

## EXERCICIOS

1. Crie um programa que leia nome, sexo e idade de várias pessoas, guardando os dados de cada pessoa em um dicionário e todos os dicionários em uma lista. No final, mostre:
  - a) Quantas pessoas foram cadastradas
  - b) A média de idade
  - c) Uma lista com as mulheres
  - d) Uma lista de pessoas com idade acima da média
2. Escreva um programa Python para criar uma tupla com números de 1 a 4 e imprima um item
3. Escreva um programa Python para desempacotar uma tupla em várias variáveis: considere uma tupla = ('aluno', 'universidade', 'nota', 'resultado') e as variáveis w, x, y, z

Consulte o tipo de dados de cada uma das variáveis e imprima os valores de cada uma delas.

4. Crie 3 conjuntos conforme estrutura a seguir:

```
setx = set(["apple", "mango"])
sety = set(["mango", "orange"])
setz = set(["mango"])
```

Faça as seguintes operações sobre conjuntos:

- a) Faça a união dos três conjuntos e imprima o resultado
  - b) Verifique quais os elementos comuns do conjunto setx e sety e imprima o resultado
  - c) Verifique se o conjunto setx é subconjunto do conjunto sety e setz utilizando `issubset()`
  - d) Verifique quais elementos do conjunto setx não existem em sety
5. Escreva um programa Python para inserir um elemento no início de um determinado DicionárioOrdenado.

DicionárioOrdenado original:

```
DicionárioOrdenado [('color1', 'Red'), ('color2', 'Green'), ('color3', 'Blue')]
```

Insira um elemento no início do referido DicionárioOrdenado:

DicionárioOrdenado atualizado:

```
DicionárioOrdenado [('color4', 'Orange'), ('color1', 'Red'), ('color2', 'Green'), ('color3', 'Blue')]
```

6. Escreva uma função chamada `right_justify`, que receba uma string chamada `s` como parâmetro e exiba a string com espaços suficientes à frente para que a última letra da string esteja na coluna 70 da tela:

```
>>> right_justify('monty')
```

Dica: Use concatenação de strings e repetição. Além disso, o Python oferece uma função integrada chamada `len`, que apresenta o comprimento de uma string, então o valor de `len('monty')` é 5.

7. Um objeto de função é um valor que pode ser atribuído a uma variável ou passado como argumento. Por exemplo, `do_twice` é uma função que toma um objeto de função como argumento e o chama duas vezes:

```
def do_twice(f):  
    f()  
    f()
```

Aqui está um exemplo que usa `do_twice` para chamar uma função chamada `print_spam` duas vezes:

```
def print_spam():  
    print('spam')  
do_twice(print_spam)
```

- a) Digite este exemplo em um script e teste-o.
  - b) Altere `do_twice` para que receba dois argumentos, um objeto de função e um valor, e chame a função duas vezes, passando o valor como um argumento.
8. Crie um programa que faça a impressão de uma mensagem e a multiplicação de dois números. Utilize módulos e funções para resolução desse problema.
- a) O usuário deve entrar com a mensagem e com o uso de módulos e funções, essa mensagem deve ser impressa na tela
  - b) O usuário deve entrar com os valores dos dois multiplicadores e o programa deve exibir o resultado na tela.
9. Dada uma lista encadeada de caracteres formada por uma sequência alternada de letras e dígitos, construa um método que retorne uma lista na qual as letras são mantidas na sequência original e os dígitos são colocados na ordem inversa. Exemplos:

A 1 E 5 T 7 W 8 G → A E T W G 8 7 5 1

3 C 9 H 4 Q 6 → C H Q 6 4 9 3

Como mostram os exemplos, as letras devem ser mostradas primeiro, seguidas dos dígitos. Sugestões:

- a) usar uma fila e uma pilha;
- b) supor um método `ehDigito()` que retorna um valor booleano, como por exemplo, verdadeiro caso um caractere seja um dígito.

**EXERCICIOS RESOLVID**