

## Introdução

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”.

Para que o texto 'kg' saia colado ao número, temos que informar ao **print()** para não usar nenhum separados entre os parâmetros. Mas nesse caso, devemos inserir um espaço em branco depois do texto 'Peso:' explicitamente, assim:

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

O resultado será o seguinte:

```
Peso: 75.57kg
```

Podemos obter o mesmo resultado usando um recurso muito interessante de formatação de números e textos disponível em Python:

```
peso = 75.57
print('Peso: %.2fkg' % peso )
```

Nesse caso, o '%.2f' indica que eu quero 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:

```
peso = 75.57
print('Peso: %.1fkg' % peso )
```

e o resultado será:

```
Peso: 75.6kg
```

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

```
peso = 75.57
print('Peso:%7.2fkg' % peso )
```

Nesse caso, o resultado será:

```
Peso: 75.57kg
```

Como o valor propriamente dito (75.57) ocupa 5 posições na tela, são adicionados dois espaços em branco para completar a largura de 7. Veja que nesse exemplo removemos o espaço em branco logo depois do ':', caso contrário teríamos três 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':

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

Nesse caso, o resultado será:

```
Tenho 18 anos.
```

Assim como o formatador '%f', com o '%d' podemos controlar também a largura:

```
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:

```
idade = 18
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.

## Roteiro de Prática

Nome do arquivo a ser entregue: **p02.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.

Faça o “download” do arquivo **p02.py**. Em seguida, entre o IDLE e abra o arquivo através do menu **File > Open....**

Segue abaixo o código que você deverá ver na tela:

```
# Nome do aluno:
# Matrícula:
# Data:
# (breve comentário dizendo o que o programa faz)

nome = 'Universidade Federal de Viçosa'
alunos = 11235          # alunos de graduação matriculados no campus Viçosa
rel_cand_vagas = 16.9   # relação candidatos / vagas total
area = 2353.94          # área física total em hectares (ha)
```

Logo abaixo da última linha do programa, acrescente comandos **print** de modo que a saída do programa siga exatamente o modelo mostrado abaixo:

## Prática 2 – INF100 – 2018/I – Valor: 1 ponto

```
Python 3.5.2 |Anaconda 4.1.1 (64-bit)| (default, Jul 5 2016, 11:41:13) [MSC v.1900
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: p02.py =====

Nome da instituição:                Universidade Federal de Viçosa
Alunos de grad. matriculados no campus Viçosa: 11235
Relação candidatos / vagas total:   16.9
Área total do campus Viçosa:        2353.94 ha

Área total do campus Viçosa (ha):
Usando largura 7 e 2 casas decimais:2353.94
Usando largura 8 e 2 casas decimais: 2353.94
Usando largura 9 e 2 casas decimais:  2353.94
Usando largura 9 e 3 casas decimais: 2353.940
Usando largura 9 e 4 casas decimais:2353.9400
Usando largura 6 e 1 casa decimal    :2353.9
Usando largura 6 e nenhuma casa decimal: 2354
>>>
```

Nesta prática **NÃO** será permitido colocar os dados diretamente dentro dos comandos **print**. Por exemplo, **NÃO** faça assim:

```
print('Alunos de grad. matriculados no campus Viçosa: 11235')
```

Ou seja, você deverá usar as variáveis criadas no início do programa para escrever seus dados da tela. Repare também a linha em branco existente antes da 1ª linha escrita na tela.

Depois de preencher o cabeçalho do programa, ele terá mais ou menos a seguinte “cara”:

```
# Nome do aluno:
# Matrícula:
# Data:
# Este programa escreve os dados do aluno de formas diferentes
# usando o comando print.

nome = 'Universidade Federal de Viçosa'
alunos = 11235          # alunos de graduação matriculados no campus Viçosa
rel_cand_vagas = 16.9   # relação candidatos / vagas total
area = 2353.94          # área física total em hectares (ha)
```

```
print()
```

Imprime uma  
linha em branco.

```
...

```

☞ Dentro dos slides da Aula 01 e do Guia Rápido de Python (páginas 2 e 3) existem diversos outros exemplos mostrando o uso do comando **print** com formatação.

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

☞ Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição do programa (pode usar a mesma descrição mostrada no quadro acima).

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p02.py**) através do sistema do LBI.