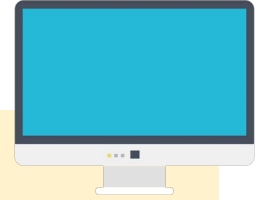


Aula 3 - Laços

Maycon Sambinelli

Fluxo de Execução

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

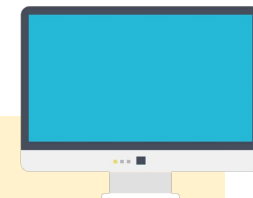
```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

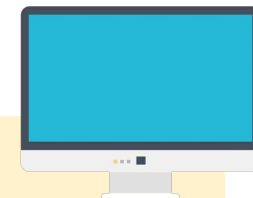
```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```



```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

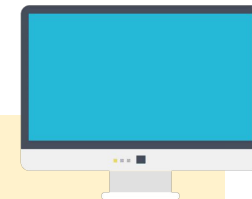
```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
    (n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

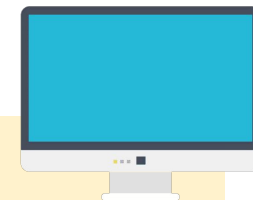
```
n1 = 10
```

```
n2 = 15
```

Definition (Definição)



Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
➔ n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

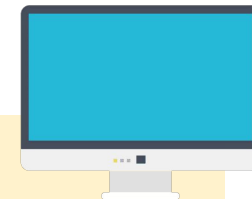
```
➔ (n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

```
função soma(a, b)
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```



```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```



```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
```

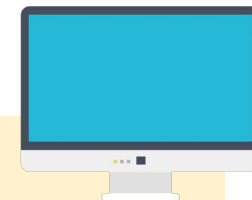
```
n2 = 15
```

```
função soma(a, b)
```

```
    a = 10
```

```
    b = 15
```


Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

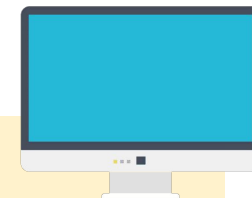
```
função soma(a, b)
```

```
    a = 10
```

```
    b = 15
```



Fluxo de Execução de um Programa



```
n1 = int(input())
n2 = int(input())
def maior(a, b):
    if a > b: False
        return a
    else:
        return b
```



```
n1 = 10
n2 = 15
função soma(a, b)
    a = 10
    b = 15
```

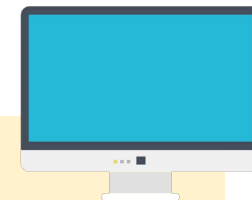


```
n12 = maior(n1, n2)
n3 = int(input())
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
      (n123, n1, n2, n3))
```



Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

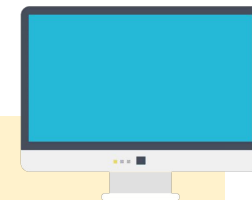
```
função soma(a, b)
```

```
    a = 10
```

```
    b = 15
```



Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
➔ n12 = maior(n1, n2) ⇒ 15
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

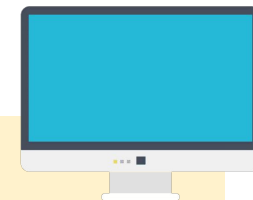
```
➔ (n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

```
função soma(a, b)
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```



```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

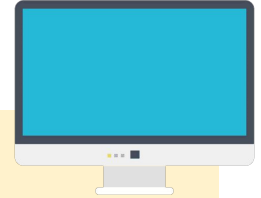
```
n1 = 10
```

```
n2 = 15
```

```
função soma(a, b)
```

```
n12 = 15
```

Fluxo de Execução de um Programa



```
n1 = int(input())
n2 = int(input())
def maior(a, b):
    if a > b:
        return a
    else:
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```



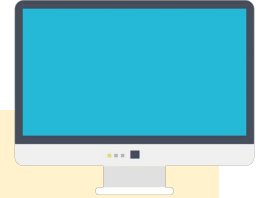
```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
n2 = 15
função soma(a, b)
n12 = 15
n3 = 7
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```



```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```



```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

```
função soma(a, b)
```

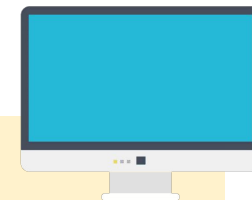
```
    a = 15
```

```
    b = 7
```

```
n12 = 15
```

```
n3 = 7
```

Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
    (n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

```
função soma(a, b)
```

```
    a = 15
```

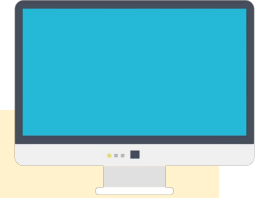
```
    b = 7
```

```
n12 = 15
```

```
n3 = 7
```



Fluxo de Execução de um Programa



```
n1 = int(input())
```

```
n2 = int(input())
```

```
def maior(a, b):
```

```
    if a > b:  $\Rightarrow$  True
```

```
        return a
```

```
    else:
```

```
        return b
```

```
n12 = maior(n1, n2)
```

```
n3 = int(input())
```

```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
```

```
(n123, n1, n2, n3))
```

```
n1 = 10
```

```
n2 = 15
```

```
função soma(a, b)
```

```
    a = 15
```

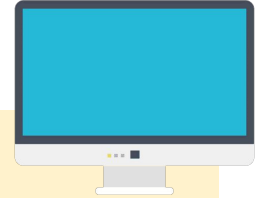
```
    b = 7
```

```
n12 = 15
```

```
n3 = 7
```



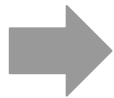
Fluxo de Execução de um Programa



```
n1 = int(input())
n2 = int(input())
def maior(a, b):
    if a > b:
        return a
    else:
        return b
```



```
n12 = maior(n1, n2)
n3 = int(input())
```

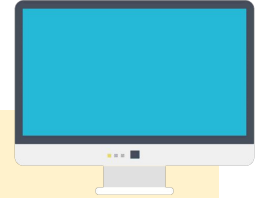


```
n123 = maior(n12, n3)
```

```
print("O %d eh o maior valor entre %d, %d, %d" %
      (n123, n1, n2, n3))
```

```
n1 = 10
n2 = 15
função soma(a, b)
    a = 15
    b = 7
n12 = 15
n3 = 7
```


Fluxo de Execução de um Programa



```
n1 = int(input())
n2 = int(input())
def maior(a, b):
    if a > b:
        return a
    else:
        return b
```

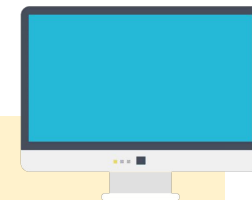
```
n1 = 10
n2 = 15
função soma(a, b)
n12 = 15
n3 = 7
```

```
n12 = maior(n1, n2)
n3 = int(input())
```

 n123 = maior(n12, n3) \Rightarrow 15


```
print("O %d eh o maior valor entre %d, %d, %d" %
      (n123, n1, n2, n3))
```

Fluxo de Execução de um Programa



```
n1 = int(input())
n2 = int(input())
def maior(a, b):
    if a > b:
        return a
    else:
        return b
n12 = maior(n1, n2)
n3 = int(input())
n123 = maior(n12, n3)
```

```
n1 = 10
n2 = 15
função soma(a, b)
n12 = 15
n3 = 7
n123 = 15
```

 `print("O %d eh o maior valor entre %d, %d, %d" %`
` (n123, n1, n2, n3))`

Precedência de Operadores

Tabela de precedência de operadores

Operadores	Nomes
+x, -x	Unários
x ** y	Potência (associativo à direita)
x * y, x / y, x // y, x % y	Multiplicação, divisão, divisão inteira, e módulo
x + y, x - y	Adição e Subtração
x < y, x <= y, x > y, x >= y, x == y, x != y	Comparações
not x	Negação lógica
x and y	e lógico
x or y	ou lógico

* Do maior para o menor

Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```


Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

|

-

Exemplo de precedência de operadores

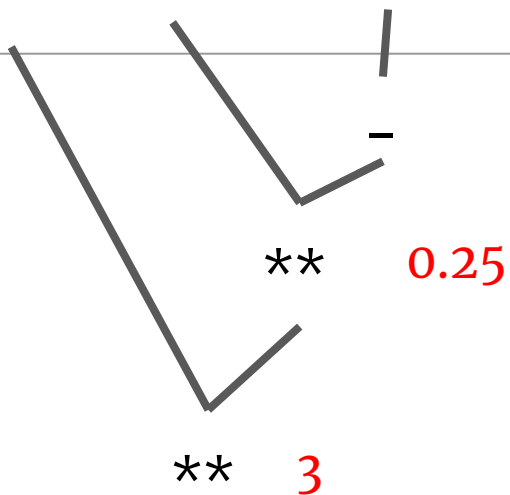
```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```



** 0.25

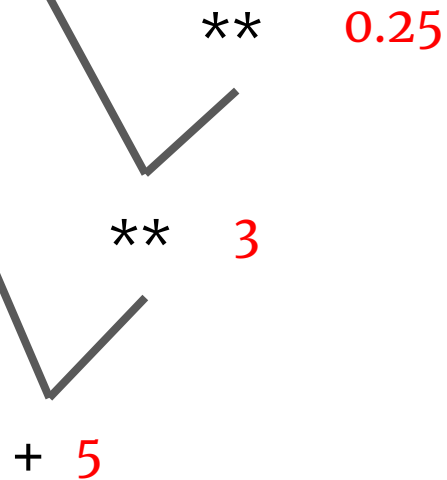
Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```



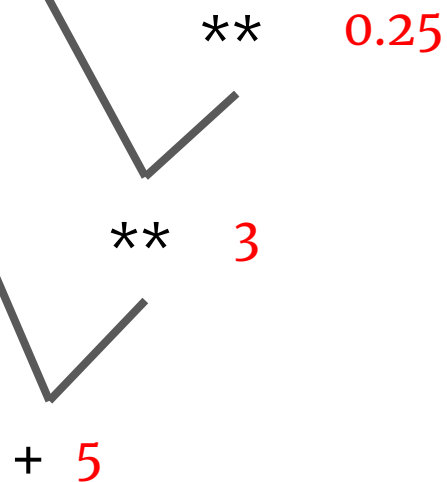
Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```



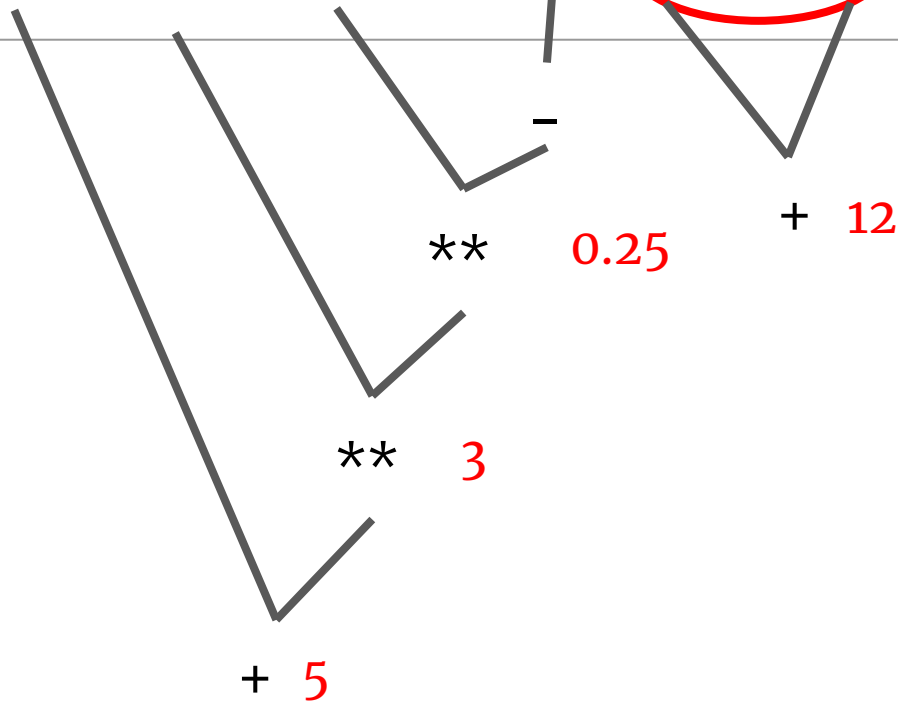
Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```



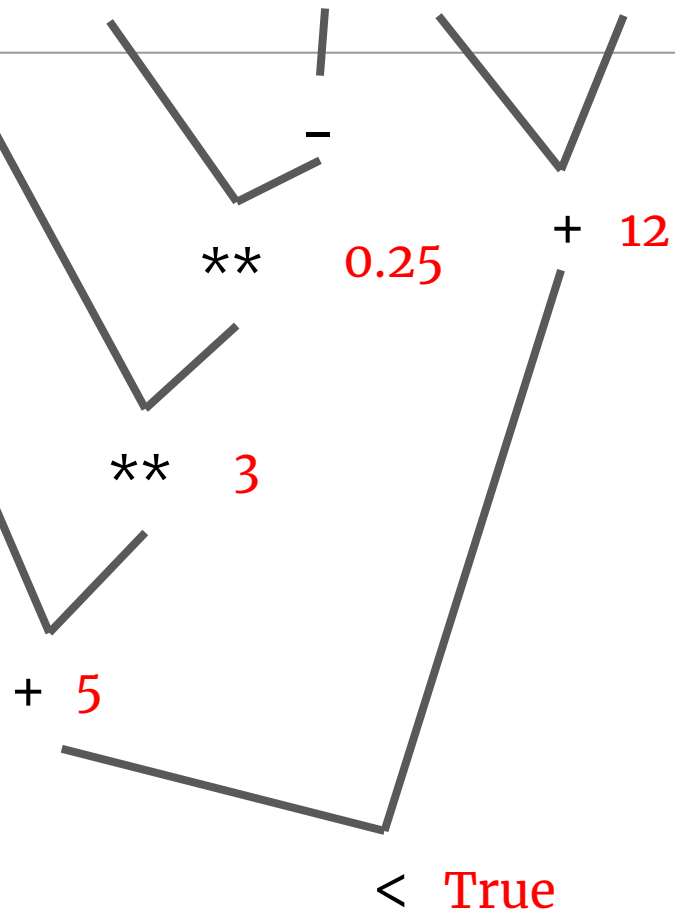
Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

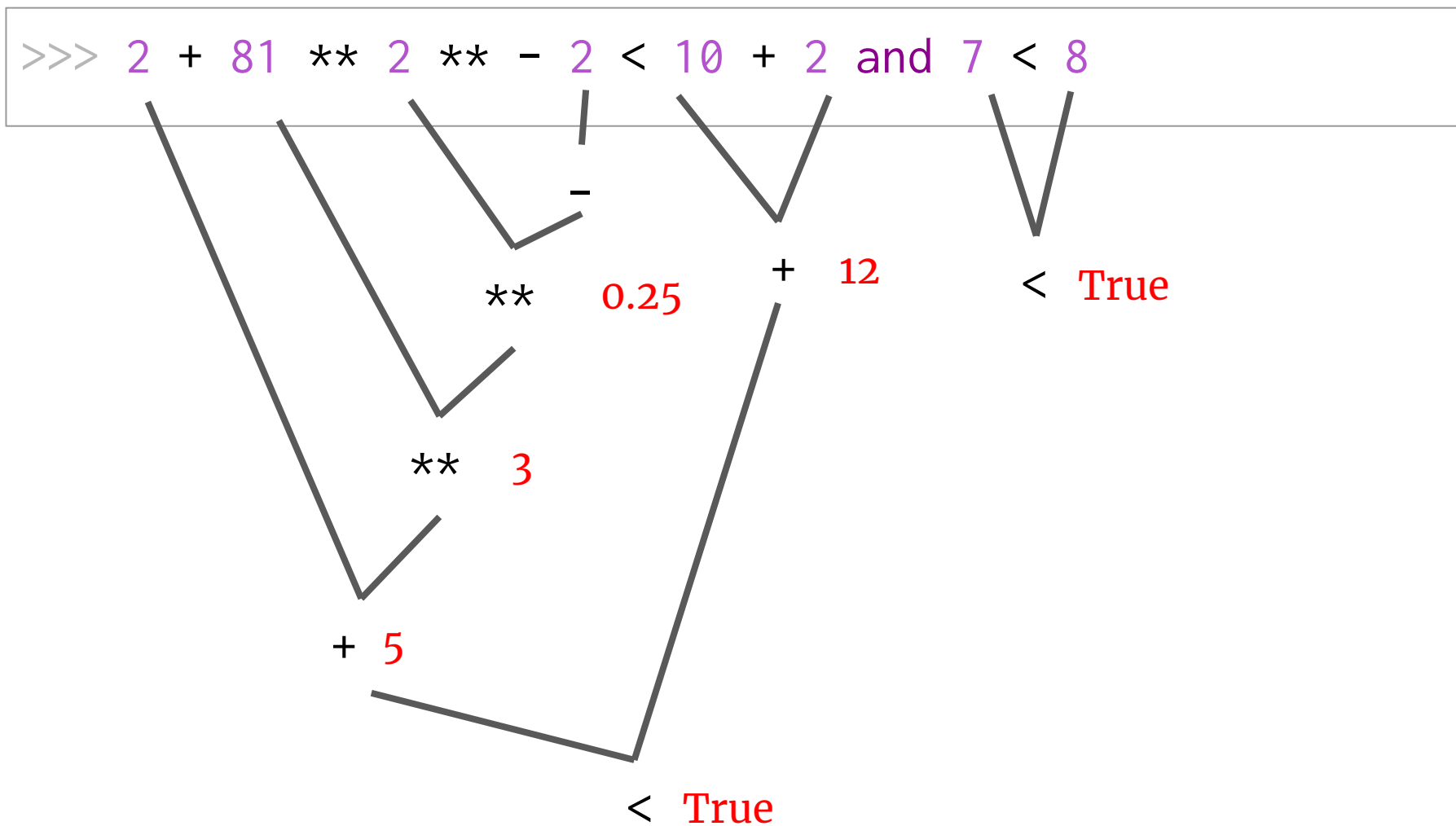


Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```



Exemplo de precedência de operadores



Exemplo de precedência de operadores

```
>>> 2 + 81 ** 2 ** - 2 < 10 + 2 and 7 < 8
```

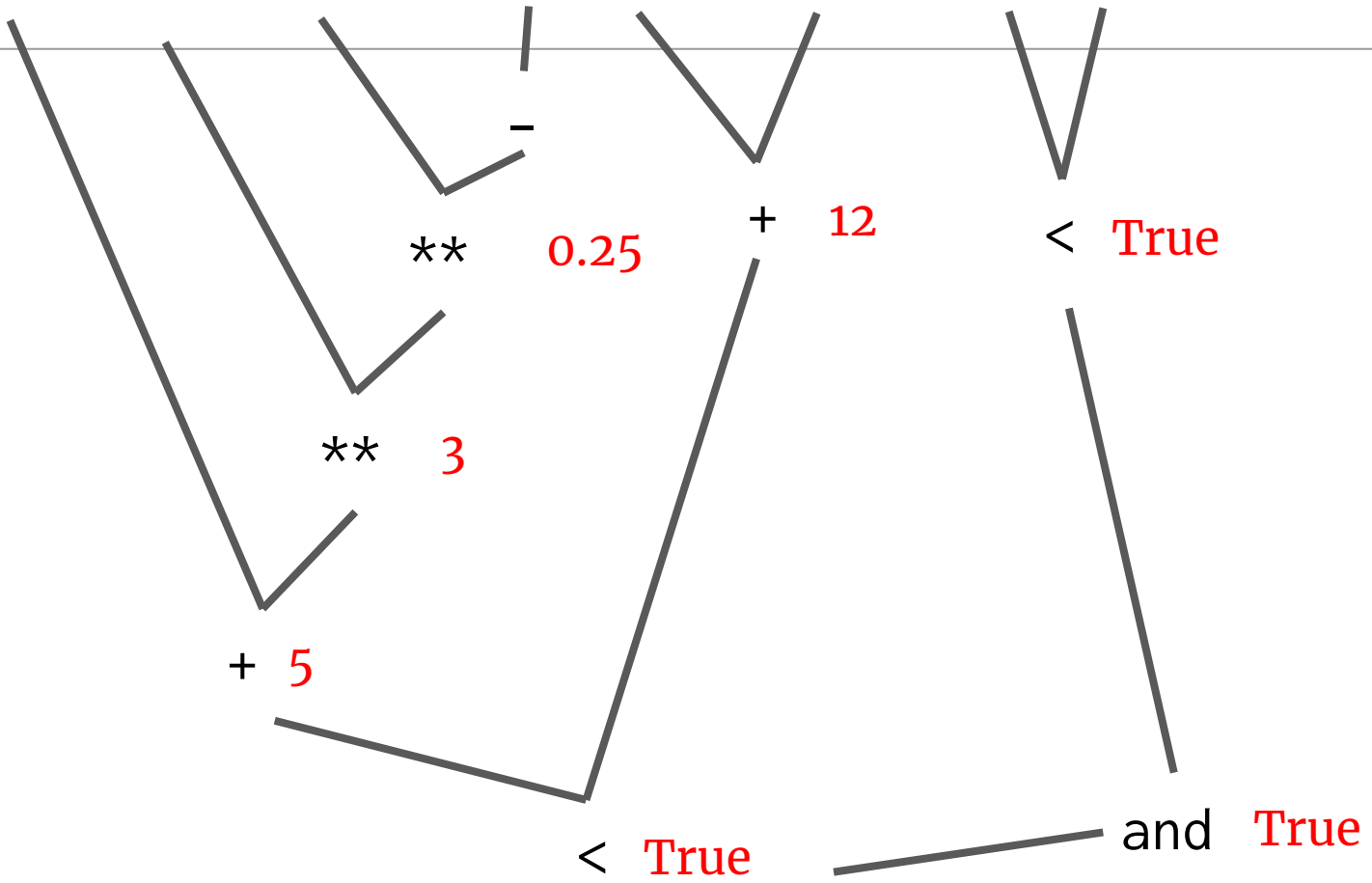


Tabela verdade

P	Q	P and Q
True	True	True
True	False	False
False	True	False
False	False	False

P	Q	P or Q
True	True	True
True	False	True
False	True	True
False	False	False

P	not P
True	False
False	True

Tabela de precedência de operadores

Operadores	Nomes
+x, -x	Unários
x ** y	Potência (associativo à direita)
x * y, x / y, x // y, x % y	Multiplicação, divisão, divisão inteira, e módulo
x + y, x - y	Adição e Subtração
x < y, x <= y, x > y, x >= y, x == y, x != y	Comparações
not x	Negação lógica
x and y	e lógico
x or y	ou lógico

* Do maior para o menor

Exemplo de precedência de operadores

```
>>> False or False and True or not True or False
```

Exemplo de precedência de operadores

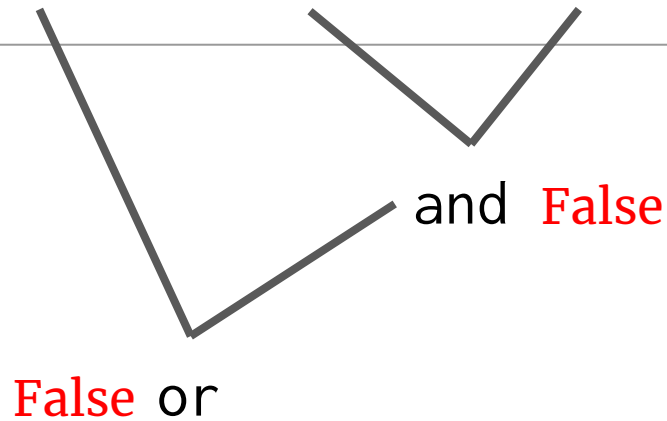
```
>>> False or False and True or not True or False
```



and **False**

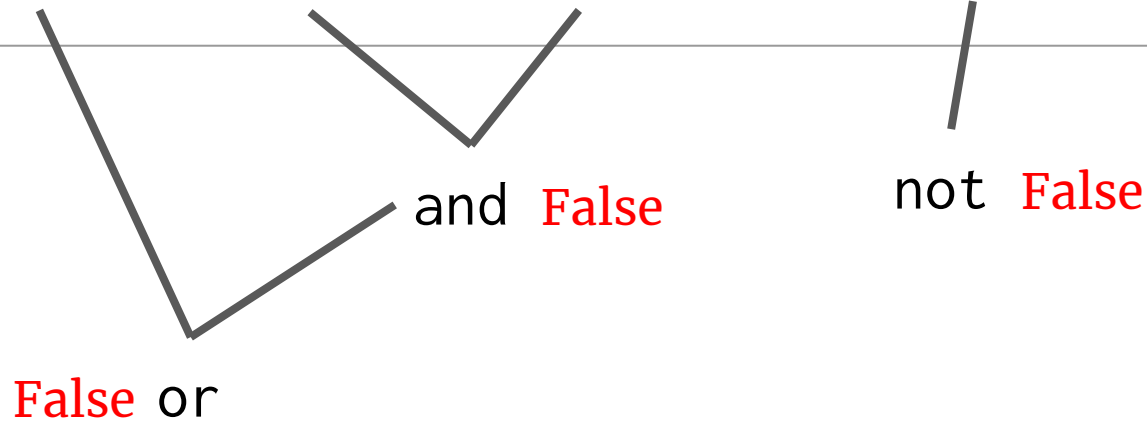
Exemplo de precedência de operadores

```
>>> False or False and True or not True or False
```



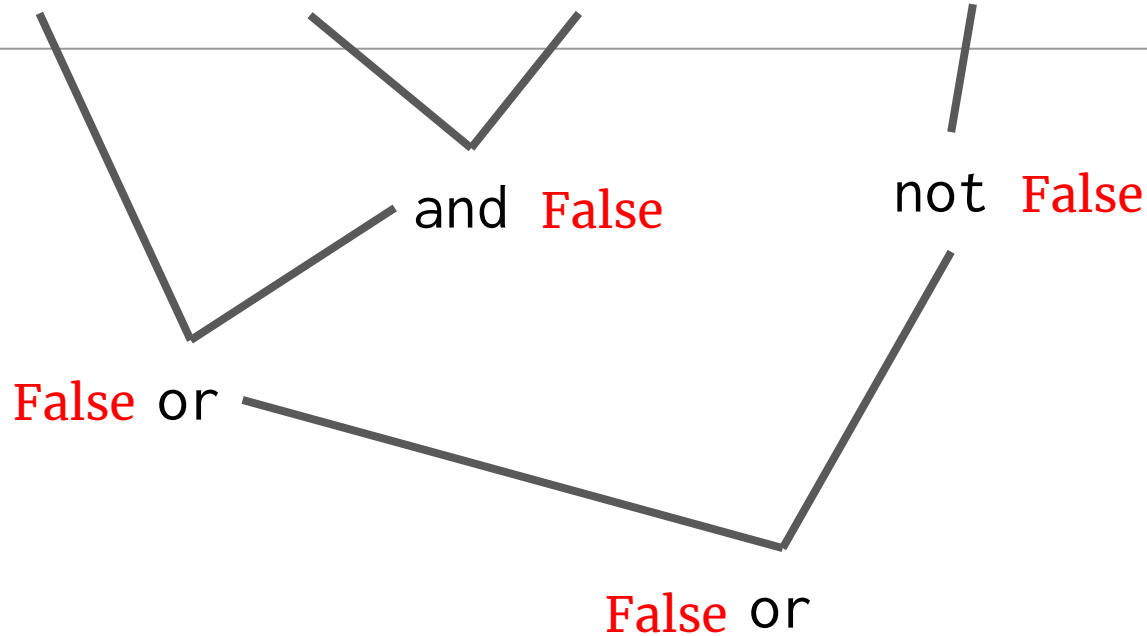
Exemplo de precedência de operadores

```
>>> False or False and True or not True or False
```



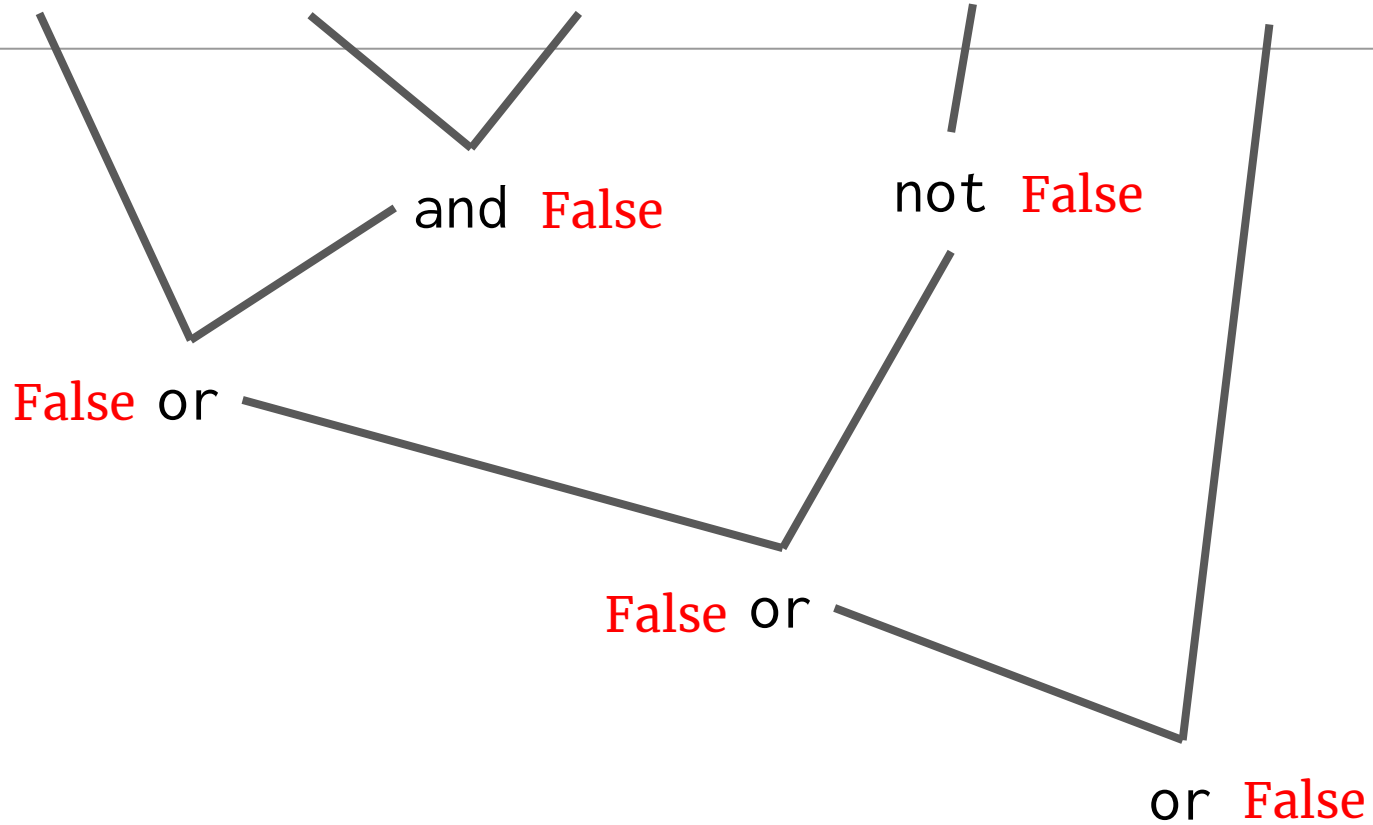
Exemplo de precedência de operadores

```
>>> False or False and True or not True or False
```



Exemplo de precedência de operadores

```
>>> False or False and True or not True or False
```



Comentários

Comentário

este código é ignorado pelo python.

Este também 😊

Comentário

```
# este código é ignorado pelo python.
```

```
# Este também 😊
```

```
# m eh o numero n com as i casas decimais da direita
```

```
# zeradas
```

```
# Ex (i = 3) 54234 -> 54000
```

```
m = n // 10 ** i * 10 ** i
```

Laços

Laço

while « expressão booleana » :



« instruções »

« instruções »

Laço

while « expressão booleana » :



« instruções »

« instruções »

indentação (TAB)




Laço

 **while** « expressão booleana » :

« instruções »

« instruções »

Laço

 **while** « expressão booleana » :

True

« instruções »

« instruções » False

Laço

while « expressão booleana » :



« instruções » True

« instruções » False

Laço

while « expressão booleana » :



« instruções » True

« instruções » False

Laço

while « expressão booleana » :



« instruções » **False**

Laço

while « expressão booleana » :

True

« instruções »



« instruções » False

Laço

while « expressão booleana » :

True

« instruções »



« instruções » False

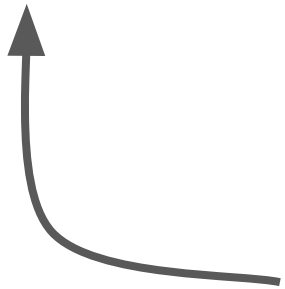
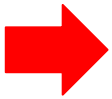
Laço

while « expressão booleana » :

True


« instruções »

« instruções » False



última instrução do bloco

Laço

 **while** « expressão booleana » :

True

« instruções »

« instruções » False

Laço

while « expressão booleana » :



« instruções » **True**

« instruções » **False**

Laço

while « expressão booleana » :

True



« instruções »

« instruções » False

Laço

while « expressão booleana » :

True

« instruções »



« instruções » False

Laço

while « expressão booleana » :

True

« instruções »



« instruções » False

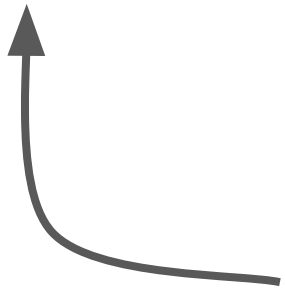
Laço

while « expressão booleana » :

True


« instruções »

« instruções » False



última instrução do bloco

Laço

 **while** « expressão booleana » :

True

« instruções »

« instruções » False

Laço

while « expressão booleana » :

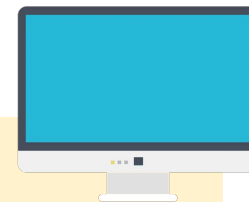
True

« instruções »



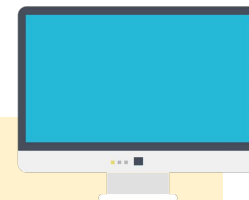
« instruções » False

Exemplo de laço



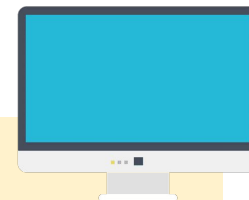
```
n = int(input())
i = n
fat = 1
while i > 1:
    fat = fat * i
    i = i - 1
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
➔ n = int(input())  
i = n  
fat = 1  
while i > 1:  
    fat = fat * i  
    i = i - 1  
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4
```

```
→ n = int(input())  
i = n  
fat = 1  
while i > 1:  
    fat = fat * i  
    i = i - 1  
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 4
```

```
n = int(input())
```

```
i = n
```



```
fat = 1
```

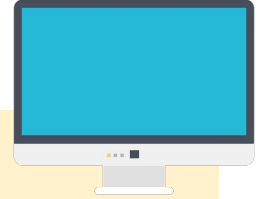
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 4  
fat = 1
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```



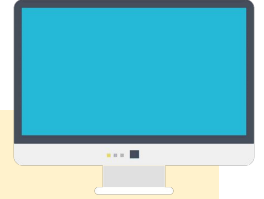
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 4  
fat = 1
```

```
n = int(input())
```

```
i = n
```

```
fat = 1      True
```



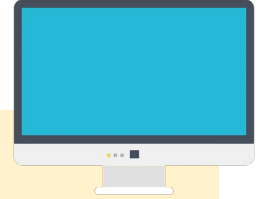
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```


Exemplo de laço



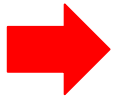
```
n = 4  
i = 4  
fat = 1
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

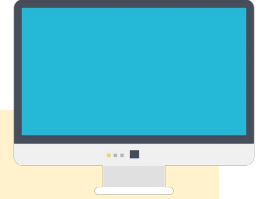


```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 4  
fat = 4
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

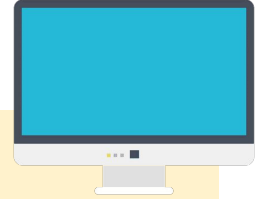
```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```



Exemplo de laço



```
n = 4  
i = 3  
fat = 4
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```



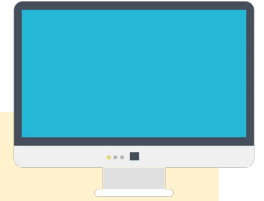
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 3  
fat = 4
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```



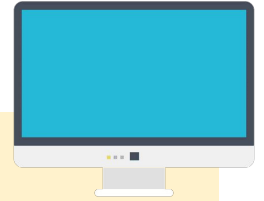
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 3  
fat = 4
```

```
n = int(input())
```

```
i = n
```

```
fat = 1      True
```



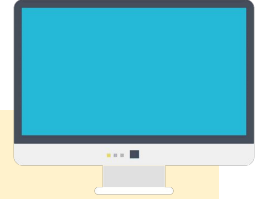
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 3  
fat = 4
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

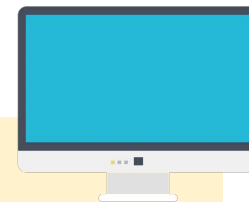


```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 3  
fat = 12
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

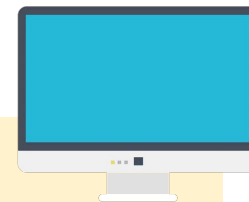
```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```



Exemplo de laço



```
n = 4  
i = 2  
fat = 12
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```



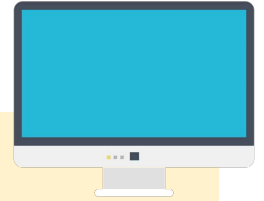
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```


Exemplo de laço



```
n = 4  
i = 2  
fat = 12
```

```
n = int(input())
```

```
i = n
```

```
fat = 1      True
```



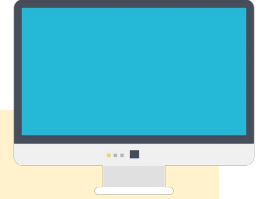
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



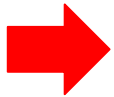
```
n = 4  
i = 2  
fat = 12
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

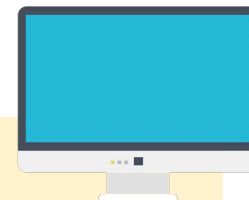


```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 2  
fat = 24
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```



Exemplo de laço



```
n = 4  
i = 1  
fat = 24
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```



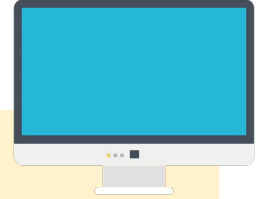
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 1  
fat = 24
```

```
n = int(input())
```

```
i = n
```

```
fat = 1      False
```



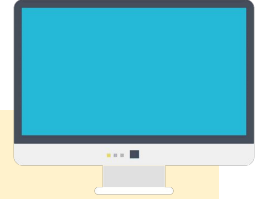
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = 4  
i = 1  
fat = 24
```

```
n = int(input())
```

```
i = n
```

```
fat = 1
```

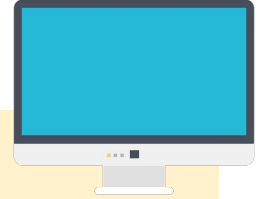
```
while i > 1:
```

```
    fat = fat * i
```

```
    i = i - 1
```

```
➔ print("%d! = %d" % (n, fat))
```

Exemplo de laço



```
n = int(input())
```

```
i = n
```

```
fat = 1
```

```
while i > 1:
```

```
    fat = fat * i
```

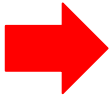
```
    i = i - 1
```

```
print("%d! = %d" % (n, fat))
```

```
n = 4
```

```
i = 1
```

```
fat = 24
```



```
4! = 24
```

Lista 1

Histograma de Feitos

