

Introdução

Nesta aula usaremos o comando condicional **if..else** para introduzir decisões em um programa. A tabela abaixo resume a sintaxe desse comando:

Algoritmo	Sintaxe Python	Exemplo
<pre>se condição_verdadeira: <comando(s) 1> senão: <comando(s) 2></pre>	<pre>if condição: <comando(s) 1> else: <comando(s) 2></pre>	<pre>if x > 0: x = x + 1 print(x) else: y = x - 1 print(y)</pre>

Nesse caso, se a “condição” for verdadeira, o bloco **<comando(s) 1>** será executado. Caso contrário, o programa executará o bloco **<comando(s) 2>**.

Lembre-se que, na linguagem Python, comandos em um mesmo bloco DEVEM TER A MESMA INDENTAÇÃO, ou seja, mesmo espaçamento antes de cada um dos comandos do bloco, como mostrado no exemplo acima.

Veja o exemplo mais completo a seguir:

```
x = float( input('Entre com um número qualquer: '))
if x > 0:
    x = x + 1
    print('x = ', x )
else:
    y = x - 1
    print('y = ', y )
```

O que será escrito na tela pelo programa acima se você digitar o valor -5? E se você entrar com 10? Se estiver em dúvida, execute esse programa dentro do IDLE para ver o que acontece.

Dicas de Indentação no IDLE

Ao digitar os dois-pontos (:) após da condição do **if** e pressionar **Enter**, o IDLE já faz a indentação da linha de baixo automaticamente. Para remover a indentação, basta usar a tecla **Backspace** (seta para a esquerda acima da tecla **Enter**). Seguem mais alguns atalhos:

Efeito	Tecla
Indentar a linha atual	Tab
Des-indentar a linha atual	Backspace
Indentar várias linhas (já selecionadas) ao mesmo tempo	Ctrl +]
Des-indentar várias linhas (já selecionadas) ao mesmo tempo	Ctrl + [

Roteiro de Prática

Importante: Como qualquer outra prática de INF100 você deve:

1. Criar o cabeçalho obrigatório (sem o cabeçalho com sua identificação, o valor da prática será 0)
2. Após finalizar o cabeçalho salve o arquivo com o nome correto
3. Leia as instruções até o final e, após finalizar sua leitura, inicie sua programação.

Nome do arquivo a ser entregue: **p03.py**

Obs.: Recomenda-se salvar o arquivo com certa frequência para não perder a digitação já feita em caso de uma falha na rede elétrica.

Roedores são mamíferos caracterizados por um único par de incisivos que crescem continuamente, em cada uma das mandíbulas. Exemplos de roedores bem conhecidos são ratos, esquilos, pacas, cutias e castores. O maior roedor do mundo é a **Capivara**. Adultos típicos pesam entre 35 e 66 kg. O maior peso registrado é de 91 kg para uma fêmea selvagem do Brasil.



Escreva um programa que pede ao usuário para digitar quatro valores reais (float), representando os pesos de quatro capivaras. À medida que cada peso for lido, o programa deve somar o peso lido se ele for maior que zero. Caso contrário, deve escrever na tela a mensagem “Peso inválido!”. No final, o programa deve escrever na tela o peso médio, considerando somente os pesos válidos (maiores que zero), e logo abaixo o número de pesos válidos considerado. Mas se nenhum peso válido for fornecido, o programa deve exibir a mensagem mostrada nos exemplos.

Seguem alguns exemplos de “telas” de execução desse programa. As entradas de dados do usuário (pelo teclado) estão **destacadas**.

Exemplo 1:

Peso da capivara 1: **45**
Peso da capivara 2: **55.3**
Peso da capivara 3: **10.5**
Peso da capivara 4: **83.1**

Peso médio: 48.5 kg
(considerando 4 pesos válidos)

Obs.: o peso médio deve ser impresso com largura 5 e uma casa decimal.

Exemplo 2:

Peso da capivara 1: **0**
Peso inválido!
Peso da capivara 2: **10.5**
Peso da capivara 3: **-1**
Peso inválido!
Peso da capivara 4: **12.5**

Peso médio: 11.5 kg
(considerando 2 pesos válidos)

Exemplo 3:

Peso da capivara 1: **90**
Peso da capivara 2: **90**
Peso da capivara 3: **90**
Peso da capivara 4: **90**

Peso médio: 90.0 kg
(considerando 4 pesos válidos)

Exemplo 4:

```
Peso da capivara 1: 0
Peso inválido!
Peso da capivara 2: 0
Peso inválido!
Peso da capivara 3: 0
Peso inválido!
Peso da capivara 4: 0
Peso inválido!

Nenhum peso válido fornecido.
```

Primeira tentativa de um algoritmo para resolução do problema:

```
leia o peso de uma capivara
se o peso não for maior que zero:
    escreva "Peso inválido"
```

Repita esse trecho 4 vezes (uma para cada capivara). Um comando condicional é usado para cada peso lido.

```
peso médio = (peso 1 + peso 2 + peso 3 + peso 4) / 4
escreva o peso médio
```

Existem pelo menos dois problemas na abordagem acima:

- Se o usuário digita pesos inválidos, eles serão considerados no cálculo do peso médio (penúltima linha), mas eles deveriam ser descartados. Ex.: O valor -1 deve ser descartado no exemplo 2.
- Para o cálculo do peso médio, o divisor utilizado não deve ser sempre 4. O correto é utilizar o número de pesos válidos. Ex.: No segundo exemplo, o total de pesos válidos é 2.

Conclusão: a média deve considerar apenas os pesos válidos e a quantidade de pesos válidos. Uma abordagem correta deve levar isso em consideração!

Vamos criar uma nova abordagem que considere apenas os pesos válidos. A ideia é de utilizar duas variáveis para armazenar a quantidade de pesos válidos e a soma dos pesos válidos após cada entrada do usuário. Essas variáveis devem iniciar com o valor 0 (zero), uma vez que, no início do programa, nenhum valor válido terá sido digitado. As variáveis deverão ser atualizadas apenas quando o valor lido for válido.

Veja o algoritmo correto para a resolução do problema:

```
soma = 0 # essa variável armazenará a soma dos pesos válidos lidos
n = 0 # essa variável armazenará a quantidade de pesos válidos lidos
```

```
leia o peso de uma capivara
se o peso for maior que zero:
    soma = soma + peso
    n = n + 1
senão:
    Escreva "Peso inválido!"
```

Repita (copie) esse trecho de código 4 vezes (uma para cada

```
se n for igual a zero:
    Escreva "Nenhum peso válido fornecido."
senão:
    Calcule o peso médio (dividindo soma por n)
    Escreva o peso médio
    Escreva a quantidade de pesos válidos
```

Repare que, para esse problema, não é necessário criar uma variável diferente para o peso de cada capivara. Você pode usar uma única variável "peso" para todas as quatro leituras.

⚠ A saída do programa deve obedecer à formatação **exata** mostrada nos exemplos acima.

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p03.py**) através do sistema do LBI.

O prazo para a entrega vai até 1h49 após o início da sua aula, pelo sistema do LBI (<http://linux-server.lbi.ufv.br>).