

## 1. Introdução

### 1.1. Comando de Saída (print)

Considere o seguinte programa bem pequeno escrito em Python:

```
peso = 75.57  
print('Peso:', peso, 'kg')
```

Esse programa escreverá na tela o seguinte:

```
Peso: 75.57 kg
```

Note que o comando **print()** escreverá o texto 'Peso:', em seguida o valor da variável **peso** definida anteriormente, e logo depois o outro texto 'kg'. Note também que o **print()** insere um espaço em branco entre cada uma dessas partes, que chamamos de “parâmetros”.

Podemos obter um resultado muito parecido usando um recurso interessante de formatação de números e textos disponível em Python (vamos supor que a variável **peso** já esteja definida, como acima):

```
print('Peso: %.2f kg' % peso )
```

Nesse caso, o ‘%.2f’ indica que se deseja formatar um valor real, também chamado de “valor em ponto flutuante”, ou simplesmente **float**, usando duas casas decimais. Nesse caso devemos usar um % em vez da vírgula para separar a variável **peso** do texto.

Se eu quiser que o peso seja escrito com UMA casa decimal, basta fazer:

```
print('Peso: %.1fkg' % peso )
```

e o resultado será (note que neste último comando não existe espaço entre o “%.1f” e o “kg”):

```
Peso: 75.6kg
```

Podemos também controlar a “largura” que o valor ocupará na tela, assim:

```
print('Peso:%10.2f kg' % peso )
```

Nesse caso, o resultado será (Note que o espaço entre “%10.2f” e o “kg” voltou!):

```
Peso:      75.57 kg
```

Como o valor propriamente dito (75.57) ocupa 5 posições na tela, são adicionados 5 espaços em branco para completar a largura de 10. Veja que nesse exemplo removemos o espaço em branco logo depois do ‘:’, caso contrário teríamos 6 espaços em branco entre o ‘:’ e o primeiro dígito do peso.

Podemos também formatar valores inteiros, usando o formatador ‘%d’ (de “decimal”) em vez do ‘%f’. Obviamente, por se tratar de valor inteiro, no %d se admite apenas um parâmetro numérico para especificar o espaço total a ser ocupado pelo valor (não existem casas decimais):

```
idade = 18  
print('Tenho%4d anos.' % idade )
```

Nesse caso, o resultado será:

```
Tenho  18 anos.
```

Para preencher o lado esquerdo do número com zeros em vez de espaços, fazemos assim (supondo sempre que o valor da variável esteja definido):

```
print('Tenho%04d anos.' % idade )
```

Nesse caso, o resultado será:

```
Tenho0018 anos.
```

Para formatar variáveis do tipo “texto” em vez de números, usamos o formatador “%s” (de *string*):

```
jantar = 'pizza'
print('Ontem jantei %s.' % jantar )
```

Isso escreverá na tela o seguinte:

```
Ontem jantei pizza.
```

Usando então esses formatadores e espaços em branco inseridos manualmente no texto entre aspas, podemos formatar e alinhar os valores da forma como quisermos.

Podemos também inserir vários números e textos na mesma linha escrita em tela. Só precisamos tomar o cuidado de colocar a lista de variáveis entre parênteses. Segue um exemplo:

```
idade = 18
rango = 'pizzas'
print('Tenho %d anos e ontem comi %02d %s.' % (idade, 3, rango) )
```

Isso escreverá na tela o seguinte:

```
Tenho 18 anos e ontem comi 03 pizzas.
```

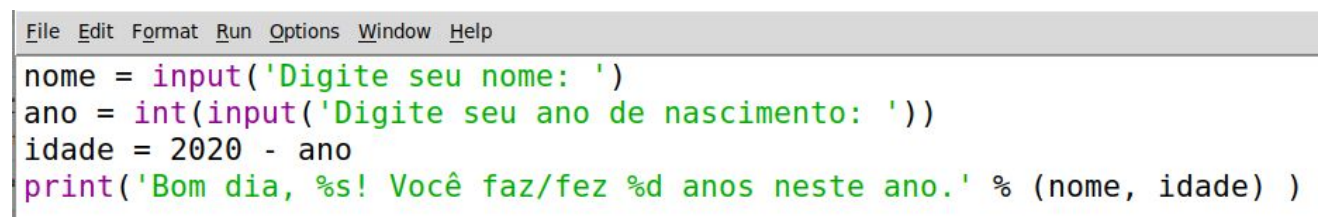
## 1.2. Comando de Entrada (input)

Para permitir que o usuário do programa entre com um valor, devemos usar o comando input:

```
nome = input('Digite seu nome: ')
```

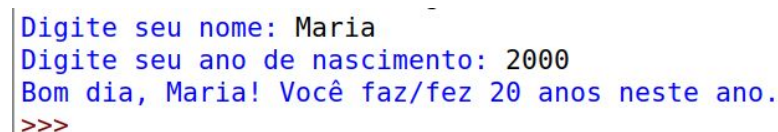
dessa forma, o texto 'Digite seu nome: ' será exibido na tela e o programa aguardará até que o usuário entre com o texto e pressione a tecla Enter. O texto digitado pelo usuário será armazenado na variável nome.

Pode-se também trabalhar com valores inteiros e reais através da conversão do texto digitado para valores numéricos. Para isso pode-se usar as palavras-chave *int* e *float* para números inteiros e reais respectivamente. Veja um exemplo completo com a leitura de um texto (nome do usuário do programa) e um número inteiro (ano de nascimento do usuário do programa). A imagem abaixo é de um trecho da janela do editor de programas do IDLE.



```
File Edit Format Run Options Window Help
nome = input('Digite seu nome: ')
ano = int(input('Digite seu ano de nascimento: '))
idade = 2020 - ano
print('Bom dia, %s! Você faz/fez %d anos neste ano.' % (nome, idade) )
```

Uma possível saída para este programa seria como a mostrada a seguir, na imagem da janela do Shell do IDLE, onde o texto escrito pelo programa é azul e o fornecido pelo usuário do programa é preto.



```
Digite seu nome: Maria
Digite seu ano de nascimento: 2000
Bom dia, Maria! Você faz/fez 20 anos neste ano.
>>>
```

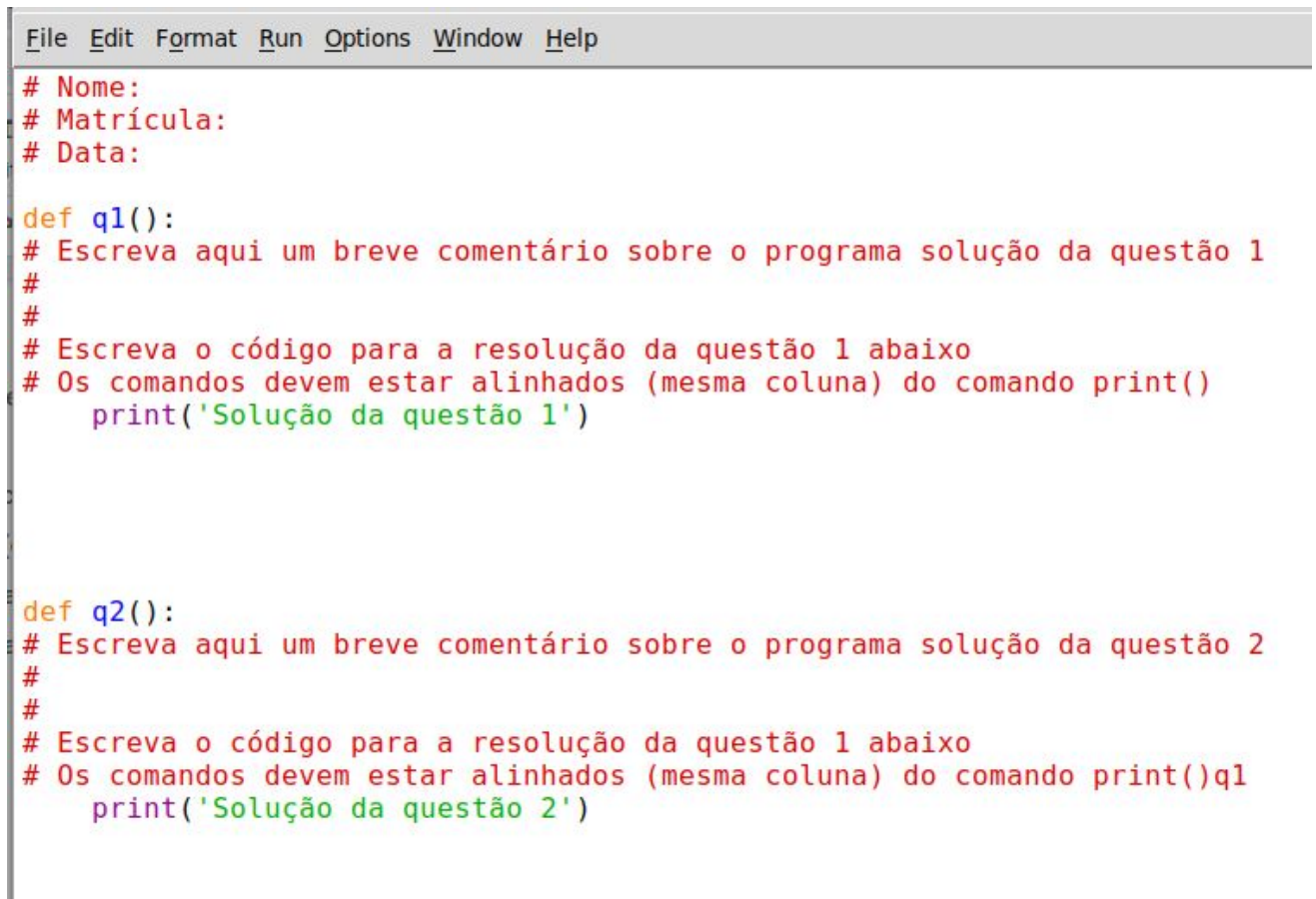
## Instruções para acessar o material da prática 01

Caso você esteja tentando fazer fora de seu horário de aula prática, acesse o LBI em <http://linux-server.lbi.ufv.br> com seu número de matrícula e a senha criada na Aula Prática 00.

Vá em Atividades e faça o “download” do arquivo **p01.py**. O outro arquivo “Roteiro da Prática 01” é este arquivo que você está lendo.

Caso você esteja no horário de sua aula prática, o(a) seu(ua) professor(a) já te informou que você deveria ter feito o download dos 2 arquivos.

Em seguida, entre no IDLE e abra o arquivo através do menu **File > Open....** Segue abaixo a imagem da janela de programas do IDLE, com o código que você deverá ver na tela:



```
File Edit Format Run Options Window Help
# Nome:
# Matrícula:
# Data:

def q1():
# Escreva aqui um breve comentário sobre o programa solução da questão 1
#
#
# Escreva o código para a resolução da questão 1 abaixo
# Os comandos devem estar alinhados (mesma coluna) do comando print()
    print('Solução da questão 1')

def q2():
# Escreva aqui um breve comentário sobre o programa solução da questão 2
#
#
# Escreva o código para a resolução da questão 1 abaixo
# Os comandos devem estar alinhados (mesma coluna) do comando print()q1
    print('Solução da questão 2')
```

Você terá que completar o arquivo p01.py com o código Python para a solução das duas questões propostas mais adiante. Portanto, o nome do arquivo a ser entregue é:

### **p01.py**

Para cada questão, utilize o espaço indicado para incluir o seu código, como mostrado acima. Não altere nada nas linhas:

```
def q1():
```

e

```
def q2():
```

Cada uma das questões está com apenas um comando de impressão indicando seu número.

Para executar uma questão do seu programa e verificar a resposta, pressione a tecla F5 e digite a letra **q**, o **número da questão** que quer executar, **parênteses** e pressione **Enter**. A figura abaixo, mostra a execução (janela do Shell do IDLE) da questão 2 e depois a execução da questão 1 para o programa p01.py original.

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/goulart/Documentos/INF100/PER/Praticas/p01.py =====
>>> q2()
Solução da questão 2
>>> q1()
Solução da questão 1
>>>
```

Portanto, você pode implementar e testar a solução das questões em qualquer ordem.

👉 importante: os comandos de cada questão deverão estar alinhados (mesma coluna) que o comando print existente no arquivo original. Veja nos dois exemplos abaixo, os erros de alinhamento (ou indentação do código):

Exemplo 1: comando ficou à esquerda do alinhamento correto.

```
File Edit Format Run Options Window Help
# Nome:
# Matrícula:
# Data:

def q1():
# Escreva aqui um programa solução da questão 1
#
# Escreva o código para a resolução da questão 1 abaixo
# Os comandos devem estar alinhados (mesma coluna) do comando print()
    print('Solução da questão 1')
    print()          # indentação correta
    print()          # indentação INCORRETA
    print()          # indentação correta
```

SyntaxError  
unindent does not match any outer indentation level  
OK

Exemplo 2: comando ficou à direita do alinhamento correto.

```
# Nome:
# Matrícula:
# Data:

def q1():
# Escreva aqui um programa solução da questão 1
#
# Escreva o código para a resolução da questão 1 abaixo
# Os comandos devem estar alinhados (mesma coluna) do comando print()
    print('Solução da questão 1')
    print()          # indentação correta
    print()          # indentação INCORRETA
    print()          # indentação correta
```

SyntaxError  
unexpected indent  
OK

☞ 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 (vale 10% da nota!!!). Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p01.py**) através do sistema do LBI.

### 1) Questão 1 - código de q1()

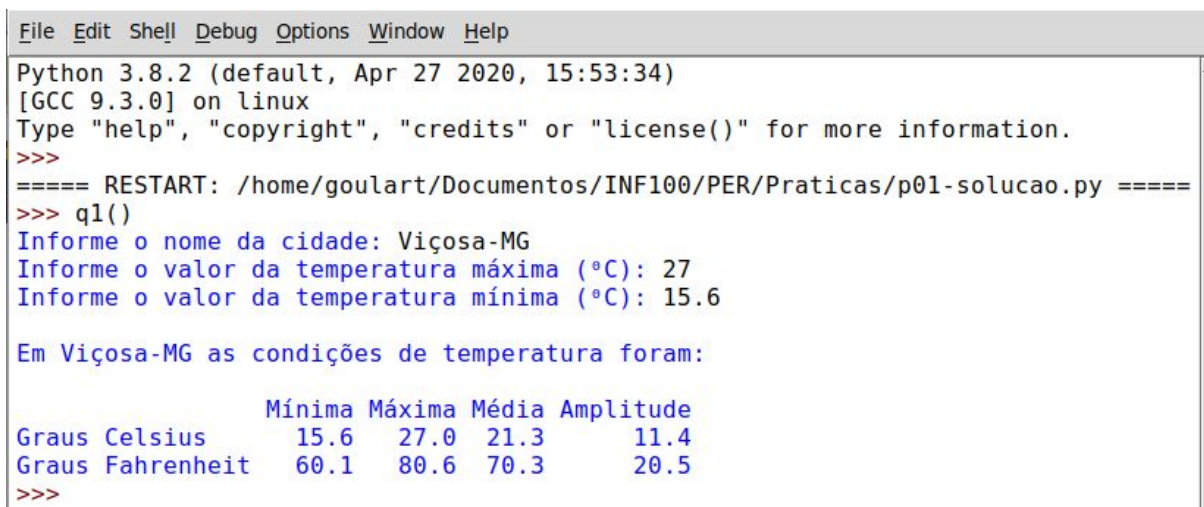
Escreva um programa para ler o nome de uma cidade e os valores da temperatura máxima e mínima de um mesmo dia, em graus Celsius. Após ler as informações, o programa deverá informar qual foi a amplitude térmica (diferença entre a temperatura máxima e mínima) e a temperatura média (considerando apenas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

A conversão calcule e informe o valor correspondente em graus Celsius, de acordo com 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.



```
File Edit Shell Debug Options Window Help
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/goulart/Documentos/INF100/PER/Praticas/p01-solucao.py =====
>>> q1()
Informe o nome da cidade: Viçosa-MG
Informe o valor da temperatura máxima (°C): 27
Informe o valor da temperatura mínima (°C): 15.6

Em Viçosa-MG as condições de temperatura foram:

          Mínima Máxima Média Amplitude
Graus Celsius      15.6   27.0   21.3      11.4
Graus Fahrenheit   60.1   80.6   70.3      20.5
>>>
```

Observe que o comando de impressão que estava no programa original foi removido (apagado ou comentado).

### 2) Questão 2 - código de q2()

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:

$$a_2 = a_1 + r$$

$$a_3 = a_2 + r = a_1 + r + r = a_1 + 2r$$

$$a_4 = a_3 + r = a_1 + 2r + r = a_1 + 3r$$



Um exemplo de PA com  $a_1 = 2$  e razão  $r = 5$ : 2, 7, 12, 17, 22, 27.

Um elemento na posição  $n$  pode ser calculado usando a expressão:

$$a_n = a_1 + (n - 1).r$$

E a soma dos  $n$  primeiros elementos de uma Progressão Aritmética é dada pela expressão:

$$S_n = \frac{n.(a_1 + a_n)}{2}$$

Escreva um programa para informar o valor do  $N$ -ésimo elemento de uma Progressão Aritmética (PA) e a soma dos  $N$  primeiros elementos desta mesma PA. 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: 0.5  
Informe o valor da razão da PA: 0.25  
Informe o valor de N: 6  
  
O 6º elemento da PA é = 1.750  
A soma dos 6 primeiros elementos é = 6.750  
>>>
```

Observe que o comando de impressão que estava no programa original também foi removido (apagado ou comentado).

## Entrega

Na mesma página do servidor do LBI, onde você buscou os arquivos, conforme instrução apresentada na tela.



Certifique-se de que o arquivo (**p01.py**) que você enviará é a versão mais recente do seu programa Python.

Até a próxima!!!