CEUB - Ciência da Computação
Lógica de Programação
Professor: Antônio **Barbosa** Júnior

- Operadores relacionais

Em problemas de programação é extremamente comum precisar fazer algum tipo de comparação. Para permitir que elas sejam feitas existem os **operadores relacionais**, que são comparam e decidem sobre a relação entre seus operandos.

- O Python tem os seguintes operadores relacionais:

==	igual	verdade se os dois lados são iguais, falso caso contrário	5 == 5 é verdade 5 == 3 é falso
!=	diferente	verdade se os dois lados são diferentes, falso caso contrário	5 != 3 é verdade 5 != 5 é falso
>	maior	verdade se o lado esquerdo é maior que o direito, falso caso contrário	5 > 3 é verdade 3 > 5 é falso
<	menor	verdade se o lado esquerdo é menor que o direito, falso caso contrário	3 < 5 é verdade 5 < 3 é falso
>=	maior ou igual	verdade se o lado esquerdo é maior ou igual ao lado direito, falso caso contrário	5 >= 3 é verdade 5 >= 5 é verdade 3 >= 5 é falso
<=	menor ou igual	verdade se o lado esquerdo é menor ou igual ao lado direito, falso caso contrário	3 <= 5 é verdade 3 <= 3 é verdade 5 <= 3 é falso

Obs.: não use espaço entre os dois operadores.

O uso de operadores relacionais não é restrito apenas a números, uma infinidade de outros tipos de dados em Python aceita o uso de operadores relacionais também. Nesses casos, o significado do operador relacional pode mudar, portanto, deve-se verificar o comportamento desses operadores antes de usá-los em outros tipos de dados.

Os **operadores lógicos** servem para avaliar o resultado de expressões lógicas de acordo com a lógica booleana:

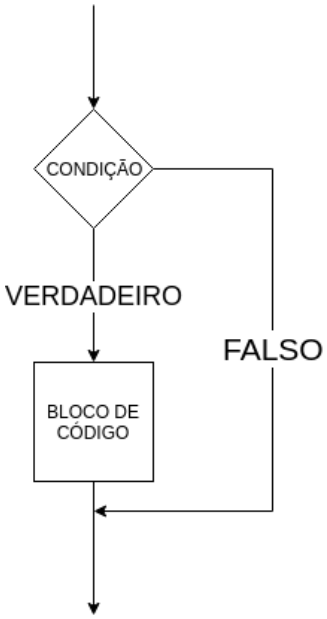
and	E lógico	verdade se ambos os lados são verdadeiros, falso caso contrário	falso and falso é falso falso and verdadeiro é falso verdadeiro and falso é falso verdadeiro and verdadeiro é verdadeiro

or	OU lógico	verdade se pelo menos um ou ambos os lados são verdadeiros, falso caso contrário	falso or falso é falso falso or verdadeiro é verdadeiro verdadeiro or falso é verdadeiro verdadeiro or verdadeiro é verdadeiro
not	NÃO lógico	Inverso do lado direito	not falso é verdadeiro not verdadeiro é falso

- Observação: o resultado dessas operações (relacionais ou lógicas) é um valor booleano que pode assumir True (verdadeiro) ou False (falso).

- if

if é uma estrutura de condição que permite executar um determinado bloco de código caso uma condição estabelecida seja verdadeira e não o executar caso a condição seja falsa, como mostra o **fluxograma**:



- Exemplo 1:

. . .
numero = 5
if numero > 3:
    print(“O número é maior que 3”) # Indentação obrigatória.
. . .

o código **print(“O número é maior que 3”)** será executado **somente se numero > 3 for verdade**. Como **5 > 3 é verdade**, **“O número é maior que 3”** será exibido na tela.

- Exemplo 2:

. . .
numero = 1
if numero > 3:
    print(“O número é maior que 3”)
. . .

o trecho **print(“O número é maior que 3”) NÃO será executado**, porque **1 > 3 é falso**.

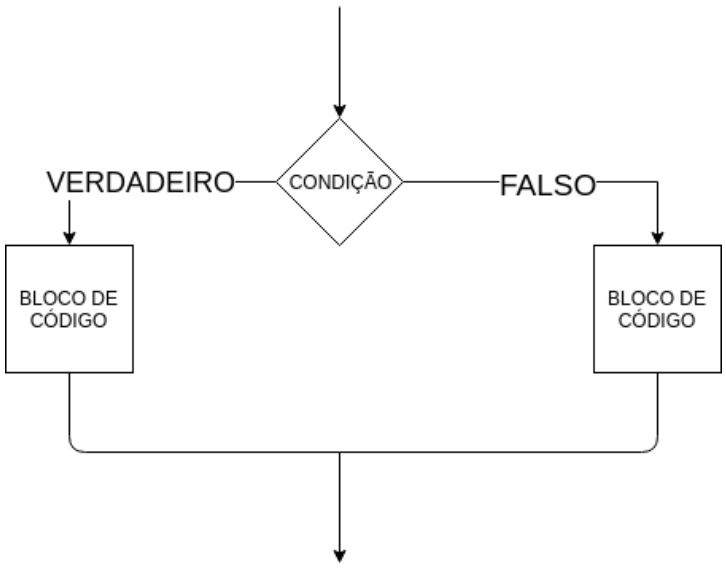
Veja que o **if** é estruturado com:

- a palavra reservada **if**
- a condição que determina se o código dentro do **if** será executado
- caractere **:** (dois pontos)
- trecho de código que será executado caso a condição seja satisfeita

Para encerrar um bloco **if** basta retornar a indentação para a coluna onde o código é alinhado fora do **if**. Por exemplo, no trecho
. . .
if condição:
    código1
    código2
código3
. . .
código1 e **código2** estão dentro do bloco **if**, enquanto **código3** está fora dele.

- else

No pseudocódigo acima, **código3 será executado independentemente do resultado da condição**. Porém, em algumas ocasiões, pode ser útil definir um trecho que só será executado caso a **condição** do **if não** seja verdadeira, ou seja, **caso a execução não entre no bloco dentro do if**, como no **fluxograma** abaixo:



A ferramenta que permite fazer isso é o **else**. A sua estrutura é bem semelhante à do **if**, porém ele não possui condição, uma vez que a sua condição de execução é exatamente o oposto da condição do **if**. Por essa razão, também, um bloco **else** só pode existir atrelado a um bloco **if**. Em outras palavras, **um else só pode existir após um if**. Veja o exemplo:

```
. . .
if numero > 3:
        print("O número é maior que 3")          # Indentação obrigatória
        . . .
else:
        print("O número não é maior que 3")
        . . .
. . .
```

Assim como **print("O número é maior que 3")** só é executado caso a condição do **if** seja verdadeira, **print("O número não é maior que 3") só será executado caso a condição do if seja falsa**.

Note ainda que ambas as opções são mutuamente exclusivas:

- se o **if** for executado, o **else** não será executado
- se o **if** não for executado, o **else** será executado

Sua estrutura também é bem parecida com a do **if**:

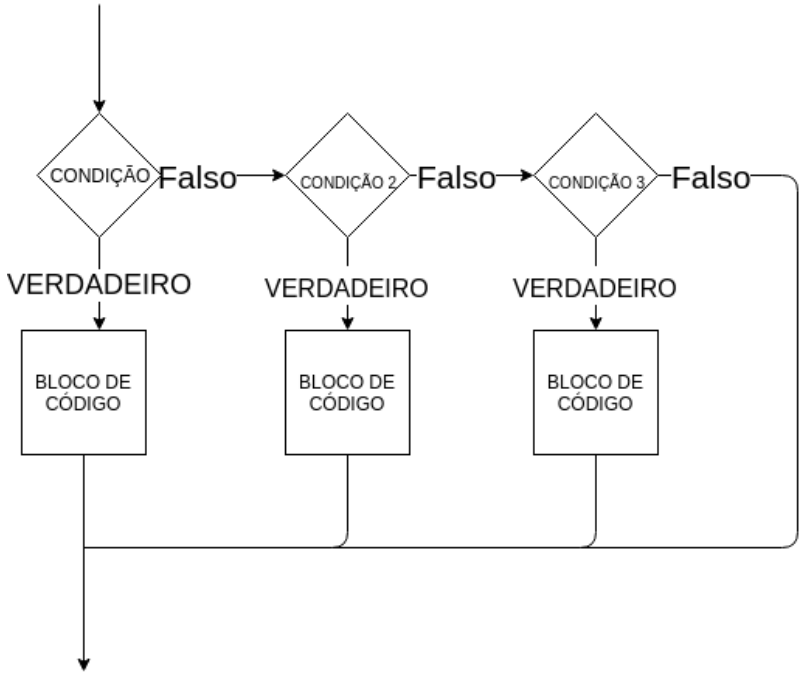
- a palavra reservada **else**
- caractere **:** (dois pontos)
- trecho de código que será executado caso a condição do **if** não seja satisfeita.

Para encerrar um bloco **else** basta retornar a indentação para a coluna onde o código é alinhado fora do **else**, assim como foi com o **if**.

- elif

Em alguns casos pode ser útil definir mais de uma condição alternativa ao **if**. Para fazer isso é necessário utilizar um **elif**.

Como o nome sugere, **elif** é um **else** com uma condição **if** extra. Mais ainda, ele permite que outro **elif** ou **else** seja usado em seguida.



No exemplo:

. . .
if horas < 0:
    print(“Hora inválida”)
elif horas < 6:
    print(“É madrugada”)
elif horas < 12:
    print(“É manhã”)
elif horas < 18:
    print(“É tarde”)
elif horas < 24:
    print(“É noite”)
else:
    print(“Hora inválida”)
. . .

suponha que **horas = 15**. A execução segue o seguinte caminho:

- **if horas < 0** falha, pois 15 < 0 é falso. O código irá testar o elif abaixo dele.
- **elif horas < 6** falha, pois 15 < 6 é falso. O código irá testar o elif abaixo dele.
- **elif horas < 12** falha, pois 15 < 12 é falso. O código irá testar o elif abaixo dele.
- **elif horas < 18** passa, pois 15 < 18 é verdade. O bloco referente a esse **elif** será executado, ou seja, mostrará a mensagem **“É tarde”**.
- todos os outros **elif-else** irão falhar, já que **elif horas < 18** executou.

É importante tomar cuidado com a ordem na qual os elif's são colocados, pois dependendo das condições fará total diferença. Por exemplo, o trecho

```
. . .
if numero > 3:
        print("O número é maior que 3")
elif numero > 5:
        print("O número é maior que 5")
. . .
```

Nunca retornará a mensagem dizendo que o número é maior que 5, pois

- um número maior que 3 irá fazer o **if** executar, o que implica que o **elif** será ignorado
- um número menor que 3, que faria o **if** não executar, também é necessariamente um número menor do que 5, fazendo o **elif** também não executar

A estrutura é exatamente a mesma do **if**, a única restrição é que o **elif** deve vir após um **if** ou outro **elif:**

- a palavra reservada **elif**
- a condição que determina se o código dentro do **elif** será executado
- caractere **:** (dois pontos)
- trecho de código que será executado caso a condição seja satisfeita

E para encerrar um **elif** a forma é a mesma do **if** e do **else**, retornando a indentação à coluna onde o código é alinhado antes dele.