

# ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

## AULA 3: ESTRUTURA DE SELEÇÃO SIMPLES E COMPOSTA

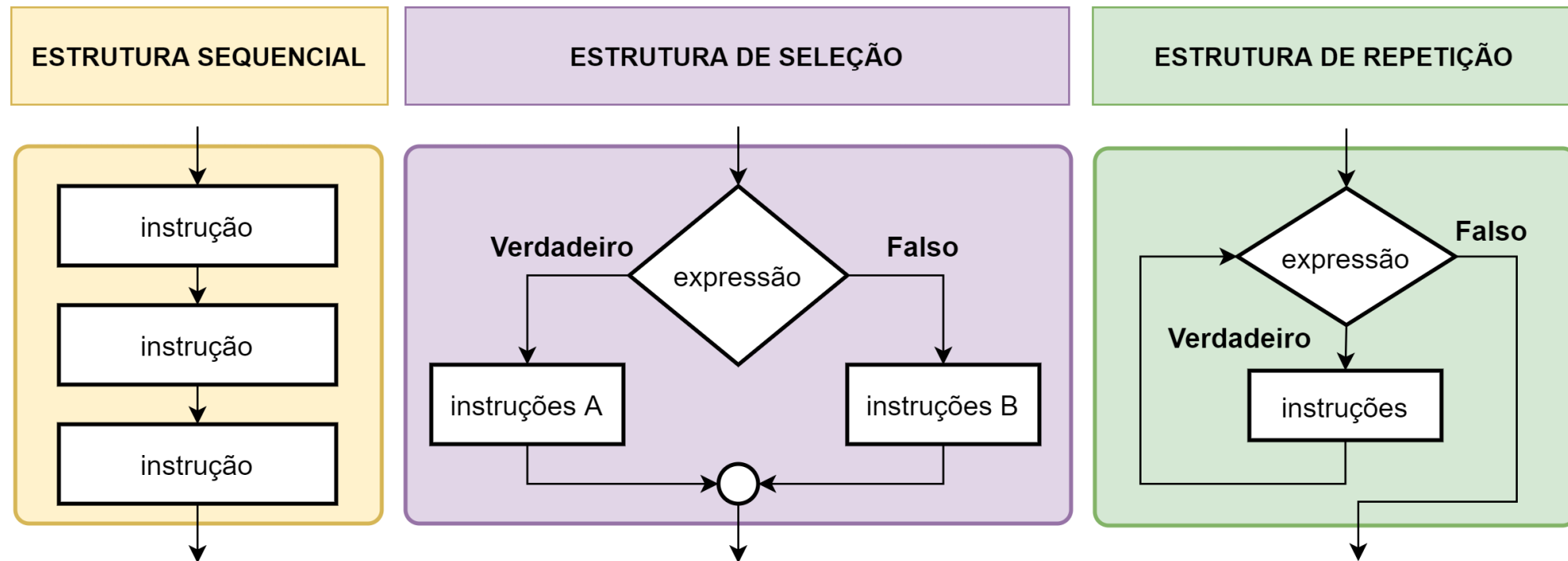
# DO QUE VAMOS FALAR

- **1.** Estruturas de controle de fluxo
- **2.** Estrutura sequencial
- **3.** Estrutura de seleção
  - **3.1** Indentação
  - **3.2** Estrutura de seleção simples
  - **3.3** Estrutura de seleção composta
  - **3.4** Exercícios de seleção simples e composta



# ESTRUTURAS DE CONTROLE DE FLUXO

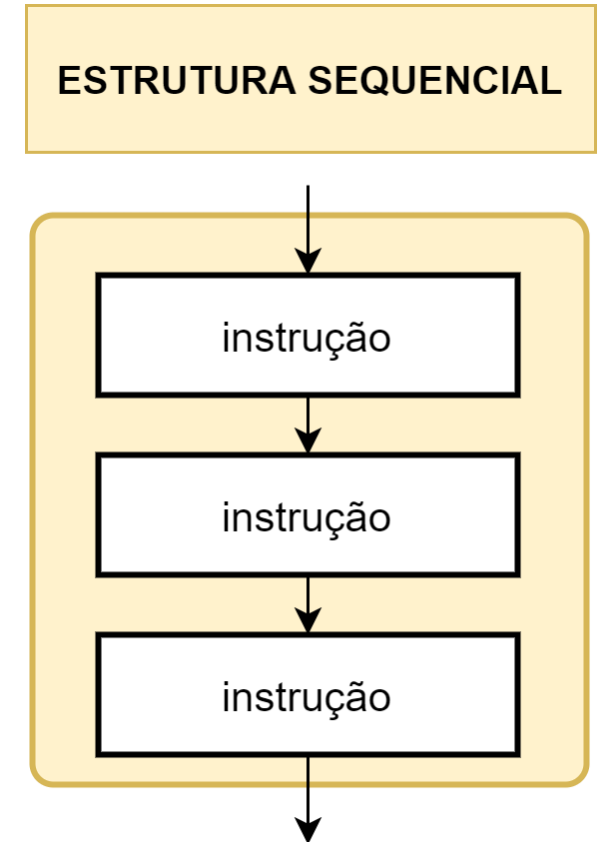
A ordem em que as instruções são executadas em um programa chama-se fluxo de execução. Existem três estruturas fundamentais para o controle do fluxo de execução:



# ESTRUTURA SEQUENCIAL

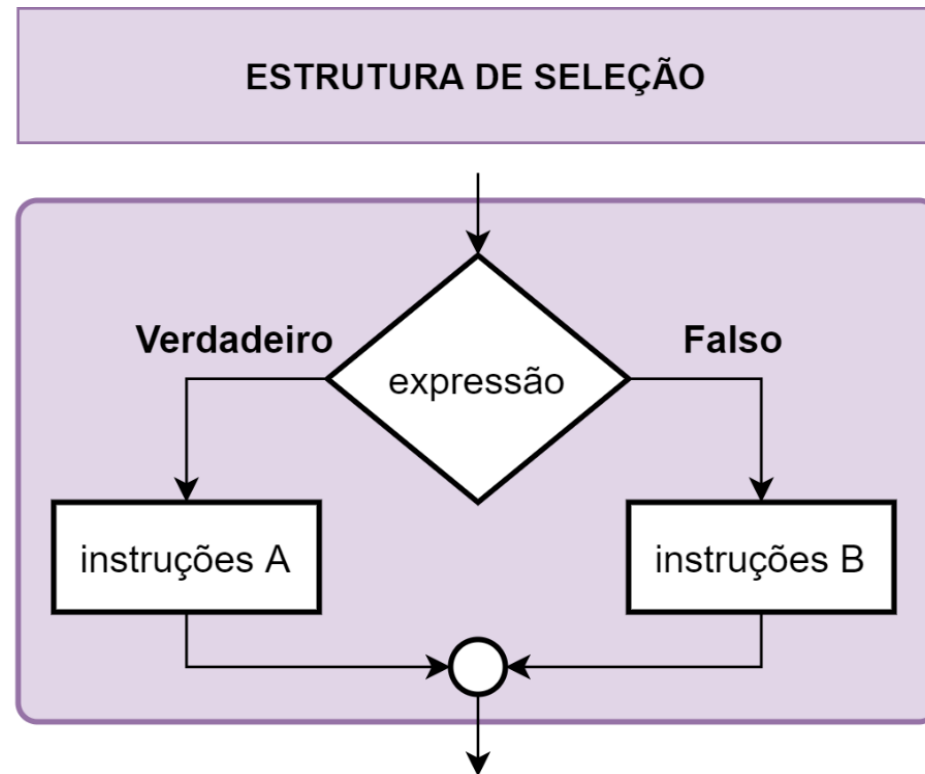
Na estrutura sequencial de controle de fluxo, a ordem de execução das instruções está implícita, pois todas as instruções são executadas e na sequência em que foram escritas.

```
prod = float(input('Preço produto: '))  
qtd  = int(input('Quantidade: '))  
total = prod * qtd  
print('Total a pagar: R$ %.2f' % total)
```



# ESTRUTURA DE SELEÇÃO

Também chamada de **estrutura condicional**, permite selecionar uma sequência de instruções conforme a expressão (expressão condicional).



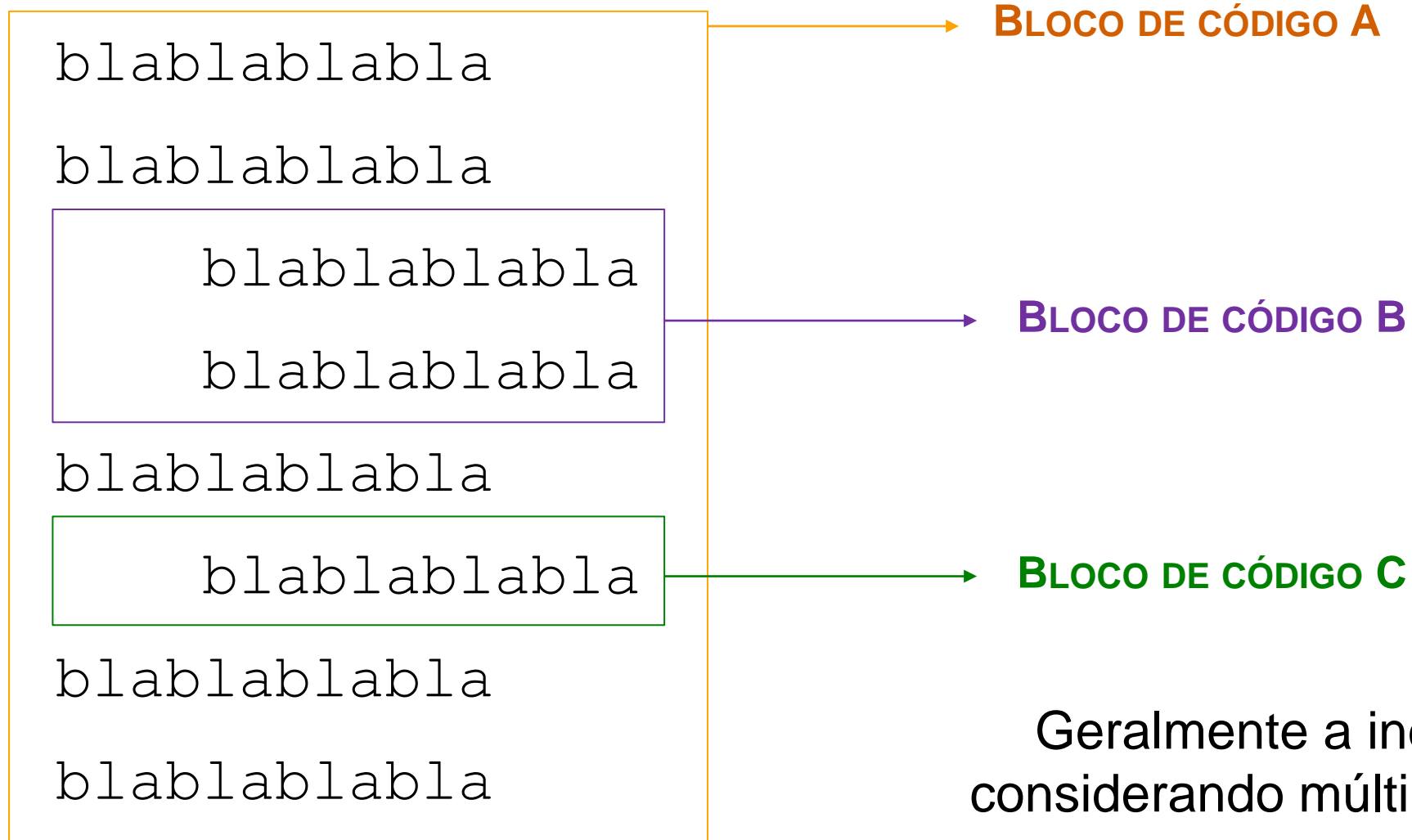
# INDENTAÇÃO

A indentação se refere aos **espaços em branco** à esquerda de uma linha de código, de forma a alinhar as instruções em uma determinada coluna.

Em diversas linguagens de programação o uso de indentação serve apenas para **melhorar a legibilidade** do código e é considerada uma boa prática de programação. Em Python, a indentação é ainda mais importante, pois influencia a forma como a **linguagem interpretará** o que foi escrito.

Python utiliza a indentação para **delimitar os blocos de códigos**, ou seja, linhas de códigos sequenciais que estiverem com a mesma quantidade de espaços (consequentemente estarão na mesma coluna) pertencem ao mesmo bloco de código.

# INDENTAÇÃO

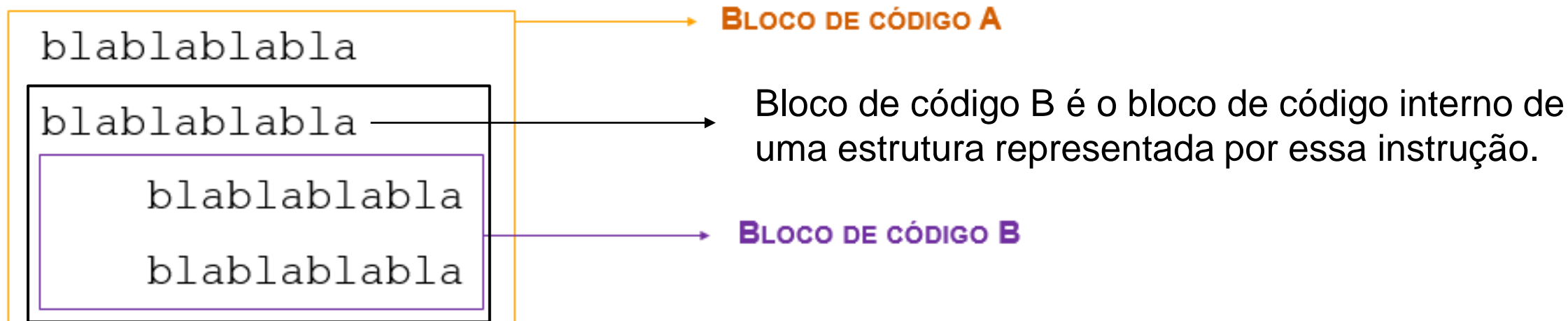


Geralmente a indentação é feita considerando múltiplos de 4 espaços.

# INDENTAÇÃO

Entender a importância da indentação, tanto para melhorar a legibilidade quanto para escrever corretamente em Python, é crucial.

A indentação em Python também serve para indicar que o bloco de código mais indentado (mais espaços à esquerda) **está interno** à estrutura imediatamente anterior que está menos indentada (menos espaços à esquerda).

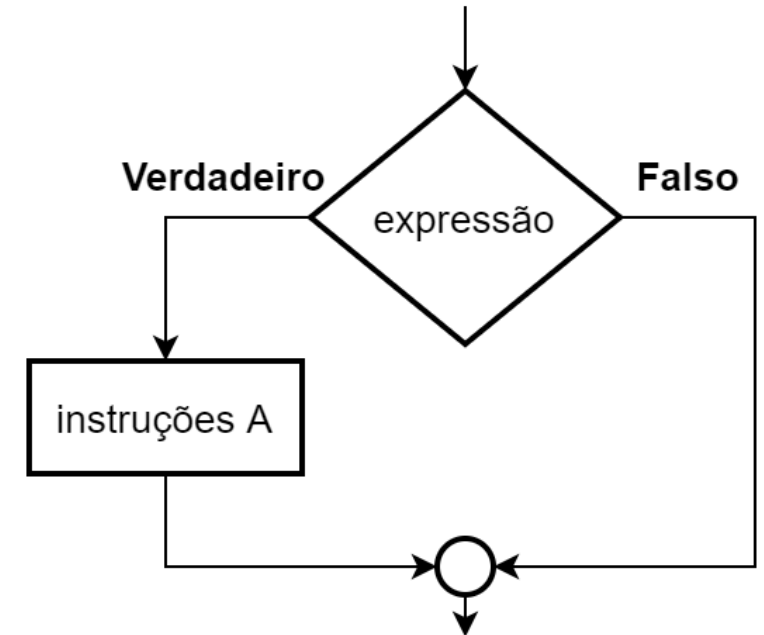




# ESTRUTURA DE SELEÇÃO SIMPLES

Se a avaliação da expressão resultar em **True**, as instruções A serão executadas, caso resulte **False** não serão executadas.

```
prod = float(input('Preço produto: '))
qtd  = int(input('Quantidade: '))
total = prod * qtd
if total >= 150.00:
    total = 0.85 * total
print('Total a pagar: R$ %.2f' % total)
```

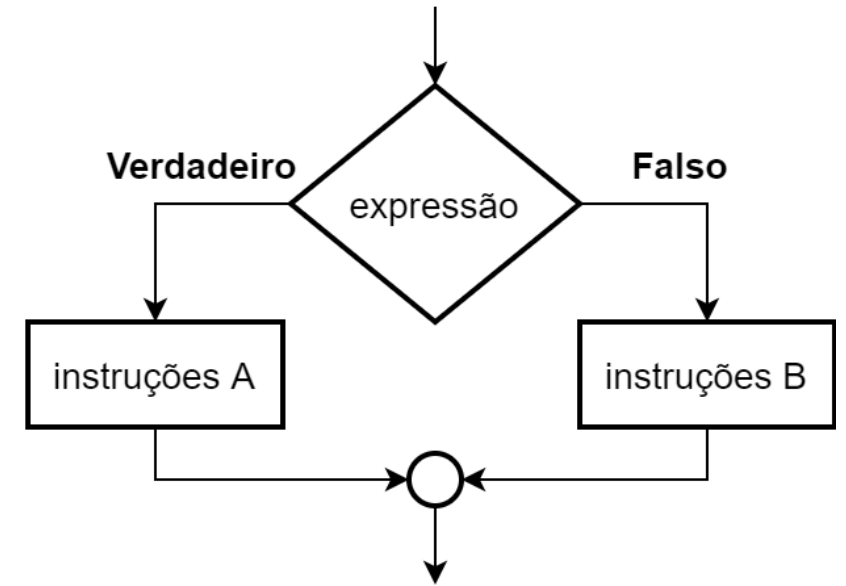


```
...
if expressão:
    instruções A
...
```

# ESTRUTURA DE SELEÇÃO COMPOSTA

Se a avaliação da expressão resultar em **True**, as instruções A serão executadas, caso resulte **False** as instruções B serão executadas.

```
n1 = int(input('número 1: '))
n2 = int(input('número 2: '))
if n1 < n2:
    print('o menor número é:', n1)
    print('o maior número é:', n2)
else:
    print('o menor número é:', n2)
    print('o maior número é:', n1)
print('A soma dos números é:', n1+n2)
```



```
...
if expressão:
    instruções A
else:
    instruções B
...
```