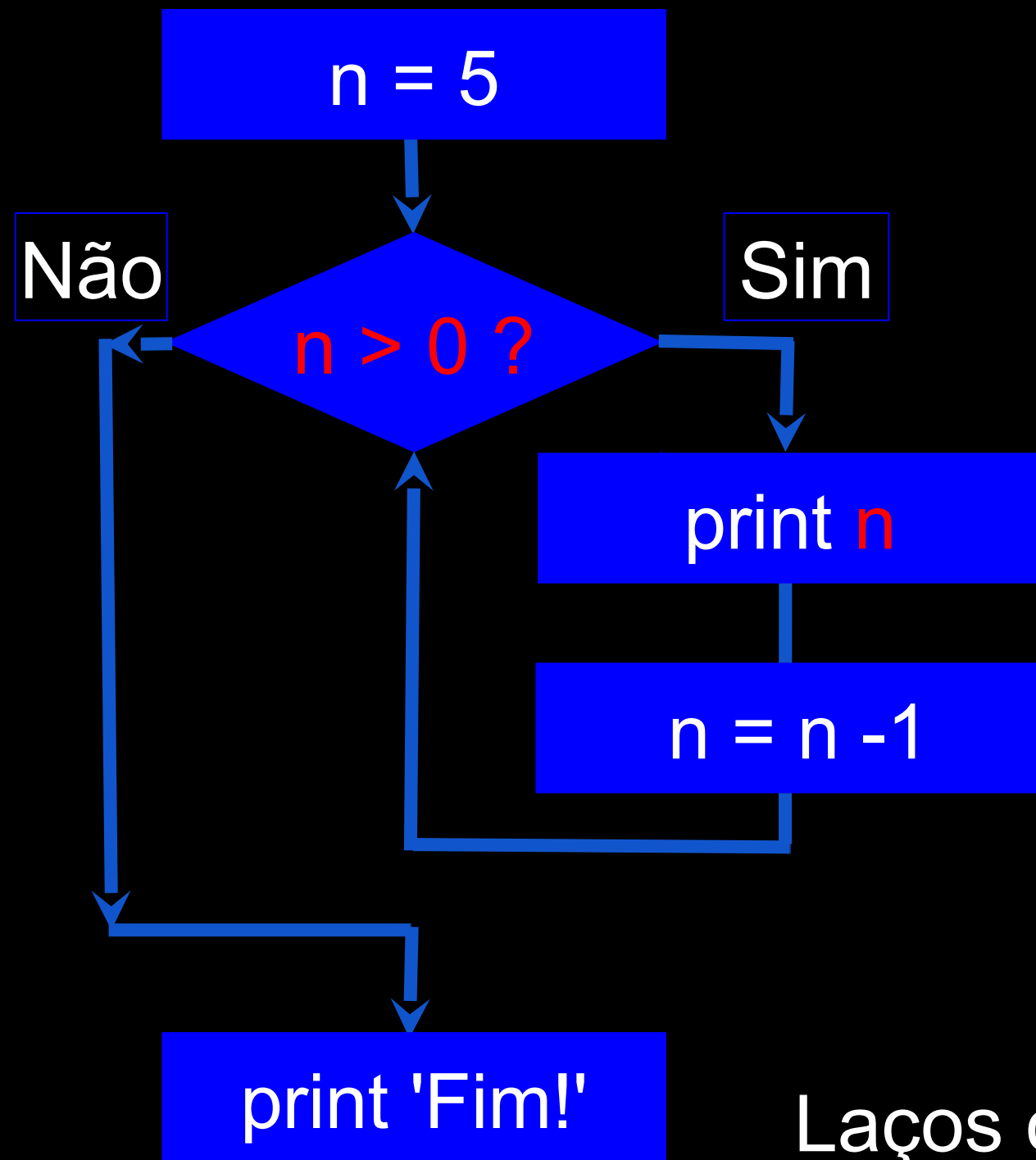


Laços e Iteração (Loops)

Prof. Alberto Costa Neto
Programação em Python

Passos Repetidos



Programa:

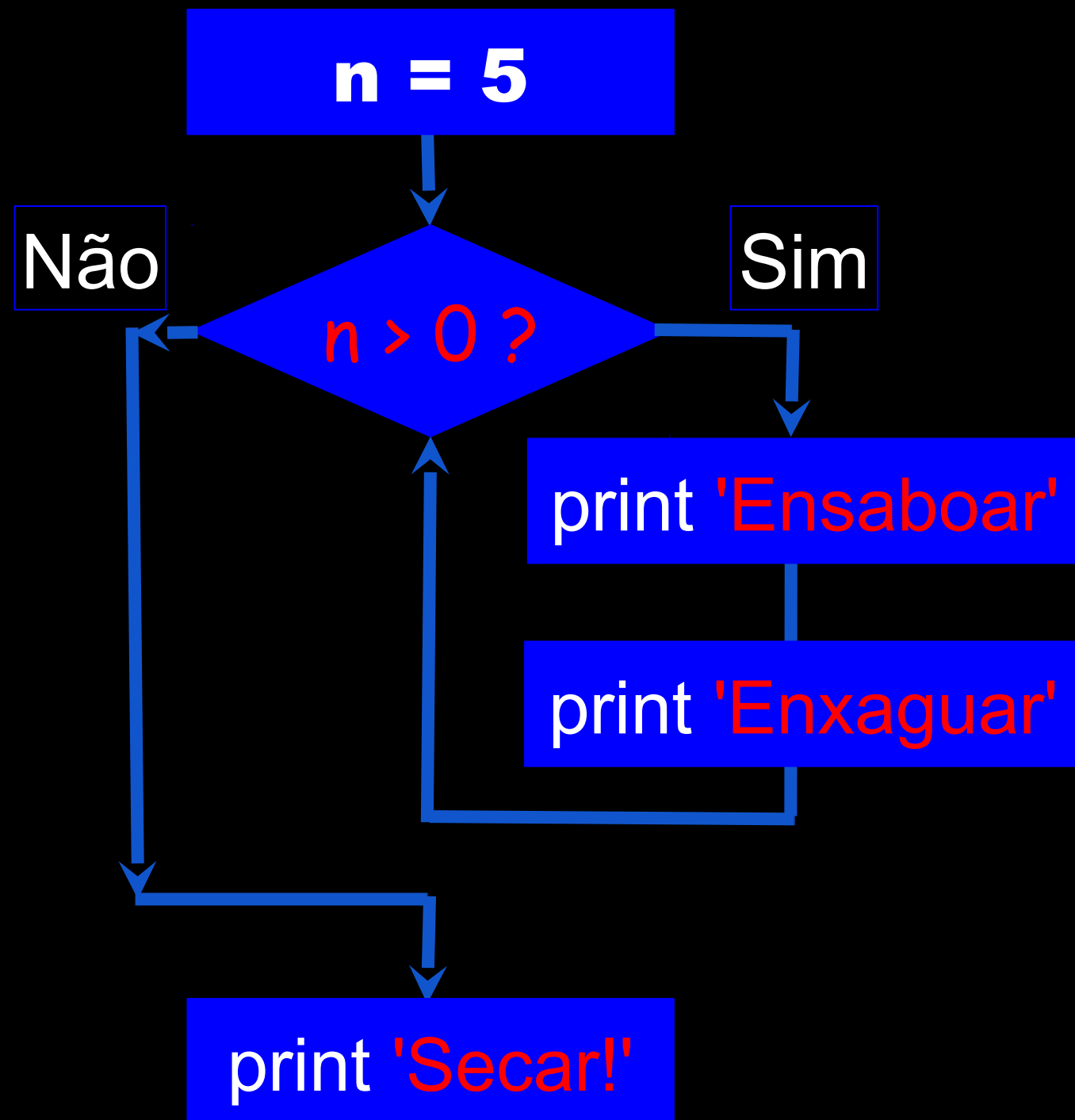
```
n = 5
while n > 0 :
    print n
    n = n - 1
print 'Fim!'
```

Saída:

5
4
3
2
1
Fim!

Laços ou Loops (passos repetidos) têm variáveis de **variáveis de iteração** que mudam a cada passada do laço. Frequentemente estas **variáveis de iteração** recebem uma sequência de números.

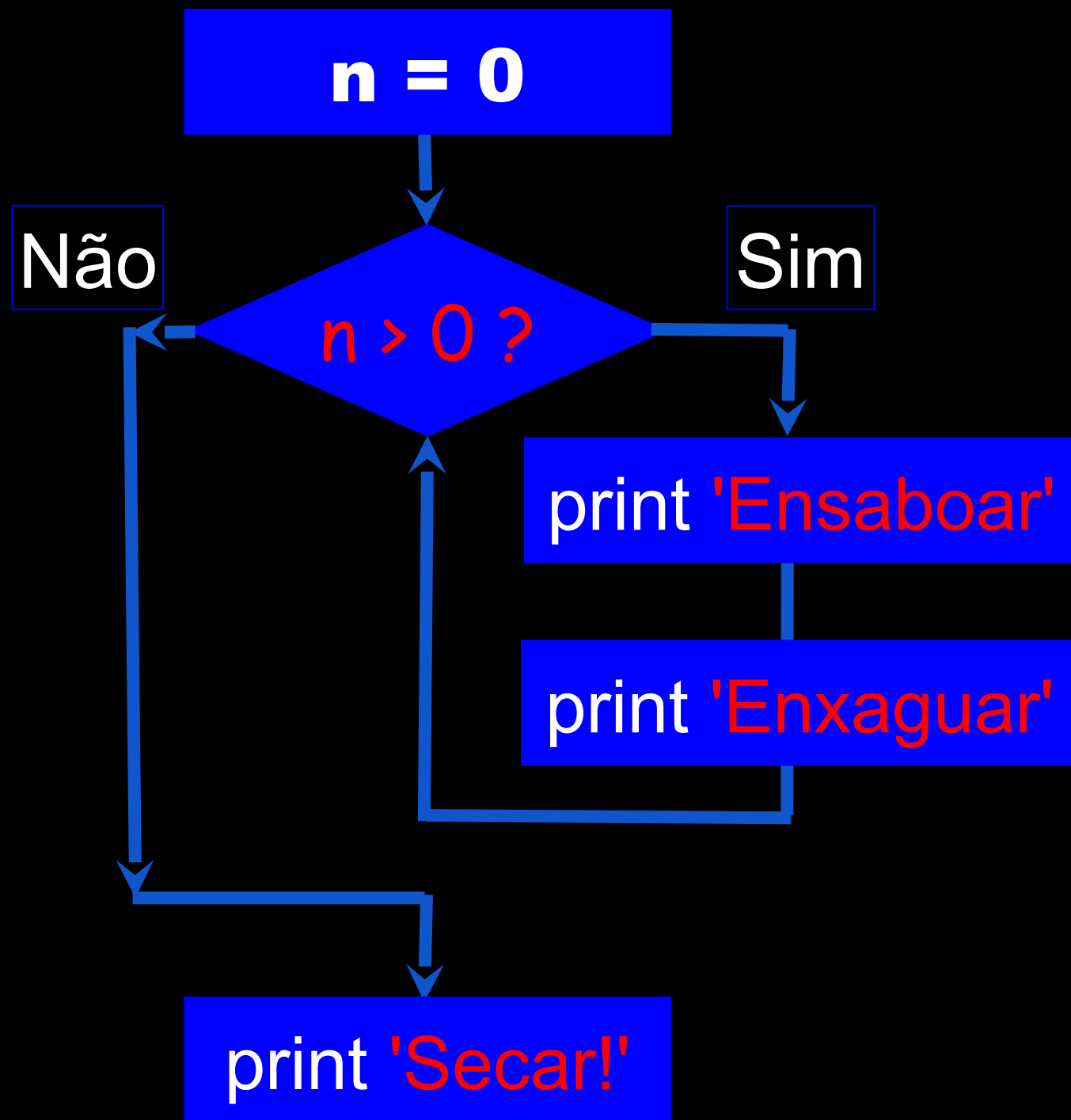
Um Laço Infinito



```
n = 5
while n > 0 :
    print 'Ensaboar'
    print 'Enxaguar'
print 'Secar!'
```

O que há de errado com este laço?

Outro Laço



```
n = 0
while n > 0 :
    print 'Ensaboar'
    print 'Enxaguar'
print 'Secar!'
```

O que este laço faz?

Quebrando um Loop

- O comando **break** encerra o laço corrente e pula para o próximo comando imediatamente após o laço

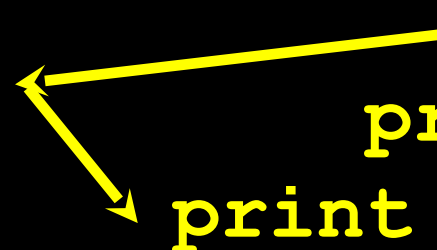
```
while True:
    linha = raw_input('> ')
    if linha == 'fim' :
        break
    print linha
print 'Agora é o fim!'
```

```
> ola
ola
> terminou
terminou
> fim
Agora é o fim!
```

Quebrando um Loop

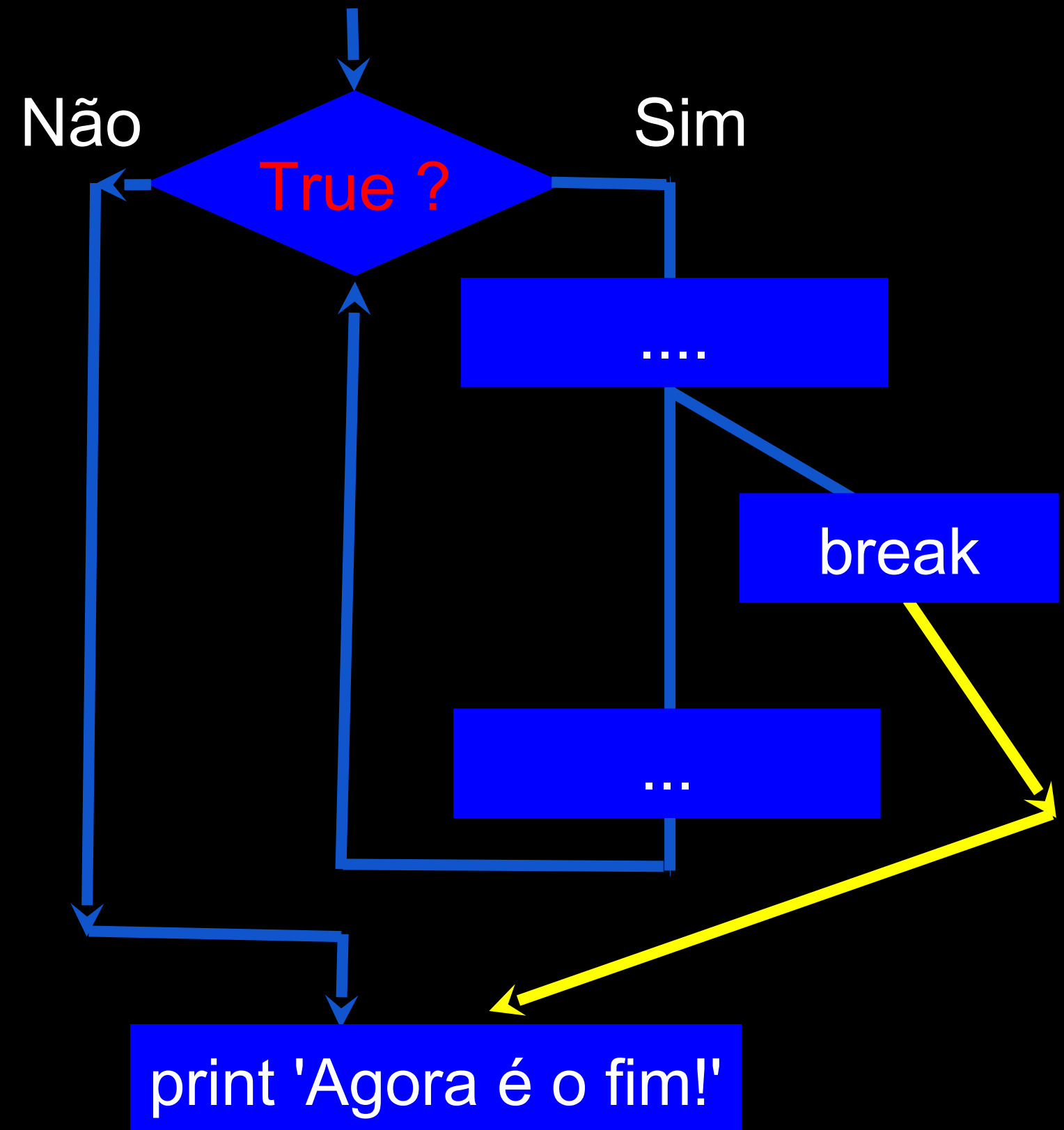
- O comando **break** encerra o laço corrente e pula para o próximo comando imediatamente após o laço

```
while True:
    linha = raw_input('> ')
    if linha == 'fim' :
        break
    print linha
print 'Agora é o fim!'
```



```
> ola
ola
> terminou
terminou
> fim
Agora é o fim!
```

```
while True:
    linha = raw_input('> ')
    if linha == 'fim' :
        break
    print linha
    print 'Agora é o fim!'
```



Encerrar uma iteração com o comando continue

O comando **continue** encerra a iteração corrente e volta ao topo do laço, testa a condição. Caso seja **True**, inicia a próxima iteração

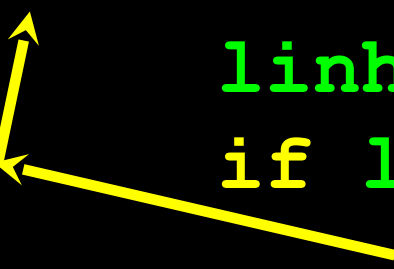
```
while True:
    linha = raw_input('> ')
    if linha[0] == '#':
        continue
    if linha == 'fim':
        break
    print linha
print 'Agora é o fim!'
```

```
> ola
ola
> # nada
> ok!
ok!
> fim
Agora é o fim!
```


Encerrar uma iteração com o comando continue

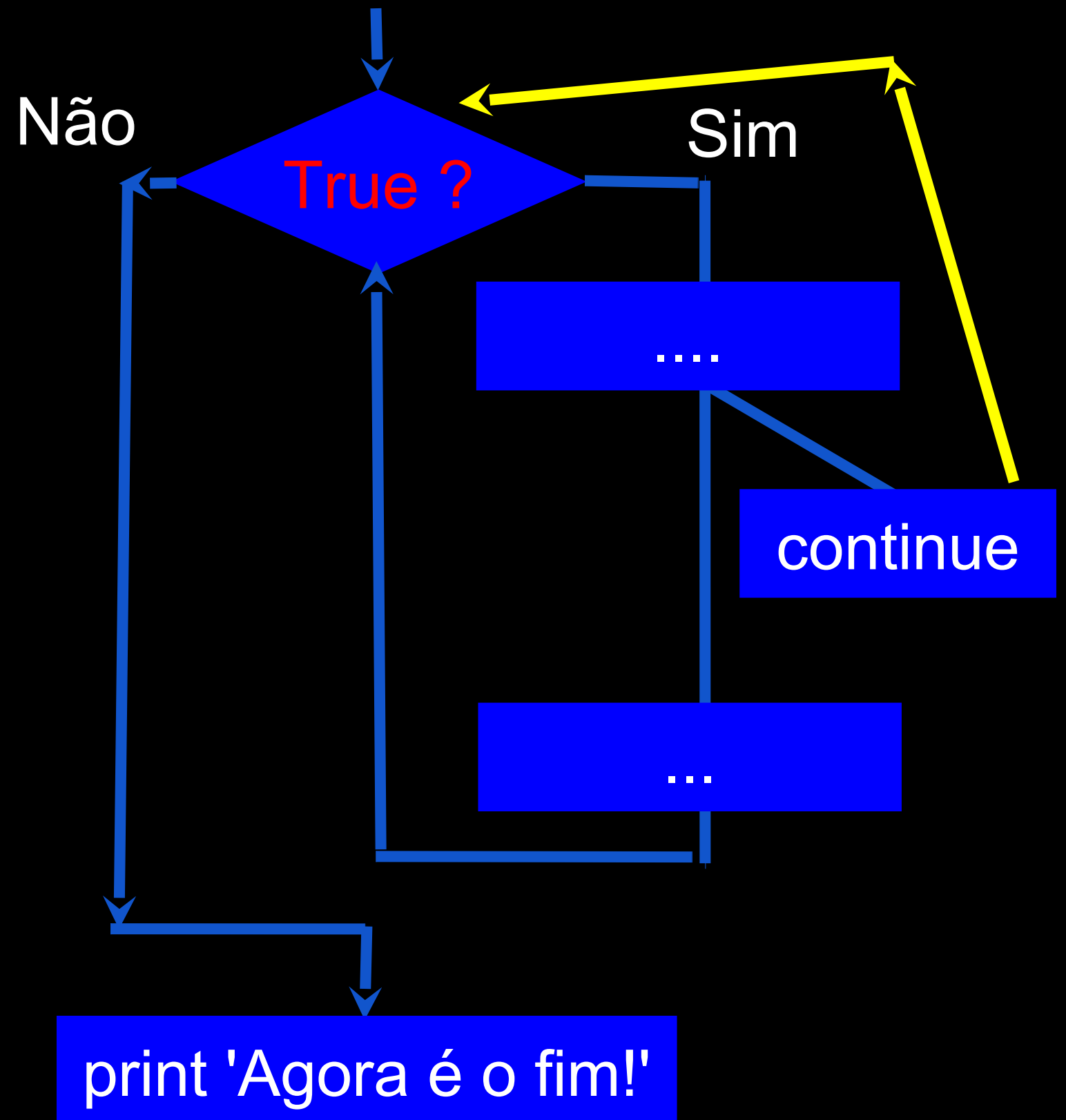
O comando `continue` encerra a **iteração corrente** e volta ao **topo do laço, testa a condição**. Caso seja **True**, inicia a próxima iteração

```
while True:
    linha = raw_input('> ')
    if linha[0] == '#':
        continue
    if linha == 'fim':
        break
    print linha
print 'Agora é o fim!'
```



```
> ola
ola
> # nada
> ok!
ok!
> fim
Agora é o fim!
```

```
while True:
    linha = raw_input('> ')
    if linha[0] == '#' :
        continue
    if linha == 'fim' :
        break
    print line
print 'Agora é o fim!'
```



Laços Indefinidos

- Laços **while** são chamados “**laços indefinidos**” porque eles continuam executando até o teste da condição lógica de entrada no laço dar **False**
- Os laços que vimos até agora são bem fáceis de examinar para identificar se irão terminar ou se serão “laços infinitos”
- Às vezes é difícil ter certeza de que um laço irá terminar

Laços Definidos

- É muito comum se ter uma **lista** de itens de **linhas em um arquivo** – efetivamente um **conjunto finito** de valores
- Podemos escrever um laço que rode uma iteração para cada um dos itens do conjunto usando a construção **for** do Python
- Estes laços são chamados “**laços definidos**” porque executam um número exato de vezes
- Dizemos que “**laços definidos iteram sobre os membros de um conjunto**”

Um Laço Definido Simples

```
for i in [5, 4, 3, 2, 1] :  
    print i  
print 'Fim!'
```

5
4
3
2
1
Fim!

Um Laço Definido com Strings

```
amigos = ['Joseph', 'Glenn', 'Sally']  
for amigo in amigos :  
    print 'Feliz Ano Novo:', amigo  
print 'Fim!'
```

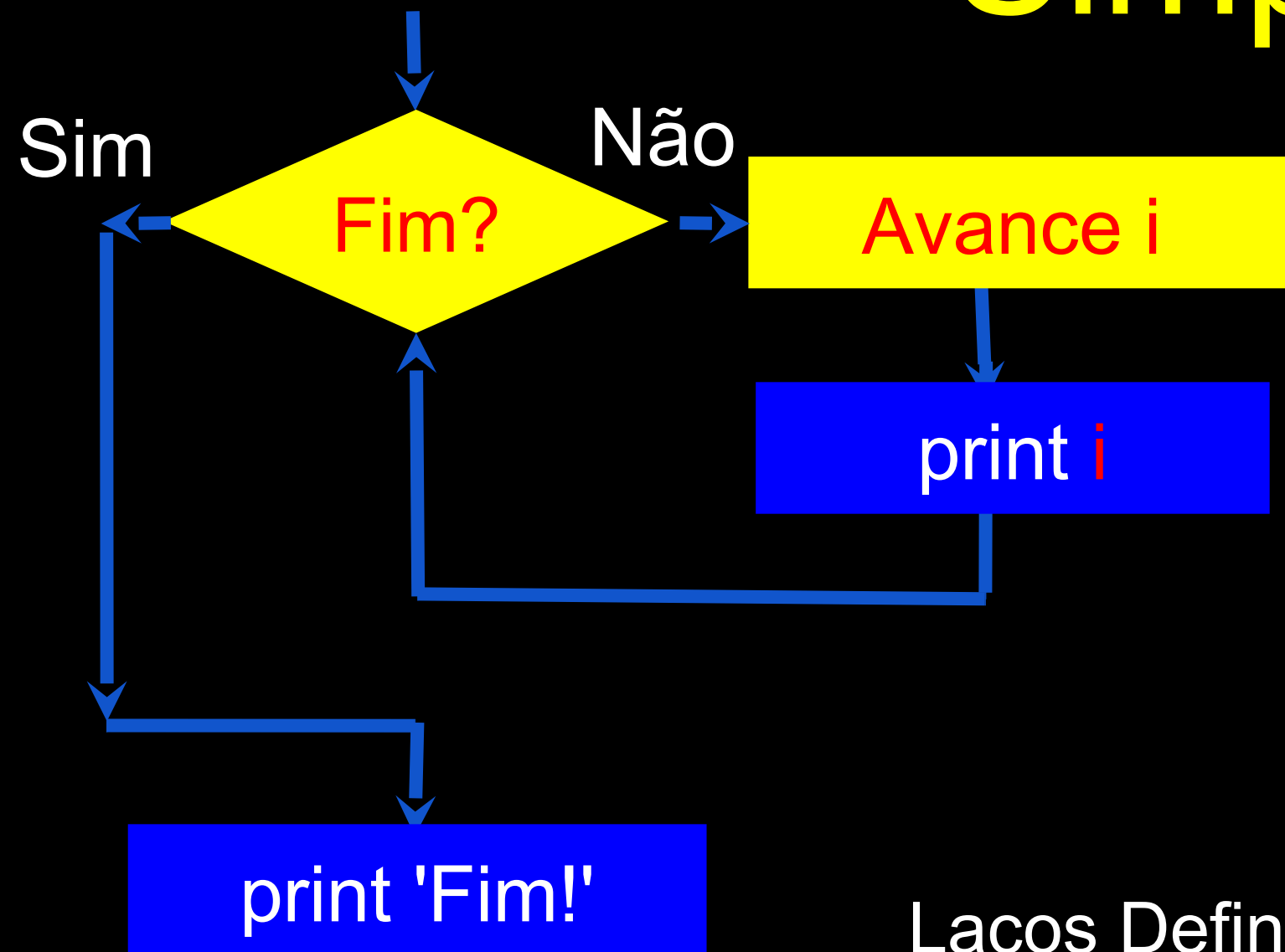
Feliz Ano Novo: Joseph

Feliz Ano Novo: Glenn

Feliz Ano Novo: Sally

Fim!

Um Laço Definido Simples



```
for i in [5, 4, 3, 2, 1] :  
    print i  
print 'Fim!'
```

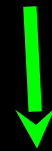
5
4
3
2
1
Fim!

Laços Definidos (laços **for**) têm **variáveis de iteração** explícitas que mudam a cada iteração do laço. Estas **variáveis de iteração** movem-se através de uma seqüência ou conjunto de valores.

Entendendo o **in**...

- A **variável de iteração** “itera” pela **seqüência** (conjunto ordenado)
- O **bloco (corpo)** do laço é executado uma vez para cada valor na **seqüência**
- A **variável de iteração** move-se por todos os valores na **seqüência**

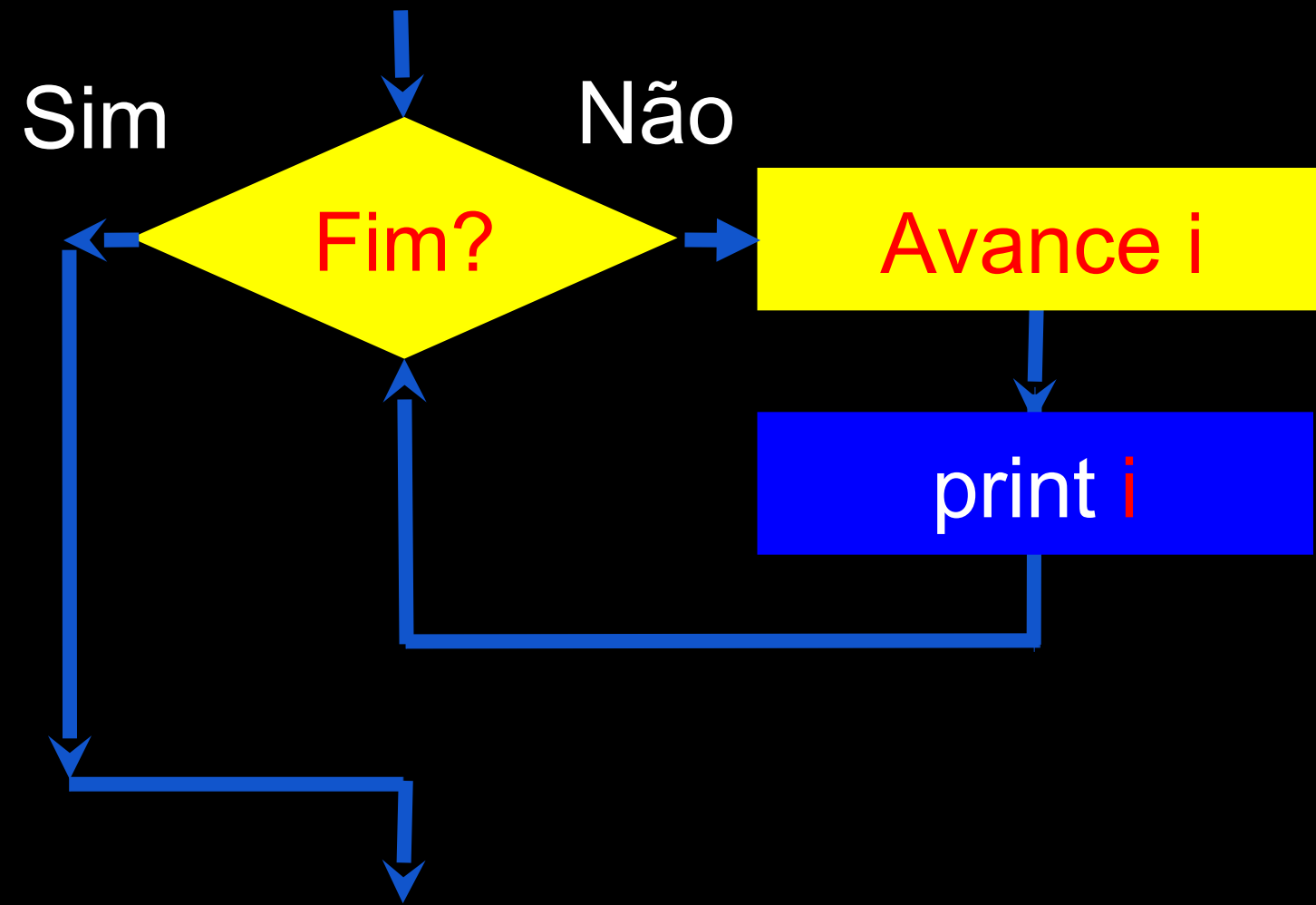
**Variável de
Iteração**



**Seqüência com 5
elementos**

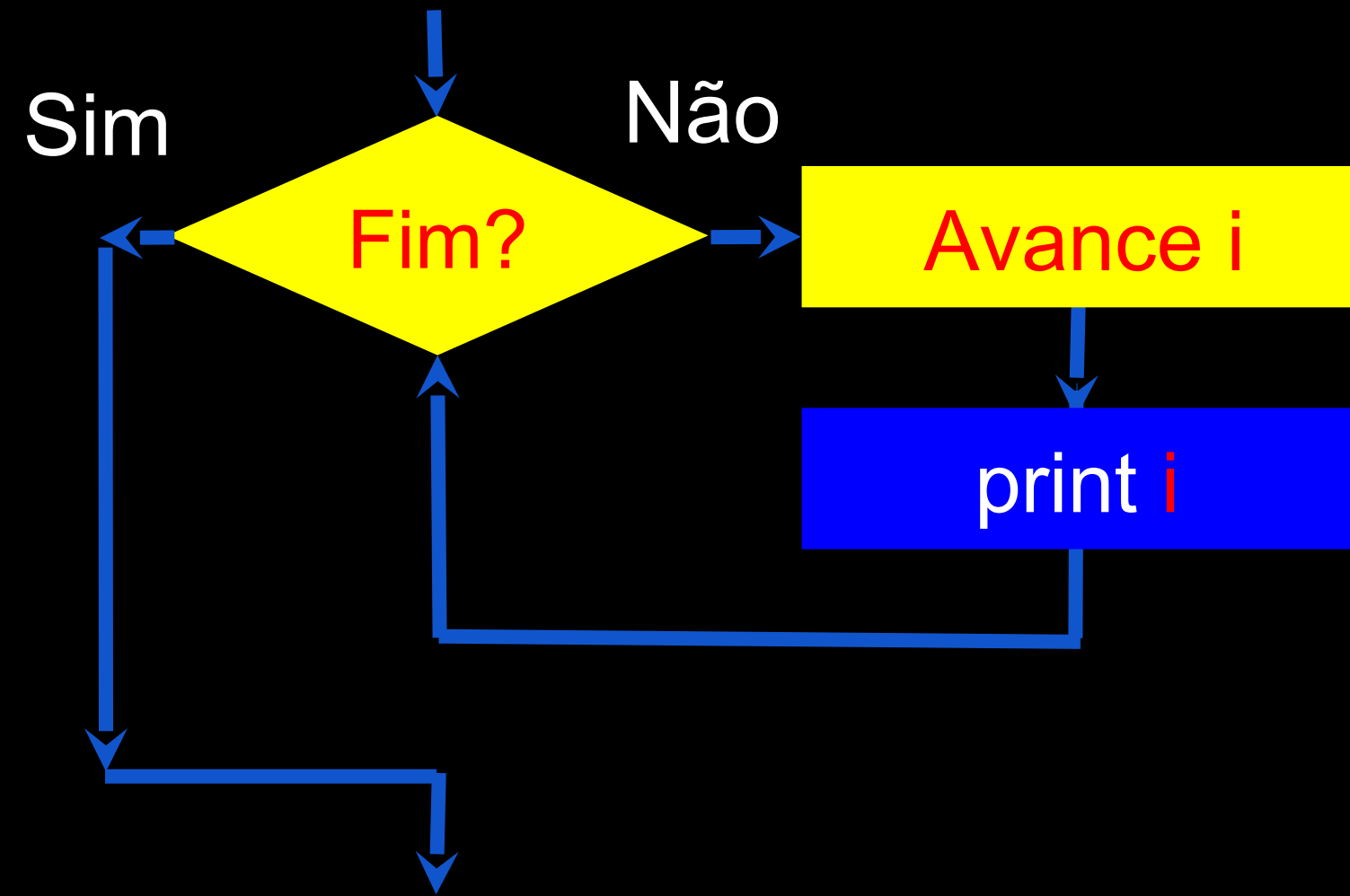


```
for i in [5, 4, 3, 2, 1] :  
    print i
```

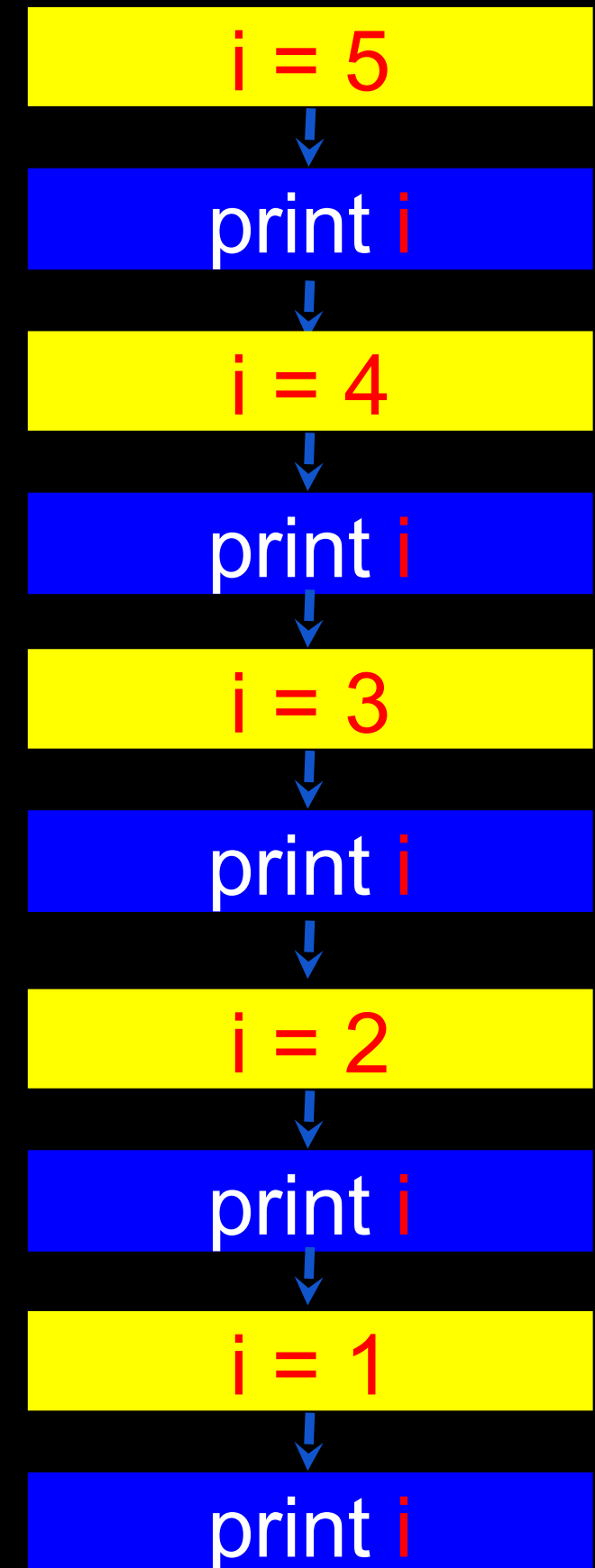



```
for i in [5, 4, 3, 2, 1] :  
    print i
```

- A **variável de iteração** “itera” pela **seqüência** (conjunto ordenado)
- O **bloco (corpo)** do laço é executado uma vez para cada valor na **seqüência**
- A **variável de iteração** move-se por todos os valores na **seqüência**



```
for i in [5, 4, 3, 2, 1] :  
    print i
```





Acknowledgements / Contributions

Agradecimentos / Contribuições



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and open.umich.edu and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information



These slides were translated and adapted by Alberto Costa Neto (albertocn.sytes.net) of the Federal University of Sergipe