

PROGRAMAÇÃO EM PYTHON



python

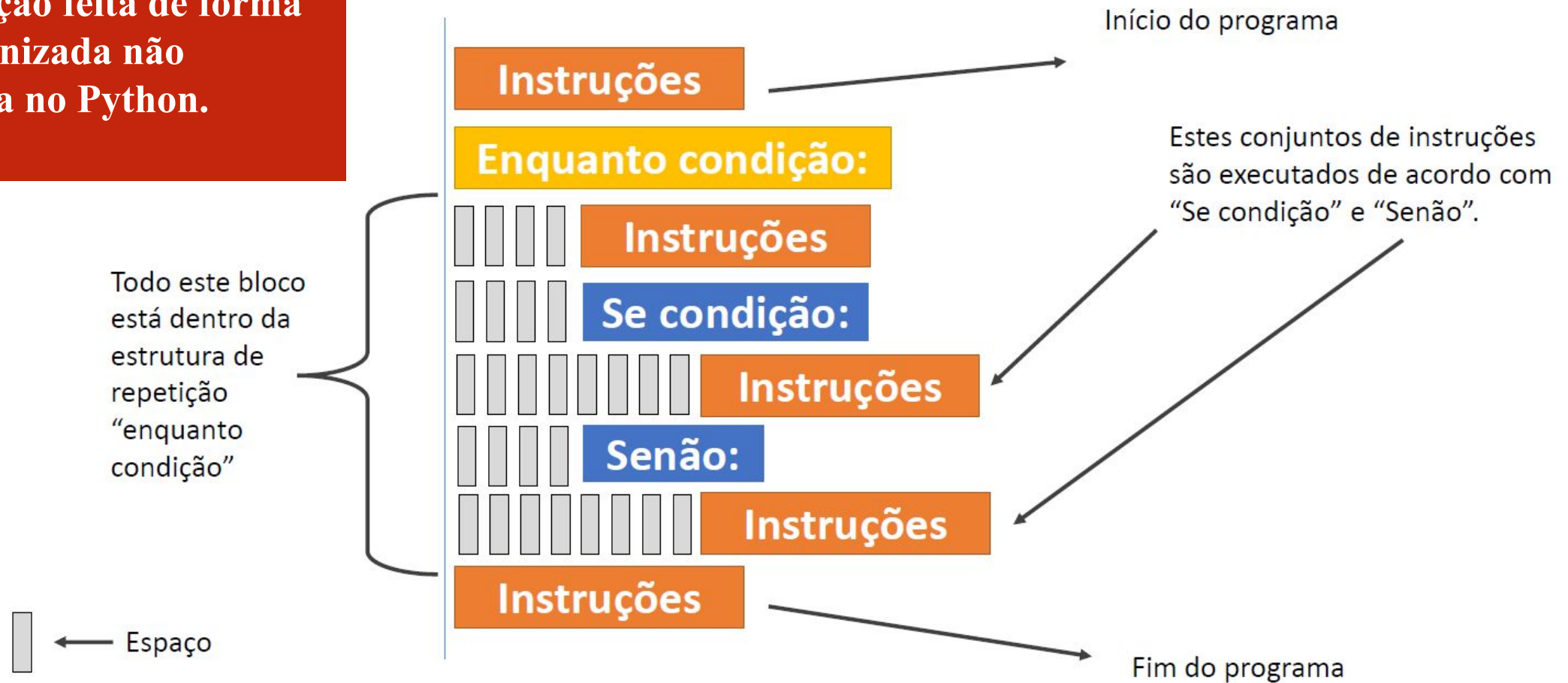


BLOCOS DE CÓDIGO

- Blocos de código são delimitados pelo uso de indentação
- A indentação deve ser constante no bloco de código
- É uma boa prática não misturar tabulação com espaços
- Usar quatro espaços para indentação é uma convenção amplamente aceita, além de ser uma recomendação oficial (<http://python.org/dev/peps/pep-0008>)

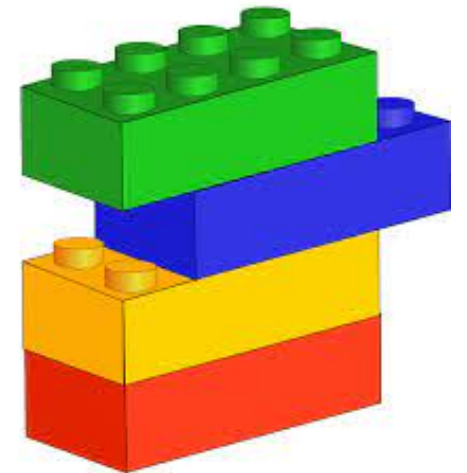
BLOCOS DE CÓDIGO

Um código com a indentação feita de forma desorganizada não funciona no Python.



BLOCO DE CÓDIGO

- O caractere de dois pontos determina a linha anterior a um bloco
Representa uma estrutura de controle da linguagem ou a declaração de uma nova estrutura (uma função, por exemplo)
- Se você não indentar corretamente seu código o mesmo não será executado



ESTRUTURA CONDICIONAL SIMPLES

Uma estrutura de decisão examina uma ou mais condições e decide quais instruções serão executadas dependendo se a condição foi ou não foi.

O comando **if** é uma estrutura de decisão muito utilizada.

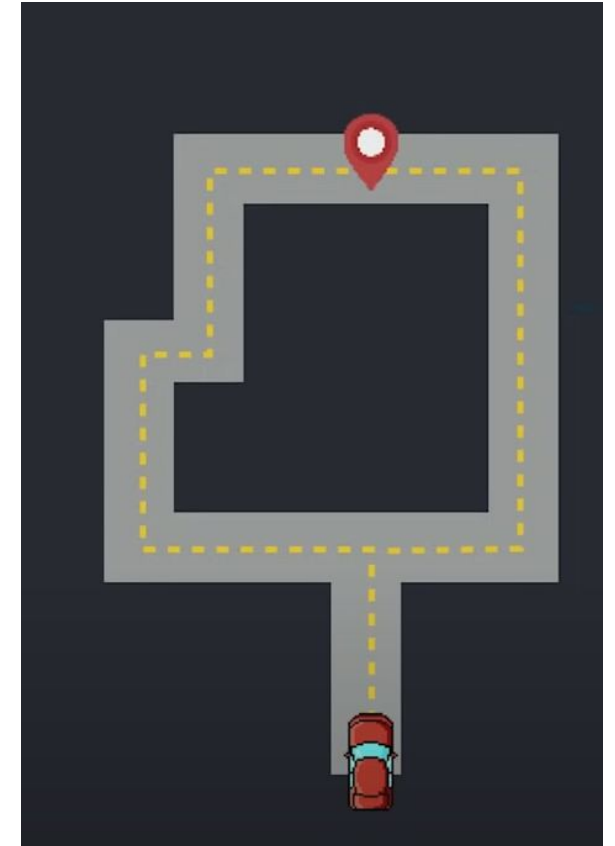
```
media = float(input("Informe a média: "))  
if media > 5:  
    print("Aprovado")
```

ditional

```
C:\Users\Evaldo\AppData\Local\Programs\Pythor
```

```
Informe a média: 6.5
```

```
Aprovado
```



ESTRUTURA CONDICIONAL ELSE

- Podemos pensar no comando **else** como sendo um complemento do comando **if**. O comando **if** completo tem a seguinte forma geral:

```
media = float(input("Informe a média: "))  
if media > 5:  
    print("Aprovado")  
else:  
    print("Reprovado")
```

- É importante nunca esquecer que, quando usamos a estrutura **if-else**, estamos garantindo que uma das duas declarações será executada. Nunca serão executadas as duas ou nenhuma delas.

OPERADORES DE COMPARAÇÃO

Igual a (==) - Retorna True se os operandos forem iguais.

$$X == Y$$

Diferente de (!=) - Retorna True se os operandos não forem iguais.

$$X != Y$$

Maior que (>) - Retorna True se o operando à esquerda for maior que o operando à direita.

$$X > Y$$

Menor que (<) - Retorna True se o operando à esquerda for menor que o operando à direita

$$X < Y$$

OPERADORES DE COMPARAÇÃO

Maior ou igual a (\geq) - Retorna True se o operando à esquerda for maior ou igual ao operando à direita.

$X \geq Y$

Menor ou igual a (\leq) - Retorna True se o operando à esquerda for menor ou igual ao operando à direita.

$X \leq Y$

OPERADORES LÓGICOS OR E AND

Os operadores lógicos **or**, **and** e **not** são fundamentais para construir expressões lógicas em Python. Agora iremos ver uma explicação sobre cada um deles:

or (ou) - O operador **or** retorna **True** se pelo menos uma das condições for verdadeira.

```
4  if (condição1 or condição2):  
5      print("Bloco de código a ser executado se pelo menos uma das condições for verdadeira")  
6
```

and (e) - O operador **and** retorna **True** apenas se todas as condições forem verdadeiras.

```
4  if (condição1 and condição2):  
5      print("Bloco de código a ser executado se ambas as condições forem verdadeiras")  
6
```

OPERADOR LÓGICO NOT

not (não) - O operador not é usado para inverter o valor de uma condição. Se a condição for verdadeira, not a torna falsa, e vice-versa.

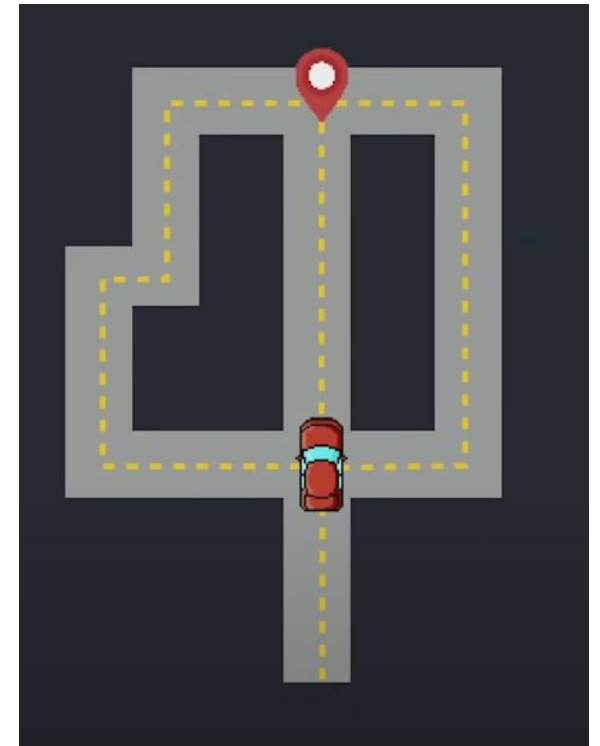
```
4  if not condição:  
5      print("Bloco de código a ser executado se a condição for falsa")  
6
```

VAMOS PRATICAR

ESTRUTURA CONDICIONAL O **ELIF**

- A estrutura **elif** é apenas uma extensão da estrutura **if-else**. Sua forma geral pode ser escrita como sendo:

```
media = float(input("Informe a média: "))  
if media > 7:  
    print("Aprovado")  
elif media < 5:  
    print("Reprovado")  
else:  
    print("Recuperação")
```



VAMOS PRATICAR