

## Introdução

Nesta prática usaremos o comando repetitivo *while*. Não se preocupe, caso ainda não tenha tido a apresentação deste comando na aula teórica. Todas as informações necessárias para a execução deste roteiro estão explicadas aqui. Você também pode assistir a vídeo aula sobre comando repetitivo (parte 1), em caso de dúvidas. A tabela abaixo resume a sintaxe desse comando:

Algoritmo	Sintaxe Python	Exemplo
<code>&lt;comando-seq1&gt;</code>  <code>enquanto condição:</code> <code>&lt;comando-rep&gt;</code>  <code>&lt;comando-seq2&gt;</code>	<code>&lt;comando-seq1&gt;</code>  <code>while condição:</code> <code>&lt;comando-rep&gt;</code>  <code>&lt;comando-seq2&gt;</code>	<code>x = 0</code>  <code>while x &lt; 3 :</code> <code>print('rep:', x)</code> <code>x = x + 1</code>  <code>print('seq: ', x)</code>

Nesse caso, quando a “condição” for verdadeira, o bloco **<comando-rep>** será executado e ao final, do bloco de comandos repetitivos, o fluxo de execução retorna à cláusula enquanto para testar novamente a “condição”. Caso a condição continue verdadeira, uma nova execução do bloco **<comando-rep>** acontecerá. Ou seja, enquanto a condição for verdadeira, o comando repetitivo continuará a ser executado. Note que um comando repetitivo com esta estrutura pode ser executado zero ou mais vezes. Se após a execução do comando sequencial `<comando-seq1>`, na primeira vez em que a condição do comando repetitivo for testada, o resultado do teste for falso, o bloco repetitivo **<comando-rep>** não será executado.

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 = 0
while x < 3:
    print('rep:', x)
    x = x + 1
print('seq: ', x)
```

A saída gerada pelo código acima será a mostrada na figura a seguir.

```
rep: 0
rep: 1
rep: 2
seq: 3
```

Os textos “rep: ” e “seq: ” foi colocado no comando `print()` para ficar mais fácil de identificar a execução dentro e fora do bloco de comandos repetitivos. Note que o bloco de comandos repetitivos foi executado 3 vezes (para  $x = 0, 1$  e  $2$ ). Quando o valor de  $x$  foi incrementado na terceira execução do bloco repetitivo e passou a valer 3, no próximo teste da condição ( $x < 3$ ) o resultado foi falso e o comando repetitivo terminou.

## Lembrete de dicas de Indentação no IDLE

Ao digitar os dois-pontos ( : ) após da condição do **if** e pressionar **Enter**, o IDLE já faz a indentação CORRETA 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(s)
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 + [

**OBS:** Ctrl + ] significa, manter a tecla Ctrl pressionada e apertar 1 vez a tecla ]

## Instruções

Nome do arquivo a ser entregue: **p04.py**

**A entrega deverá ser feita até 23h59 do dia 08/10/2020.**

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

Faça o “download” do arquivo **p04.py**. Em seguida, entre no IDLE e abra o arquivo através do menu **File > Open....** Segue abaixo um código semelhante ao que você deverá ver na tela:

```
# Nome do aluno:
# Matrícula:
# Data:

def q1(): ← Não apagar
    # Escreva aqui um breve comentário dizendo o que o programa 1 faz
    #                                     ← Espaço para o comentário
    # Escreva o código para a resolução da questão 1 abaixo
    print('solução da questão 1') # apague ou comente este comando!!!

    #Espaço para o código da q1


def q2():
    # Escreva aqui um breve comentário dizendo o que o programa 2 faz
    #
    # Escreva o código para a resolução da questão 2 abaixo
    print('solução da questão 2') # apague ou comente este comando!!!


    #Espaço para o código da q2
```

Para cada questão, utilize o espaço indicado para incluir o seu código. Não edite o cabeçalho das questões, como mostrado acima.

Para executar o seu programa e verificar a resposta, pressione F5 e digite (**na janela do Shell**) a letra **q**, o **número da questão** que quer executar, **parênteses** e pressione **Enter**, da seguinte forma:

```
>>> q2()  
solução da questão 2  
>>>
```

 As saídas dos programas devem obedecer à formatação **exata** mostrada nos exemplos.


 Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição para cada programa. Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p04.py**) através do sistema do LBI.

## Roteiro de Prática

Use o arquivo **p04.py** para implementar a solução das duas questões a seguir.

**Use também o código da sua prática 01, pois os dois problemas da prática 04 são os mesmos da prática 01, porém com novas tarefas para as quais você terá que utilizar comandos repetitivos *while*.**

**ATENÇÃO:** Não usem nomes de variáveis com caracteres acentuados (ç, ã, á, à, etc.)! Primeiramente, isso deve ser evitado pois na maioria das linguagens de programação dá erro. Mas, principalmente porque mesmo sendo um programa Python, o código de caracteres varia entre os sistemas operacionais e o seu programa pode não ser interpretado corretamente e, conseqüentemente **NÃO FUNCIONAR** em outra máquina (por exemplo, a máquina de quem vai corrigir o seu trabalho).

 A saída do seu programa deve obedecer à formatação **exata** mostrada nos exemplos a seguir.

- 1) Escreva um programa para ler o nome de uma cidade e os valores da temperatura máxima e mínima de um mesmo dia, em graus Celsius. Após ler as informações, o programa deverá informar qual foi a amplitude térmica (diferença entre a temperatura máxima e mínima) e a temperatura média (considerando apenas os valores lidos), apresentando também as temperaturas lidas.

A leitura do valor da temperatura máxima não precisa ser verificada. Entretanto, a leitura do valor da temperatura mínima deve ser verificada pelo programa. Caso, o usuário forneça um valor para a temperatura mínima, o programa deve exibir uma mensagem de advertência para o usuário ("A temperatura mínima não pode ser maior que a máxima") e repetir o comando de leitura. Enquanto for fornecida uma temperatura inválida, o programa deverá repetir o procedimento aqui descrito.

Os valores de temperatura deverão ser apresentados nas escalas Celsius e Fahrenheit. A conversão pode ser feita usando a fórmula:

$$\frac{C}{5} = \frac{F - 32}{9}$$

Onde, C é a temperatura em graus Celsius e F é a temperatura em graus Fahrenheit.

A figura a seguir mostra como deve ser a saída gerada pelo seu programa. Preste atenção nos alinhamentos dos valores impressos.

```
>>> q1()
Informe o nome da cidade: Viçosa
Informe o valor da temperatura máxima (°C): 38
Informe o valor da temperatura mínima (°C): 39
A temperatura mínima não pode ser maior que a máxima
Informe o valor da temperatura mínima (°C): 42
A temperatura mínima não pode ser maior que a máxima
Informe o valor da temperatura mínima (°C): 39.7
A temperatura mínima não pode ser maior que a máxima
Informe o valor da temperatura mínima (°C): 29.7

Em Viçosa as condições de temperatura foram:

          Mínima Máxima Média Amplitude
Graus Celsius    29.7   38.0   33.9      8.3
Graus Fahrenheit  85.5  100.4  92.9     14.9
>>>
```

- 2) Uma Progressão Aritmética é uma sequência de números que tem o primeiro elemento ( $a_1$ ) como o elemento base e os demais elementos são gerados de forma recorrente seguindo a propriedade de que o próximo elemento é igual ao anterior mais a razão ( $r$ ). Ou seja:

$$a_k = a_{k-1} + \text{razão}$$

Escreva um programa para informar o valor de todos os N primeiros elementos de uma Progressão Aritmética (PA) e a soma destes N primeiros elementos. A soma dos N primeiros elementos deve ser feita, obrigatoriamente, com a totalização dos valores à medida que eles forem sendo gerados.

O programa deverá solicitar ao usuário o valor do primeiro elemento da PA, a razão da PA e o valor de N.


O programa deverá gerar uma saída como a mostrada na figura mostrada a seguir.

```
>>> q2()
Este programa calcula o valor do elemento da posição N e a soma
dos N primeiros número de uma Progressão Aritmética (PA)

Informe o valor do primeiro elemento da PA: 0.755
Informe o valor da razão da PA: 0.255
Informe o valor de N: 8

PA = {0.755, 1.010, 1.265, 1.520, 1.775, 2.030, 2.285, 2.540}

O 8º elemento da PA é = 2.540
A soma dos 8 primeiros elementos é = 13.180
>>>
```

 A saída do seu programa deve obedecer à formatação **exata** mostrada no exemplo acima.

Após certificar-se que seu programa, com o código das questões 1 e 2, está correto, envie o arquivo do programa fonte (**p04.py**) através do sistema do LBI.

**A entrega deverá ser feita até 23h59 do dia 08/10/2020.**