

mentorama



modulo 2

01

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

Nesta questão você precisa solicitar que o usuário informe nome, sexo e idade de várias pessoas. Estes dados informados devem ficar guardados em um dicionário. Vai ser um dicionário para cada cadastro e todos os dicionários ficarão armazenados em uma única lista.

Feito isso, você vai calcular quantas pessoas foram cadastradas, a média das idades das pessoas. Você vai, também, criar uma nova lista contendo os dados de todas as mulheres e uma lista contendo os dados das pessoas que possuem idade acima da média.



Importante

Para fazer comparações, lembre-se de utilizar as condicionais (if, else) e, para percorrer todas as pessoas cadastradas utilize estrutura de repetição (for de preferência)



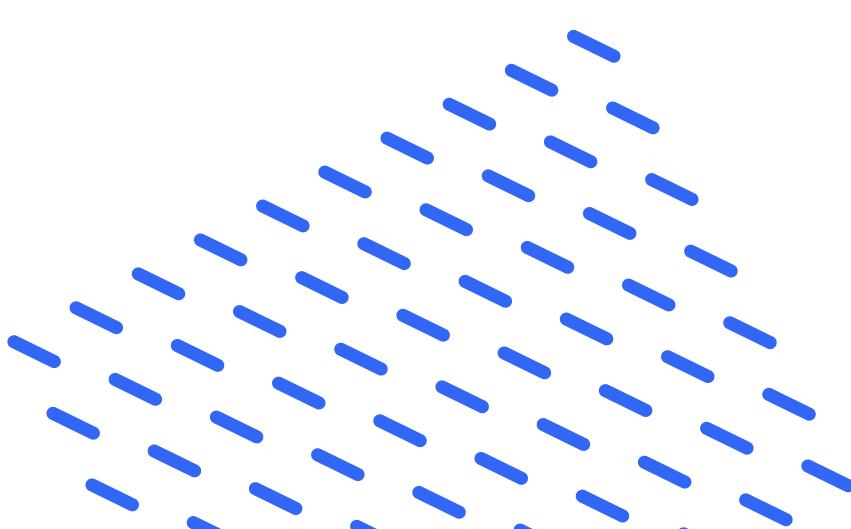
Dica

Primeiro solicite os dados e armazene-os na estrutura de dados solicitada no enunciado. Depois, realize as questões a, b, c e d. Pois, para todas elas, você precisará de todos os dados.



Dica

Vale lembrar que na lista cada elemento conterá um dicionário, sendo assim, o tamanho da lista corresponderá a quantidade de pessoas cadastradas



02

Escreva um programa Python para criar uma tupla com números de 1 a 4 e imprima um item.

Neste programa você vai inserir os números 1, 2, 3 e 4 em uma tupla. Feito isso, acesse um dos números da tupla e exiba na tela.



Importante

Para preencher a tupla você pode utilizar a estratégia que preferir como por exemplo: preencher usando estrutura de repetição, preencher manualmente, solicitar que o usuário digite os números, etc...

03

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



Importante

Seu programa deve funcionar para qualquer string, não apenas para "Mentorama"

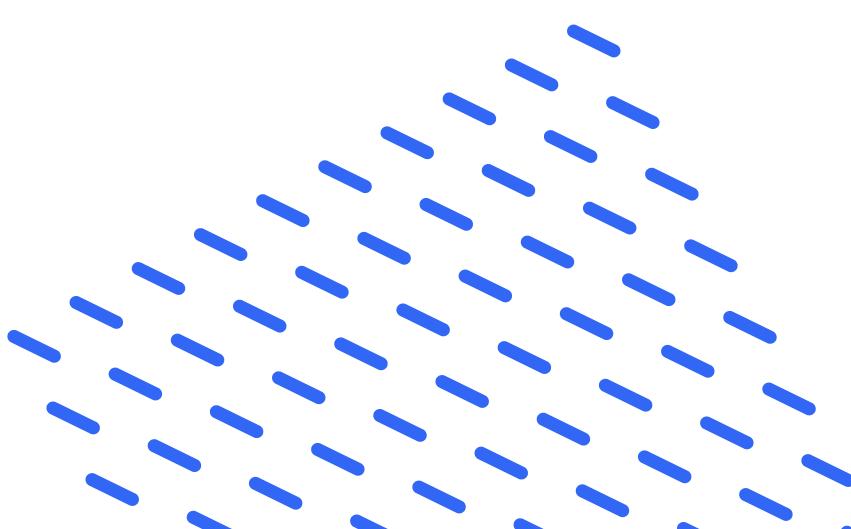


Dica

Utilize o comando type

Para esta questão você vai, primeiro criar a tupla:
tupla = ('aluno', 'universidade', 'nota', 'resultado').

Depois disso, você precisa implementar um programa que desempacote os elementos 'aluno', 'universidade', 'nota', 'resultado' para as variáveis w, x, y, z.



04

Crie 3 conjuntos conforme estrutura a seguir:

- a) `setx = set(["apple", "mango"])`
- b) `sety = set(["mango", "orange"])`
- c) `setz = set(["mango"])`

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



Dica

A partir dos conjuntos criados utilize as operações de conjuntos que já existem no python, que são: união, interseção, subconjunto e diferença

05

Escreva um programa Python para inserir um elemento no início de um determinado DicionárioOrdenado.

DicionárioOrdenado original:

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



Dica

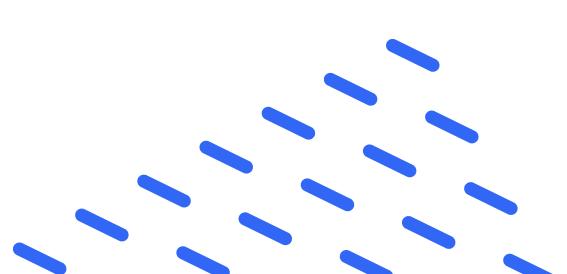
Se preferir, utilize a função `dic.insert` do Python

Insira um elemento no início do referido DicionárioOrdenado:

DicionárioOrdenado atualizado:

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

Você vai criar um dicionário conforme primeiro exemplo. Depois, você vai inserir o elemento 'Orange' cujo índice é o 'color4' ('color4', 'Orange') no início deste dicionário.



06

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')
```

Você precisa criar um programa que dada uma string qualquer, ele retorne esta string com espaços suficientes na frente para que a última letra da string fique na posição 70. Vamos supor que a string é "MENTORAMA". Observe que esta string tem tamanho 9. Então, o programa precisa retornar esta string com tamanho 70. Para isso, você vai adicionar 61 espaços em branco na frente da palavra MENTORAMA para que a última letra da string fique na posição 70.



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



Importante

Vale lembrar que você precisa criar uma lógica para o programa definir quantos espaços colocar na frente da palavra, não vale adicioná-los manualmente, ok?

07

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)
```



Dica

Se preferir, utilize a função `dic.insert` do Python

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.

Na letra a) você vai apenas testar o código do enunciado.

Na letra b) você vai alterar o do_twice para que ele receba dois argumentos/parâmentros (um objeto função e um valor) e chamar a função duas vezes



Dica

A função do_twice chama o objeto de função "f" duas vezes:

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

Seguindo este raciocínio, suponha que você quer utilizar a função do_twice para chamar, duas vezes, a função print_spam. Para isso, fazemos:

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

```
#chama a função do_twice e envia como parâmetro o objeto de função print_spam  
do_twice(print_spam)
```

Observe que até o momento enviamos como parâmetro apenas o objeto de função. Agora, suponha que queremos enviar como parâmetro o objeto de função e, também, um valor. Este valor, deverá ser exibido duas vezes. Para isso, faça:

```
#função do_twice recebe f (objeto de função) e g (um valor qualquer)
```

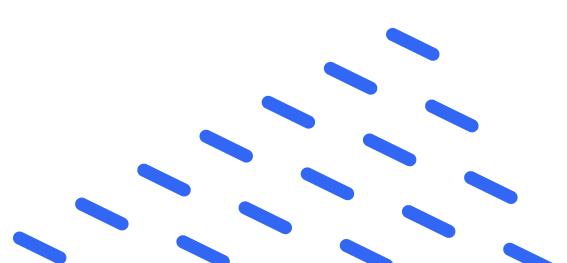
```
def do_twice(f,g):  
    f(g)  
    f(g)
```

```
#esta é a função que será chamada pelo do_twice. Ela recebe o valor g como  
parâmetro para que ele seja exibido
```

```
def print_valor(g):  
    print(g)
```

```
#chama a função do_twice e, como parâmetro envia a função objeto e um valor qualquer.
```

```
do_twice(print_valor, "Olá, Milena")
```



08

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



Importante

Os módulos ficam em arquivos separados do código principal, para utilizar as funções de um módulo você precisa importá-los no código principal.

09

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:

A1E5T7W8G → A E T W G 8 7 5
3C9H4Q6 → C H Q 6 4 9 3

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



Dica

Mostre para seu programa o que é um dígito por meio da implementação de uma lista: listaDigitos=[0,1,2,3,4,5,6,7,8,9]. Verifique se cada elemento da sequência é um dígito ou letra analisando se ele está na lista de dígitos ou não.

Bom, vamos lá... vou te explicar esta questão.

Você tem uma lista contendo uma sequência de letras e dígitos, correto? Então, agora você precisa criar um programa que retorne os valores desta sequencia da seguinte maneira:

- As letras vão ser mantidas na sequencia original, porém ficarão no início da lista;
- Os dígitos (números) ficarão na sequencia inversa e no final da lista.

Como mostra o exemplo do enunciado.



end of modulo 2