

[◀ VOLTAR](#)

Estruturas de decisão: IF, ELIF e ELSE

Apresentar os comandos da estrutura de decisão da linguagem Python.

NESTE TÓPICO

> OPERADORES
RELACIONAIS

> OPERADORES LÓGICOS

> Dê uma olhada nos links
▼ Marcar
tópico

AUTOR(A)
PROF.
DENILSON
JOSE
SCHAFFER



Olá alunos,

Vamos ver um dos principais recursos da lógica de programação: os comandos de decisão.

O **if** executa um conjunto de comandos de acordo com uma ou mais condições.

Lembre-se que o **if** é um bloco de instruções dentro de um código, portanto devemos abrir e fechar este bloco. Mas como fazer isto em Python?

Na lógica, em português estruturado temos:

SE condição ENTÃO

 Comando;

FIM SE;

Agora em Python:

if condição:

comando

Para abrir: basta colocar o comando **if** (em letras minúsculas) e em outras linguagens, como Pascal, podemos utilizar a cláusula **THEN** representando o **ENTÃO**, em C podemos utilizar a chave "{", mas observe que em Python, basta utilizar os dois pontos ":" depois da condição. O interpretador então irá executar o comando ou os comandos que estão logo após os **dois pontos**.

Agora, note que os comandos do **if** estão com um recuo (indentação) e para encerrar o bloco **if**, basta, na próxima linha do código, tirar o recuo.

O interpretador vem executando linha a linha do código de acordo com a sequência programada, quando encontra um bloco **if**, o interpretador verifica a **condição do if**, se for **verdadeira**, o interpretador "entra" no bloco **if** e realiza o comando ou os comandos deste bloco, caso contrário, não entra no bloco **if** e continua a execução do código. Por isso, o bloco **if** também é conhecido como um desvio.

Podemos utilizar recursos opcionais em conjunto com o bloco **if** que são os comandos **else** e **elif**:

if condição:

relação_de_comandos

else:

relação_de_comandos

No caso acima, o **else** pertence ao bloco **if** e isto significa, que se o interpretador entrar neste bloco, não sairá sem realizar um dos comandos, pois se a condição do **if** não for verdadeira, serão executados o comando ou comandos do **else** (**jamais** coloque alguma condição no **else**).

if condição_1:

relação_de_comandos_1

elif condição_2:

relação_de_comandos_2

else:

relação_de_comandos_3

Uma outra opção é utilizar o comando **elif** que corresponde aos dois comandos juntos: **else + if**, que significa: senão se. Este recurso é utilizado quando há necessidade que o mesmo bloco **if** analise mais de uma condição e podemos utilizar quantos **elif** forem necessários em um mesmo bloco **if**.

Lembrando que para criar as condições, utilizaremos os seguintes operadores:

OPERADORES RELACIONAIS

> **Maior que**

< **Menor que**

>=	Maior ou Igual
<=	Menor ou Igual
==	Igual
!=	Diferente

OPERADORES LÓGICOS

and

or

not

Os dois pontos ":" depois da expressão ou condição no comando **if** ou **elif** e depois do **else**, indica o início do bloco de instrução a ser executado. É equivalente ao "then" em Pascal ou a "{" em C, Java.

if condição_1: <=== A partir daqui inicia a execução dos comandos

**** ---->** relação_de_comandos_1

elif condição_2: <=== A partir daqui inicia a execução dos comandos

relação_de_comandos_2

elif condição_3: <=== A partir daqui inicia a execução dos comandos

relação_de_comandos_3

else: <=== A partir daqui inicia a execução dos comandos

relação_de_comandos_4

EXEMPLO 1:

INDENTAÇÃO

Cuidado com a INDENTAÇÃO, se o bloco de instruções não for indentado, o programa não roda.

```
1. # Entra com um número e verifica se este número é par ou impar
2. vResultado = ""
3. vNum = int(input('Entre com um número: '))
4. if vNum % 2 == 0:
5.     vResultado = "PAR"
6. else:
7.     vResultado = "IMPAR"
8. print("O número é: ",vResultado) #Apresenta o resultado
```

Veja o resultado do exemplo:

```
1. Entre com um número: 45
2. O número é: IMPAR
3. >>>
4.
5. Entre com um número: 54
6. O número é: PAR
7. >>>
```

Neste exemplo **1**, na linha **2**, criamos uma variável **vResultado** e atribuímos um valor vazio entre aspas e o interpretador já vai considerar esta variável como tipo **string** (caractere).

Na linha **3**, utilizamos o comando **input** para receber um número inteiro via teclado, criando e atribuindo à variável **vNum** e determinamos como **int()**, tipo inteiro, para a variável **vNum**.

Na linha **4**, abriu-se um bloco **if**, passando uma condição: se a variável carregada com o valor dividida por 2, tiver como resto um valor igual a zero (o operador **%** retorna o resto de uma divisão), então a variável **vResultado** recebe "PAR".

Na linha **6**, temos um **else**, se a condição do **if** não for verdadeira, então a variável **vResultado** recebe "IMPAR" (linha **7**).

Na linha **8** temos o comando **print** que apresenta o resultado, este comando não pertence ao bloco **if**, por isso está sem recuo, o interpretador considerará que a linha **7** é a última linha de comando do bloco **if**. Neste caso, se você utilizasse o comando **print** com a indentação, alinhando à linha **7**, este comando faria parte do comando **else**.

EXEMPLO 2:

```
1. # Ao entrar com a média do aluno, classifica o conceito
2. media = float(input('Entre com a média do aluno: '))
3. if media <= 5:
4.     conceito = str("REGULAR")
5. elif media < 7:
6.     conceito = "BOM"
7. else:
8.     conceito = "EXCELENTE"
9. print("Conceito: ",conceito)
```

Veja o resultado do exemplo 2:

```
1. Entre com a média do aluno: 4.5
2. Conceito:  REGULAR
3. >>>
4.
5. Entre com a média do aluno: 6.5
6. Conceito:  BOM
7. >>>
8.
9. Entre com a média do aluno: 9
10. Conceito:  EXCELENTE
1. >>>
```

Na linha **2**, já criamos a variável **media**, atribuímos um valor de entrada via teclado e determinamos que a variável será de tipo **float** colocando a função **input** dentro do **float ()**.

Na linha **3**, abrimos um bloco **if**, passamos uma condição: se a média for menor ou até 5,0 então criamos a variável **conceito**, definimos como tipo string (caractere) com **str()** e atribuímos o valor “REGULAR”.

Na linha **5**, passamos uma outra condição com **elif** (senão se) se a média for maior que 5,0 e menor que 7,0, o conceito será “BOM”.

Na linha **7**, colocamos um **else**, que significa se nenhuma das condições anteriores forem verdadeiras, então será atribuído à variável **conceito** o valor de ‘EXCELENTE’.

Na última linha mostramos o valor que estará carregado na variável **conceito**: se a média for de 0 a 5,0 o conceito será regular, se a média for entre 5,5 até 6,5 o conceito será bom e se a média for entre 7,0 e 10 o conceito será excelente.

EXERCÍCIO:

Crie um script para calcular o desconto do INSS, de acordo com o salário informado pelo usuário, conforme a tabela de desconto abaixo:

Salário	Desconto
Até R\$ 1693,72	8%
De R\$ 1693,73 a R\$ 2822,90	9%
De R\$ 2822,91 a R\$ 5645,80	11%
Acima de R\$ 5645,80	O desconto é de R\$ 621,04.

Tente fazer o exercício antes de olhar a resposta abaixo!

```
1. print("\n* * * MOSTRA O DESCONTO DO INSS! * * *\n")
2. salario = float(input('Entre com o salário: '))
3. if salario <= 1693.72:
4.     inss = float(salario * 0.08)
5. elif salario >= 1693.73 and salario <= 2822.90:
6.     inss = salario * 0.09
7. elif salario >= 2822.91 and salario <= 5645.80:
8.     inss = salario * 0.11
9. else:
10.    inss = 621.04
11. print("O desconto do INSS será de: \n", "%.2f"% inss)
12. input('Pressione ENTER para sair...')
```

A penúltima linha mostra o desconto do INSS conforme o salário digitado, a parte do comando **print** com **"%.2f"% inss** é uma formatação do valor da variável **inss**, como o resultado pode mostrar várias casas decimais, então estamos determinando para considerar somente até duas casas decimais: **2f** (já que estamos trabalhando com centavos). Por exemplo, se fosse para considerar quatro casas decimais para o pi (3.1416), então poderíamos formatar: **"%.4f"% pi**. Também poderemos utilizar um outro recurso que produz o mesmo resultado, a função **format** da seguinte forma: **"{: .2f}".format(inss)**.

```
1.  * * * MOSTRA O DESCONTO DO INSS! * * *
2.
3.  Entre com o salário: 2645.86
4.  O desconto do INSS será de:
5.    238.13
6.  Pressione ENTER para sair...
7.  >>>
8.
9.
10. * * * MOSTRA O DESCONTO DO INSS! * * *
11.
12. Entre com o salário: 26458.66
13. O desconto do INSS será de:
14.    621.04
15. Pressione ENTER para sair...
16. >>>
```

SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:

<https://www.python.org/doc/> (<https://www.python.org/doc/>)

<https://wiki.python.org/moin/PythonBooks>
(<https://wiki.python.org/moin/PythonBooks>)

Neste tópico vimos um dos principais recursos da lógica para resolvermos problemas: o bloco **if**, sua estrutura, os operadores com o qual trabalha e seus recursos opcionais como o comando **else** e **elif**.

Quiz

Exercício Final

Estruturas de decisão: IF, ELIF e ELSE

[INICIAR >](#)

Referências

SUMMERFIELD, M. *Programação em Python 3*: Uma introdução completa à linguagem Python. Rio de Janeiro Alta Books, 2012. 495 p.

MENEZES, N. N. C. *Introdução à programação com Python*: algoritmos e lógica de programação para iniciantes. 2. ed. São Paulo: Novatec, 2014. 328 p.

SWEIGART, AL. *Automatize tarefas maçantes com Python*: programação prática para verdadeiros iniciantes. São Paulo: Novatec, 2015. 568 p.

PYTHON, doc. Disponível em: <<https://www.python.org/doc/>>. Acesso em: Junho/2018.

PYTHON, books. Disponível em: <<https://wiki.python.org/moin/PythonBooks>>. Acesso em: Junho/2018.



Avalie este tópico



ANTERIOR

Desvio condicional encadeado



Índice

Biblioteca
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

PRÓXIMO

Estruturas de repetição: encaixada (https://ava.uninove.br/ava/uninove/estrutura-de-repeticao-encaixada.php?idCurso=)

© Todos os direitos reservados

