

Matrizes

5 questions

1
point

1.

Assinale as alternativas CORRETAS:

- ☐ Uma matriz denominada A de n linhas e m colunas, terá o último elemento na linha n-1 e coluna m-1 e para acessar este elemento utiliza-se o seguinte comando: A[n-1][m-1]
- ☐ Uma matriz denominada A de n linhas e m colunas, terá o último elemento na linha n-1 e coluna m-1 e para acessar este elemento utiliza-se o seguinte comando: A[m-1][n-1]
- ☐ O primeiro elemento da matriz A se encontra na linha 1 e coluna 1 e para acessar este elemento utiliza-se o seguinte comando: A[1][1]
- ☐ O primeiro elemento da matriz A encontra-se na linha 0 (zero) e coluna 0 (zero) e para acessar este elemento utiliza-se o seguinte comando: A[0][0]
- ☐ Matrizes, em Python, são listas de listas

1
point

2.

Observe o código abaixo:

```
1 def cria_matriz(tot_lin, tot_col, valor):
2     matriz = [] #lista vazia
3     for i in range(tot_lin):
4         linha = []
5         for j in range(tot_col):
6             linha.append(valor)
7         matriz.append(linha)
8     return matriz
```

Assinale a alternativa que representa a matriz que será armazenada em x se for digitado o seguinte comando: x = cria_matriz(2, 3, 99)

- ☐ Nenhuma das alternativas representa a matriz resultante
- ☐ [[], []]
- ☐ []
- ☐ [[99, 99, 99, 99], [99, 99, 99, 99], [99, 99, 99, 99]]
- ☐ [[99, 99, 99], [99, 99, 99]]
- ☐ [[]]

☐ [[99, 99, 99]]

1
point

3.

Observe o código abaixo:

```
1 def cria_matriz(tot_lin, tot_col, valor):
2     matriz = [] #lista vazia
3     for i in range(tot_lin):
4         linha = []
5         for j in range(tot_col):
6             linha.append(valor)
7         matriz.append(linha)
8     return matriz
```

Assinale a alternativa que representa a matriz que será armazenada em x se for digitado o seguinte comando: x = cria_matriz(2,3,99)

- ☐ [[99, 99, 99, 99], [99, 99, 99, 99], [99, 99, 99, 99]]
- ☐ Nenhuma das alternativas representa a matriz resultante
- ☐ [[]]
- ☐ [[], []]
- ☐ [[99, 99, 99], [99, 99, 99]]
- ☐ []
- ☐ [[99, 99, 99]]
-

1
point

4.

Além de alguns comandos novos, vamos ver como anda seu conhecimento de alguns comandos apresentados em vídeos anteriores (relembrando). Assinale as afirmações CORRETAS:

- ☐ return é o comando usado toda vez que se trabalha com listas dentro de uma função
- ☐ O comando append serve para acrescentar um novo dado no final de uma lista. Exemplo: suponha que A é uma lista. Para acrescentar o valor 44 no final desta lista utiliza-se o seguinte comando: A.append(44)
- ☐ O comando "in" serve para verificar se um determinado elemento se encontra em uma lista. Exemplo: 44 in pares, este comando irá verificar se o valor 44 está na lista pares
- ☐ Os comentários dentro do código são precedidos do símbolo %
- ☐ Para criar uma lista vazia é só atribuir [] para a variável lista. Exemplo: suponha que A é a variável que será uma lista, para criar esta lista vazia utiliza-se o comando A = []
- ☐ O comando range cria um objeto que vai do número inicial ao número final informado dentro dos parênteses, porém, quando se informa somente um número nos parênteses, a contagem começa com 0 e vai até o valor imediatamente anterior ao informado. Por

exemplo: range (20) cria um objeto que varia de 0 até 19 e range (5, 10) varia de 5 até 9

- ☐ Ao criar uma função, pode-se criar uma documentação que aparecerá ao usuário quando ele quiser chamar esta função, orientando-o na forma de construir a chamada. Para isto, utiliza-se três pontinhos no início e no final do texto da documentação, desta forma: ... aqui vai o texto da documentação ...

1
point

5.

O professor, no vídeo, mostrou como acrescentar em uma matriz valores informados pelo usuário, conforme código abaixo:

```
1 def cria_matriz(num_linhas, num_colunas):
2     matriz = [] #lista vazia
3     for i in range(num_linhas):
4         linha = []
5         for j in range(num_colunas):
6             valor = int(input("Digite o elemento [" + str(i) + "][" + str(j) +
7                             "]: "))
8             linha.append(valor)
9         matriz.append(linha)
10    return matriz
```

Este código faz com que primeiramente toda a primeira linha seja preenchida, em seguida a segunda e assim sucessivamente. Se nós quiséssemos que a primeira coluna fosse preenchida e em seguida a segunda coluna e assim por diante, como ficaria o código?

Um exemplo: se o usuário digitasse o seguinte comando "x = cria_matriz(2,3)" e em seguida informasse os seis números para serem armazenados na matriz, na seguinte ordem: 1, 2, 3, 4, 5, 6; o x teria ao final da função a seguinte matriz: [[1, 3, 5], [2, 4, 6]].

- ☐ Nenhuma das outras alternativas está correta.
- ☐ O código para realizar a tarefa solicitada é:


```
1 def cria_matriz(num_linhas, num_colunas):
2     matriz = [] #lista vazia
3     for i in range(num_colunas):
4         linha = []
5         for j in range(num_linhas):
6             valor = int(input("Digite o elemento [" + str(j) + "][" +
7                             str(i) + "]: "))
8             linha.append(valor)
9         matriz.append(linha)
10    return matriz
```

- ☐ O código para realizar a tarefa solicitada é:

```
1 def cria_matriz(num_linhas, num_colunas):
2     matriz = [] #lista vazia
3     for i in range(num_colunas):
4         coluna = []
5         for j in range(num_linhas):
6             valor = int(input("Digite o elemento [" + str(j) + "][" +
7                             str(i) + "]: "))
8             coluna.append(valor)
9         matriz.append(coluna)
10    return matriz
```

- ☐ O código para realizar a tarefa solicitada é:

```
1 def cria_matriz(num_linhas, num_colunas):
2     matriz = [] #lista vazia
3     for i in range(num_linhas):
4         linha = []
5         for j in range(num_colunas):
6             linha.append(0)
7         matriz.append(linha)
8
9     for i in range(num_colunas):
10        for j in range(num_linhas):
11            matriz[j][i] = int(input("Digite o elemento [" + str(j) +
12                                     "]" + str(i) + "]: "))
13
14    return matriz
```

 O código para realizar a tarefa solicitada é:

```
1 def cria_matriz(num_linhas, num_colunas):
2     matriz = [] #lista vazia
3     for i in range(num_linhas):
4         coluna = []
5         for j in range(num_colunas):
6             coluna.append(0)
7         matriz.append(coluna)
8
9     for i in range(num_linhas):
10        for j in range(num_colunas):
11            matriz[j][i] = int(input("Digite o elemento [" + str(i) +
12                                     "]" + str(j) + "]: "))
13
14    return matriz
```

☐ I understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera's Honor Code

Enter your legal name

3 questions unanswered

Submit Quiz

