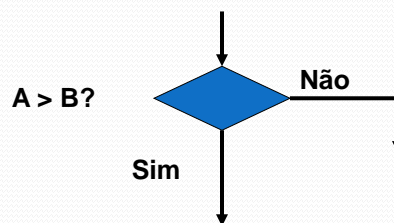


# Python - Comandos de Controle Condicional

Prof. André Backes

## Fluxogramas

- Condição ou Decisão
  - Representada por losangos
  - Normalmente contém uma pergunta do tipo Sim/Não ou um teste de Verdadeiro/Falso.
  - Representa uma mudança no fluxo do programa



## Comando if

- Na linguagem Python, o comando **if** é utilizado quando for necessário escolher entre dois caminhos dentro do programa ou quando se deseja executar um comando sujeito ao resultado de um teste.

## Comando if

- A forma geral de um comando **if** é:

```
if condição:  
    instrução 1  
    instrução 2  
    ...  
    instrução n  
  
continuação do programa
```

- A expressão, na condição, será avaliada:
  - Se a condição for verdadeira (**True**), a sequência de instruções será executada;
  - Se ela for falsa (**False**), a sequência de instruções será NÃO executada.

## Comando if

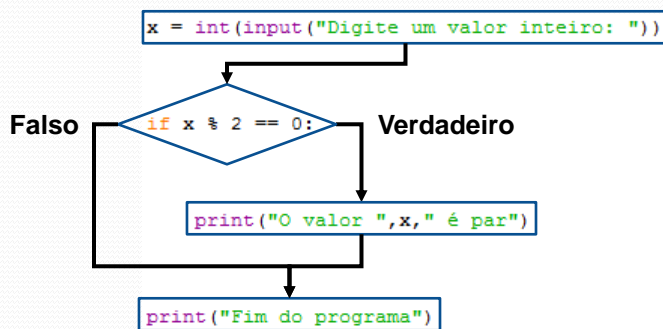
- Exemplo

```
x = int(input("Digite um valor inteiro: "))
if x % 2 == 0:
    print("O valor ", x, " é par")
print("Fim do programa")
```

- Saída

```
>>>
Digite um valor inteiro: 3
Fim do programa
>>> =====
>>>
Digite um valor inteiro: 4
O valor 4 é par
Fim do programa
>>>
```

## Comando if



## Condição do if

- A condição pode ser uma expressão usando operadores matemáticos, lógicos e relacionais
  - +, -, \*, / , %
  - and, or e not
  - >, <, >=, <=, ==, !=
- Ex:
  - $x > 10$  and  $y \leq x-1$
  - not ( $x > 0$ )

## Condição do if

- Tabela verdade

A	B	not A	not B	A and B	A or B
False	False	True	True	False	False
False	True	True	False	False	True
True	False	False	True	False	True
True	True	False	False	True	True

## Exercício

- Dada o valor da nota de um aluno, monte a expressão if que verifica se ele precisará fazer a sub. O aluno deverá fazer sub se sua nota for maior ou igual a 30 e menor do que 60.

## Exercício

- Solução

```
x = int(input("Digite a nota de um aluno: "))
if x < 60 and x >= 30:
    print("O aluno terá que fazer sub")

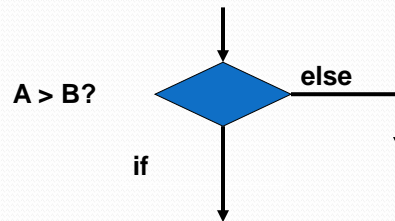
print("Fim do programa")
```

- Saídas

```
>>>
Digite a nota de um aluno: 45
O aluno terá que fazer sub
Fim do programa
>>> =====
>>>
Digite a nota de um aluno: 60
Fim do programa
>>> =====
>>>
Digite a nota de um aluno: 25
Fim do programa
>>>
```

## Comando else

- O comando **else** pode ser entendido como sendo um complemento do comando **if**.
- Se o **if** diz o que fazer quando a condição é **verdadeira**, o **else** trata da condição **falsa**.



## Comando else

- O comando **if-else** tem a seguinte forma geral:

```
if condição:
    instrução 1
    instrução 2
    ...
    instrução n
else:
    instrução 1
    instrução 2
    ...
    instrução n

continuação do programa
```

## Comando else

- A expressão da condição será avaliada:
  - Se a condição for verdadeira (**True**), a sequência de instruções do **if** será executada;
  - Se ela for falsa (**False**), a sequência de instruções do **else** será executada.
- Note que quando usamos a estrutura **if-else**, uma das duas declarações será executada.
  - O comando **else** não tem condição. Ele já é o caso contrário do **if**

## Exemplo if-else

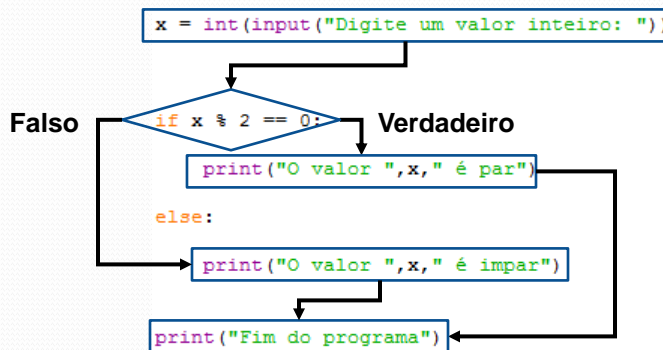
- Exemplo

```
x = int(input("Digite um valor inteiro: "))
if x % 2 == 0:
    print("O valor ",x," é par")
else:
    print("O valor ",x," é impar")
print("Fim do programa")
```

- Saída

```
>>>
Digite um valor inteiro: 5
O valor 5 é impar
Fim do programa
>>> =====
>>>
Digite um valor inteiro: 6
O valor 6 é par
Fim do programa
>>>
```

## Exemplo if-else



## Aninhamento de if

- O **if** aninhado é simplesmente um **if** dentro da declaração de um outro **if** (ou **else**) mais externo.
  - Essa estrutura é apenas uma extensão da estrutura **if-else**.
- O único cuidado que devemos ter é o de saber exatamente a qual **if** um determinado **else** está ligado.



## Aninhamento de if

```
if condição:
    instrução 1
    ...
    instrução n
else:
    if condição:
        instrução 1
        ...
        instrução n
    else:
        instrução 1
        ...
        instrução n
```

continuação do programa

```
if condição:
    if condição:
        instrução 1
        ...
        instrução n
    else:
        instrução 1
        ...
        instrução n
else:
    instrução 1
    ...
    instrução n
```

continuação do programa

## Aninhamento de if

- O programa começa a testar as condições começando pela primeira e continua a testar até achar uma expressão cujo resultado dê verdadeiro. Neste caso ele
  - Executa a sequência de comandos correspondente;
  - Só uma sequência de comandos será executada
  - A última sequência de comandos (default) é a que será executada no caso de todas as condições forem falsas e é opcional.

```
if condição:
    instrução 1
    ...
    instrução n
else:
    if condição:
        instrução 1
        ...
        instrução n
    else:
        instrução 1
        ...
        instrução n
```

continuação do programa

## Exemplo aninhamento

### Exemplo

```
x = int(input("Digite um valor inteiro: "))
if x == 0:
    print("O valor ",x," é igual a zero")
else:
    if x > 0:
        print("O valor ",x," é positivo")
    else:
        print("O valor ",x," é negativo")
print("Fim do programa")
```

### Saídas

```
>>>
Digite um valor inteiro: 0
O valor 0 é igual a zero
Fim do programa
>>> =====
>>>
Digite um valor inteiro: 12
O valor 12 é positivo
Fim do programa
>>> =====
>>>
Digite um valor inteiro: -4
O valor -4 é negativo
Fim do programa
```

## Exemplo aninhamento

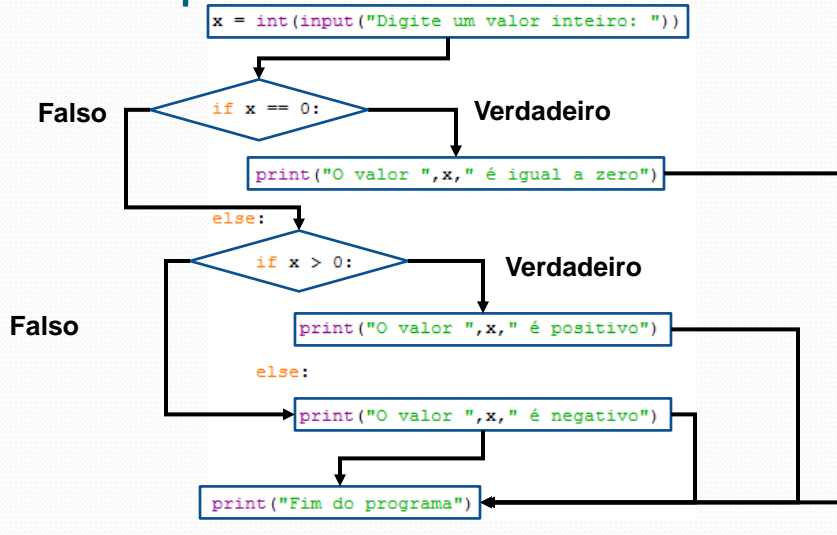
### Exemplo

```
x = int(input("Digite um valor inteiro: "))
if x != 0:
    if x > 0:
        print("O valor ",x," é positivo")
    else:
        print("O valor ",x," é negativo")
else:
    print("O valor ",x," é igual a zero")
print("Fim do programa")
```

### Saídas

```
>>>
Digite um valor inteiro: 0
O valor 0 é igual a zero
Fim do programa
>>> =====
>>>
Digite um valor inteiro: 12
O valor 12 é positivo
Fim do programa
>>> =====
>>>
Digite um valor inteiro: -4
O valor -4 é negativo
Fim do programa
```

## Exemplo aninhamento



## Aninhamento de if

- Não existe aninhamento de **else's**
  - Para cada **else** deve existir um **if** anterior, mas nem todo **if** precisa ter um **else**.

```

if condição:
    instrução 1
    instrução 2
    ...
    instrução n
else:
    instrução 1
    instrução 2
    ...
    instrução n
else:
    instrução 1
    instrução 2
    ...
    instrução n
  
```



## Exercício

- Dada o valor da nota de um aluno, monte o conjunto de if's e else's que verifica se ele foi aprovado, reprovado ou precisará fazer a sub.

## Exercício

### • Solução

```
x = int(input("Digite a nota de um aluno: "))
if x >= 60:
    print("O aluno está aprovado")
else:
    if x < 30:
        print("O aluno está reprovado")
    else:
        print("O aluno terá que fazer sub")
print("Fim do programa")
```

### • Saídas

```
>>>
Digite a nota de um aluno: 78
O aluno está aprovado
Fim do programa
>>> =====
>>>
Digite a nota de um aluno: 43
O aluno terá que fazer sub
Fim do programa
>>> =====
>>>
Digite a nota de um aluno: 22
O aluno está reprovado
Fim do programa
>>>
```

## Comando elif

- O comando **elif** pode ser entendido como sendo uma simplificação do aninhamento de um **if** dentro de um **else**.

```
if condição:
    instrução 1
    ...
    instrução n
else:
    if condição:
        instrução 1
        ...
        instrução n
    else:
        instrução 1
        ...
        instrução n
continuação do programa
```

```
if condição:
    instrução 1
    ...
    instrução n
elif condição:
    instrução 1
    ...
    instrução n
else:
    instrução 1
    ...
    instrução n
continuação do programa
```

## Comando elif

- Sem **elif**

```
x = int(input("Digite um valor inteiro: "))
if x == 0:
    print("O valor ",x," é igual a zero")
else:
    if x > 0:
        print("O valor ",x," é positivo")
    else:
        print("O valor ",x," é negativo")
print("Fim do programa")
```

- Com **elif**

```
x = int(input("Digite um valor inteiro: "))
if x == 0:
    print("O valor ",x," é igual a zero")
elif x > 0:
    print("O valor ",x," é positivo")
else:
    print("O valor ",x," é negativo")
print("Fim do programa")
```

## Comando elif

