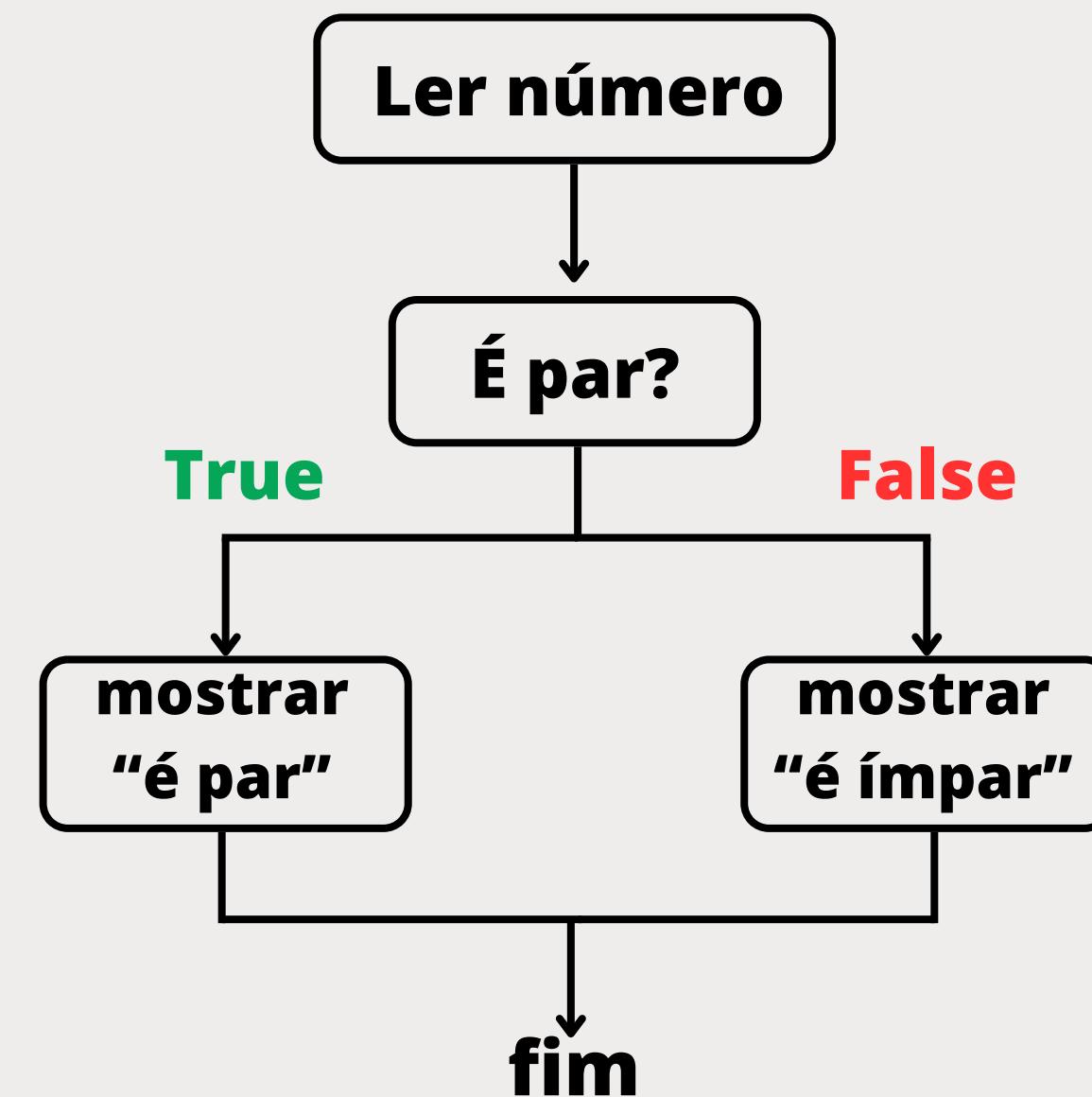


< EXEMPLO >

**Leia um número e mostre na tela se
ele é par ou ímpar**



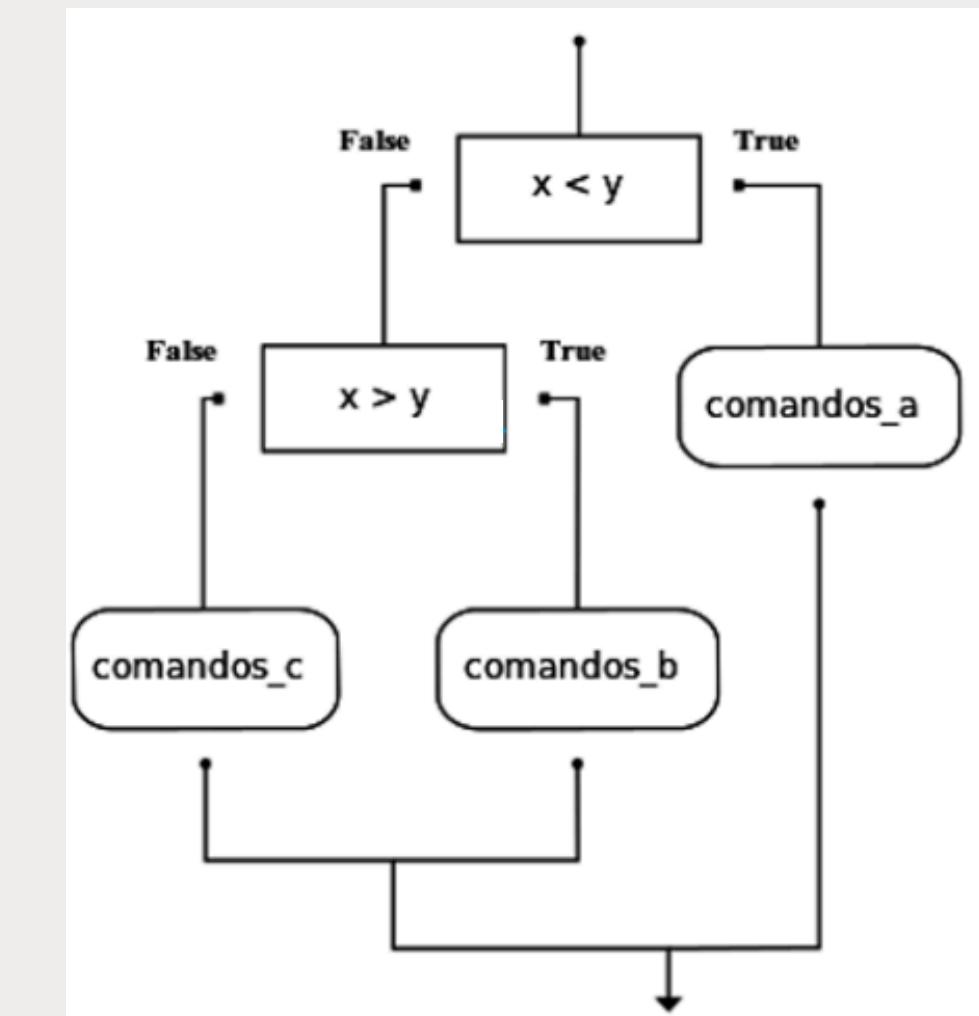
```
n = int(input("número: "))
if n%2 == 0:
    print(n,"é par")
else:
    print(n,"é ímpar")
```



COMANDOS CONDICIONAIS

```
if x < y:  
    comandos_a  
    print("y é maior")  
  
else:  
    if x > y:  
        comandos_b  
        print("x é maior")  
  
    else:  
        comandos_c  
        print("x é igual à y")
```

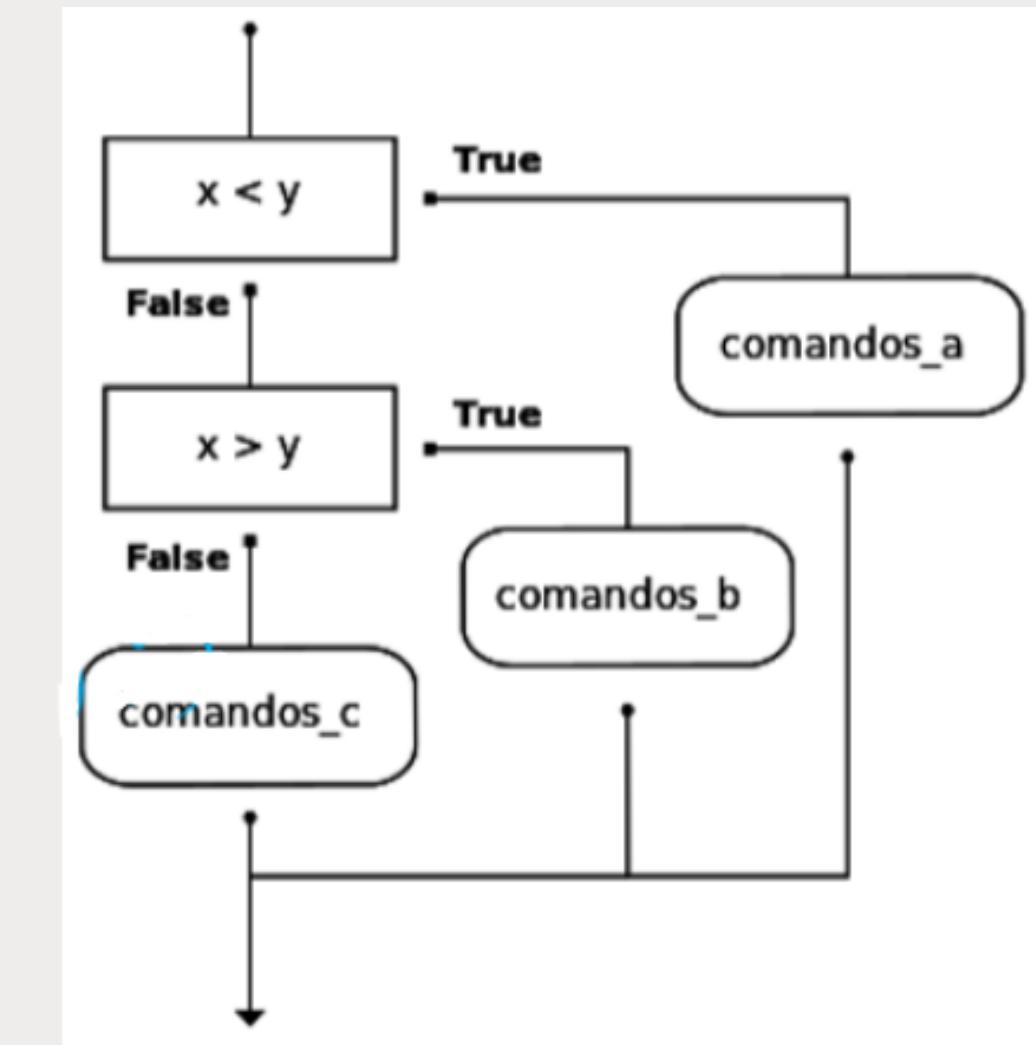
Esse tipo de
condicional é o
aninhado



COMANDOS CONDICIONAIS

```
if x < y:  
    comandos_a  
    print("y é maior")  
elif x > y:  
    comandos_b  
    print("x é maior")  
else:  
    comandos_c  
    print("x é igual à y")
```

Esse tipo de
condicional é o
encadeado



<EXERCÍCIO 2>

**Crie um programa que pergunte o nome
do usuário, a média final dele, sua
porcentagem de presença, e mostre se
ele foi aprovado ou não**



Pra passar: nota igual ou
acima de 6, e presença maior
ou igual à 75%



< SOLUÇÃO >

```
nome = input("Nome: ")
media = float(input("Média: "))
presenca = float(input("Presença (%): "))

if media >= 6 and presenca >= 75:
    print("Aprovado")
else:
    print("Reprovado")
```



<

LISTAS

>

As listas são um tipo de estrutura de dados que permitem armazenar uma coleção de itens em uma única variável.

As listas são mutáveis, o que significa que podem ser modificadas após serem criadas.



< EXEMPLO >

```
alunos = ['Antonio', 'Marina', 'Viktor']
print(alunos[0])
print(alunos[1])
print(alunos[2])
```



```
alunos = ['Antonio', 'Marina', 'Viktor']
```

```
print(alunos[0])
```

```
print(alunos[1])
```

```
print(alunos[2])
```



```
alunos = ['Antonio', 'Marina', 'Viktor']

print(alunos[0])
print(alunos[1])
print(alunos[2])
```



```
alunos = ['Antonio', 'Marina', 'Viktor']

print(alunos[0])
print(alunos[1])
print(alunos[2])
```

'Antonio'

'Marina'



```
alunos = ['Antonio', 'Marina', 'Viktor']

print(alunos[0])
print(alunos[1])
print(alunos[2])
```



< LISTAS >

**Para retornar partes específicas da lista,
podemos indicar uma combinação de
índices, separados por dois pontos, sendo
que se não indicarmos um índice antes
dos dois pontos a seleção começará a
partir do primeiro item, e se não
definirmos algo após os dois pontos a
seleção terminará no último item**



< EXEMPLO >

```
alunos = ['Antonio', 'Marina', 'Viktor']
print(alunos[1:])
print(alunos[:-1])
print(alunos[0:1])
```



```
alunos = ['Antonio', 'Marina', 'Viktor']
```

```
print(alunos[1:])
```

```
print(alunos[:-1])
```

```
print(alunos[0:1])
```



```
alunos = ['Antonio', 'Marina', 'Viktor']
```

```
print(alunos[1:])
```

```
print(alunos[:-1])
```

```
print(alunos[0:1])
```

[‘Marina’, ‘Viktor’]

```
alunos = ['Antonio', 'Marina', 'Viktor']

print(alunos[1:])
print(alunos[:-1])
print(alunos[0:1])
```

**[‘Marina’, ‘Viktor’]
[‘Antonio’, ‘Marina’]**

```
alunos = ['Antonio', 'Marina', 'Viktor']

print(alunos[1:])
print(alunos[:-1])
print(alunos[0:1])
```

**[‘Marina’, ‘Viktor’]
[‘Antonio’, ‘Marina’]
[‘Antonio’]**

< FUNÇÕES DE LISTAS >

APPEND

Anexar um item no final da lista

```
alunos = ['Antonio', 'Marina', 'Viktor']
```

```
alunos.append('Arthur')
```

=

```
alunos = ['Antonio', 'Marina', 'Viktor', 'Arthur']
```



< FUNÇÕES DE LISTAS >

SORT

Ordena em ordem crescente

minha_lista = [1, 90, 56, 100]

minha_lista.sort()

=

minha_lista = [1, 56, 90, 100]



< FUNÇÕES DE LISTAS >

REVERSE

Ordena em ordem decrescente

minha_lista = [1, 90, 56, 100]

minha_lista.reverse()

=

minha_lista = [100, 90, 56, 1]



< FUNÇÕES DE LISTAS >

REMOVE

Remove um item com base no valor

```
minha_lista = [1, 90, 56, 100]
```

```
minha_lista.remove(56)
```

=

```
minha_lista = [1, 90, 100]
```



< FUNÇÕES DE LISTAS >

POP

Remove um item com base no índice

minha_lista = [1, 90, 56, 100]

minha_lista.pop(3)

=

minha_lista = [1, 90, 100]



< FUNÇÕES DE LISTAS >

LEN

Comprimento de uma lista

minha_lista = [1, 90, 56, 100]

len(minha_lista)

=

4



< FUNÇÕES DE LISTAS >

MAX E MIN

Valores máximos e mínimos

minha_lista = [1, 90, 56, 100]

max(minha_lista)

=

100



< FUNÇÕES DE LISTAS >

COUNT

Retorna o número de ocorrências de determinado objeto

minha_lista = [1, 90, 56, 90, 100]

minha_lista.count(90)

=

2



< OPERAÇÕES COM LISTAS >

CONCATENANDO

```
minha_lista = [1, 90, 56, 90, 100]
```

```
outra_lista = [5, 6, 8]
```

```
lista_final = minha_lista + outra_lista
```

=

```
lista_final = [1, 90, 56, 90, 100, 5, 6, 8]
```



< OPERAÇÕES COM LISTAS >

PERCORRENDO

```
minha_lista = [1, 90]
for i in range(len(minha_lista)):
    print(minha_lista[i])
```

=

>> 1

>> 90



< OPERAÇÕES COM LISTAS >

PERCORRENDO

minha_lista = [1, 90]

for i in minha_lista:
print(i)

=

>> 1

>> 90



< EXERCÍCIO >

Escreva um programa que receba uma lista de 2 inteiros via teclado, em seguida o programa deve solicitar um número e informar se o número também está na lista ou não.



< SOLUÇÃO >

```
lista = []
num1 = input('Digite um número: ')
lista.append(num1)
num2 = input('Digite um número: ')
lista.append(num2)

numero_teste = input('Digite um número a verificar: ')

for j in lista:
    if j == numero_teste:
        print('O número está na lista!')
        exit()
print('O número não está na lista :/')
```

< O QUE VIMOS ATÉ AGORA? >

- ❖ **Declaração de variáveis**
- ❖ **Função print()**
- ❖ **Tipos de variáveis**
 - ❖ int
 - ❖ float
 - ❖ bool
 - ❖ string
- ❖ **Operadores matemáticos**
 - ❖ + - / * // % **
- ❖ **Operadores de comparação**
- ❖ **Expressões lógicas**
- ❖ **Comandos condicionais**
 - ❖ if, else, elif



< O QUE VEREMOS AULA QUE VEM? >

- ◆ **Comandos de repetição**
 - ▷ **while()**
 - ▷ **for()**
- ◆ **Declaração de funções**



< APOIO >





MUITO
OBRIGADO!

