

## ESTRUTURA DE CONTROLE DE EXECUÇÃO

**\*Lógica If:** a lógica If é usada como condição. Se a sentença for verdadeira execute o código do bloco if, caso contrário não execute. O Python trabalha com indentação, ou seja, para definir os blocos que serão executados deve ser colocado uma tabulação para cada bloco, de forma geral da seguinte maneira:

if <condição>

    <bloco de código indentado>

<instrução não indentada>

Veja o exemplo:

```
1 temp = eval(input('Digite a temperatura: ')) #usuário entra com a temp
2
3 if temp>36: #bloco if, se temperatura digitada for maior que 36, execute o bloco indentado
4     print('Está muito quente!!!') #essa linha esta indentada
```

```
Digite a temperatura: 40
Está muito quente!!!
```

Note que se a condição não for verdadeira, nada irá acontecer pois não há mais linhas de códigos sem a indentação. Porém é possível tratar um novo caminho, caso a afirmação não seja verdadeira, para isso usamos o if com dois caminhos, acrescentando a estrutura a palavra `else`, da seguinte maneira:

```
1 temp = eval(input('Digite a temperatura: ')) #usuário entra com a temp
2
3 if temp>36: #bloco if, se temperatura digitada for maior que 36, execute o bloco indentado
4     print('Está muito quente!!!') #essa linha esta indentada
5 else: #caso a temp digitada seja menor ou até igual a 36 a linha abaixo sera executada
6     print('Não está tão quente!!!') #bloco else, note que ele também esta indentado
```

```
Digite a temperatura: 20
Não está tão quente!!!
```

Exemplo: Fazer um programa Python que calcule a média do aluno. Sendo fornecido pelo usuário as notas de P1, P2, quiz e outros. Os pesos são: Média de P1 e P2 = 0,6. Quiz e outros 0,2 cada um.

```
1 p1=eval(input('P1= '))
2 p2=eval(input('P2= '))
3 quiz=eval(input('quiz= '))
4 outros=eval(input('outros= '))
5
6 media=(p1+p2)*0.3+(quiz+outros)*0.2
7
8 if media >=5:
9     print('Aluno aprovado com a média: ',media)
10 else:
11     print('Aluno reprovado com a média: ', media)
```

**\*Lógica de for:** essa lógica é de iteração, ou seja, realizar uma operação sobre cada objeto na sequência (pode ser uma string ou uma lista). Veja dois exemplos:

```
1 name=input('Digite uma palavra: ')
2 print('A palavra soletrada: ')
3
4 for char in name:
5     print(char)
```

```
Digite uma palavra: Teste
A palavra soletrada:
T
e
s
t
e
```

```
1 animais=['peixe','gato','cão']
2 for animal in animais:
3     print(animal)
```

```
peixe
gato
cão
```

A estrutura do for é:

```
for <variavel> in <sequencia>
    <bloco de código indentado>
<bloco de código não indentado>
```

É possível, e muitas vezes necessário, usar outras estruturas dentro do bloco indentado, da seguinte maneira:

```
1 frase = input('Digite uma frase: ')
2
3 for c in frase:
4     if c in 'aeiouAEIOU': #note a dupla indentação nesse bloco If
5         print(c)
```

```
Digite uma frase: Uma frase de teste
U
a
a
e
e
e
e
```

**\* A função range():** essa função gera uma sequência de números dentro de uma faixa definida pelo programador:

```
1 for i in range(5):
2     print(i)
```

```
0
1
2
3
4
```

A função range, também aceita que seja definido qual o valor inicial da sequência que será gerada, da seguinte forma: range(inicio, final), conforme o exemplo:

```
1 for i in range(2,5):  
2     print(i)
```

```
2  
3  
4
```

E também é possível que um terceiro elemento seja colocado na função, esse terceiro elemento irá definir o passo (incremento) da sequência. Da seguinte forma: range( inicio, fim, passo), conforme o exemplo:

```
1 for i in range(1,14,3):  
2     print(i)
```

```
1  
4  
7  
10  
13
```

Todos os exemplos e textos foram retirados do livro: Introdução à programação usando Python: um foco no desenvolvimento de aplicações. Autor Ljubomir Perkovic, editora LTC.