

{introcomp}

Working 05 : Matrizes, Tuplas e Dicionários

Objetivos:

- Entender como criar e manipular Matriz, Tupla e Dicionário;
- Aplicar essas estruturas em problemas de programação;

Prazo de Envio: sábado, 23/09, 04:00.

1 INTRODUÇÃO

Vimos que, conforme um programa acumula variáveis, se torna complexo para o programador manter a organização do programa, dificultando sua manutenção, documentação e expansão. Devido à necessidade, temos formas de organizar e trabalhar com esses dados, algumas delas são as Matrizes, Tuplas e Dicionários. As Matrizes são uma forma de organizar dados em formato de tabela, permitindo que o programador facilmente organize formulários, separe os dados por tipo, aplique algoritmos algébricos, etc. As Tuplas são listas imutáveis que armazenam diferentes tipos de dados, sendo ideal para aplicações onde se deseja manter os valores armazenados intactos. Os Dicionários são outra forma de armazenar dados, com o foco na categorização desses dados, pois abstraem a forma de procurar dados no programa por meio das chaves, que permitem um tratamento mais intuitivo das informações.

2 MATRIZ

2.1 Definição:

Segue a definição de matriz:

Matrizes são organizações de informações numéricas em uma tabela retangular formada por linhas e colunas. Essa organização em uma tabela facilita que se possa efetuar vários cálculos simultâneos com as informações contidas na matriz.^[1]

Em Python uma matriz pode ser declarada como uma lista de listas.

2.2 Declaração:

Queremos declarar a seguinte matriz em Python:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Perceba que a matriz possui 3 colunas e 2 linhas. Nesse caso deveremos criar uma lista contendo duas listas, uma para cada linha, e cada uma dessas duas listas deverá conter 3 itens.

```
1 # Declarando a matriz:
  Matriz = [[1, 2, 3],
3           [4, 5, 6]]
```

2.3 Acesso:

Agora que criamos nossa matriz precisamos acessá-la. A tabela abaixo nos mostra a saída gerada dado o comando.

Comando	Saída	Explicação
Matriz	[[1, 2, 3], [4, 5, 6]]	Ao chamar Matriz é retornada a lista "superior".
Matriz[0]	[1, 2, 3]	Ao chamar Matriz[0] é retornada a lista no índice 0 da lista "superior".
Matriz[0][0]	1	Ao chamar Matriz[0][0] é retornado o valor no índice 0 da lista no índice 0 da lista "superior".

De maneira genérica:

Comando	Saída
oNomeDaMatriz	Ao chamar oNomeDaMatriz ela é retornada por completo.
oNomeDaMatriz[i]	Ao chamar oNomeDaMatriz[i] é retornada a lista no índice i da lista "superior".
oNomeDaMatriz[i][j]	Ao chamar oNomeDaMatriz[i][j] é retornado o valor no índice j da lista no índice i da lista "superior".

3 TUPLA

3.1 Definição:

Uma tupla é como uma lista heterogênea (suporta elementos de tipos diferentes em uma mesma estrutura) com elementos imutáveis.

3.2 Declaração:

Queremos declarar a seguinte tupla em Python:

```
[1 "Dois" 5.00]
```

A declaração da tupla é feita de forma semelhante a uma lista, entretanto, seu caractere delimitador não é um '[' e sim '(' e podemos declarar elementos de vários tipos diferentes. Desse modo declaramos:

```
1 # Declarando a tupla:
  Tupla = (1, "Dois", 5.00)
3
```

ATENÇÃO: OS ELEMENTOS DE UMA TUPLA NÃO PODEM SER ALTERADOS APÓS A DECLARAÇÃO.

4 DICIONÁRIO

4.1 Definição:

A estrutura dicionário em Python é uma implementação específica de uma estrutura amplamente utilizada conhecida como tabela de dispersão, ou tabela hash. Tal estrutura de dados possui uma estrutura um pouco mais complexa do que as vistas anteriormente. Para exemplificar iremos abstrair detalhes do funcionamento das tabelas hash. A principal diferença entre uma lista e um dicionário é que o dicionário possui dois espaços por índice. Uê, mas como assim? É simples, o dicionário possui um espaço de chave e outro de valor. A chave funciona como os índices entre colchetes de uma lista (0, 1, 2, ..., i), entretanto pode ser definida pelo próprio programador, ou seja, você pode definir chaves como "João", "Gabriel", "Elias", "Marco", "Isabel" e muitos outros. Isso pode ser muito útil para a implementação de uma lista de notas de alunos por exemplo, você pode buscar a nota do aluno pelo nome, basta definir a chave como o nome do aluno e o valor como sua nota. Exemplo, João tirou 9.8 em sua prova. No dicionário de notas dessa prova a chave "João" possui o valor '9.8'. Esse dicionário teria a seguinte aparência:

$$\left[\begin{array}{l} \text{"Gabriel"} : 9.7 \\ \text{"Joao"} : 9.5 \\ \text{"Felipe"} : 9.1 \\ \text{"Julia"} : 10 \\ \text{"Andre"} : 9.5 \\ \text{"Maria"} : 9.8 \end{array} \right]$$

4.2 Declaração:

Queremos declarar o seguinte dicionário em Python:

$$\left[\begin{array}{l} \text{"Um"} : 1 \\ \text{"Dois"} : 2 \end{array} \right]$$

A declaração da tupla é feita de forma semelhante a uma lista, entretanto, seu caractere delimitador não é um '['' e sim '"' e para cada elemento devemos declarar uma chave e um valor.

Desse modo declaramos:

```
# Declarando o dicionario:
Dict = {'Um': 1, 'Dois': 2}
```

4.3 Acesso:

O acesso é feito pelo valor da chave, exemplo:

Comando	Saída
Dict['Um']	1
Dict['Dois']	2

De maneira genérica:

Comando	Saída
Dict[chave]	valor

ATENÇÃO: AS CHAVES E VALORES PODEM SER DE TIPOS DIFERENTES, MAS UMA VEZ ESCOLHIDO O TIPO DA CHAVE/VALOR, ELE DEVE SER MANTIDO EM TODOS OS ÍNDICES.

5 TABELA DE FUNÇÕES ÚTEIS

Nome	Utilidade
len(suaEstrutura)	Retorna o número de elementos na estrutura.
del seuDicionario[chave]	Deleta o índice que contém aquela chave do seu dicionário
type(suaEstrutura)	Retorna uma string informando o tipo da sua estrutura (dict, list, tuple, etc...), ou, caso seja um elemento, informa o tipo do elemento.
seuDicionario.keys()	Retorna uma tupla contendo as chaves do seu dicionário.
append(), pop(), extend(), insert(), remove(), etc...	Realizam modificações em estruturas de dados, vale a pena pesquisar sobre.

Praticando

1. Faça um programa que dado 9 números, os coloque em uma matriz 3x3 e imprima-os no formato do exemplo abaixo.

Exemplo de Entrada	Exemplo de Saida
1 2 3 4 5 6 7 8 9	[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
10 20 30 40 50 40 30 20 10	[[10, 20, 30], [40, 50, 40], [30, 20, 10]]

2. Faça um programa em que dadas as dimensões crie uma matriz nula (cheia de zeros) nas dimensões informadas e a imprima como no problema anterior.

Exemplo de Entrada	Exemplo de Saida
3 2	[[0, 0, 0], [0, 0, 0]]
1 5	[[0], [0], [0], [0], [0]]
2 4	[[0, 0], [0, 0], [0, 0], [0, 0]]

3. Faça um programa que leia nomes de alunos e suas respectivas notas até que o nome 'oooo' seja informado, após o fim da leitura, imprima o nome do aluno que possui a maior nota. Obs.: Use dicionário para resolver essa questão.

Exemplo de Entrada	Exemplo de Saida
Zezinho 0.1 oooo	Zezinho
Huguinho 7 Maria 8.2 Manoel 4.7 Dilma 6.5 oooo	Maria

4. Faça um programa em que é passado um número de alunos e em seguida um aluno por linha, onde cada linha contém nome(string), idade(int) e CRN(float). Após isso imprima uma lista contendo idade(int), nome(string) e CRN(float). (Dica: Use tuplas para um melhor aprendizado) OBSERVAÇÃO: Pode ser que na entrada existam número em (float) que devem ser interpretados como (int) e vice-versa.

Exemplo de Entrada	Exemplo de Saida
3	18 Joao 7.0
Joao 18 7.0	18 Maria 7.0
Maria 18 7.0	30 LoboMau 2.0
LoboMau 30 2.0	

DESAFIOS

1. Crie um programa que receba do teclado 18 valores inteiros e os coloque em duas matrizes 3x3, faça a multiplicação dessas matrizes e imprima no formato do exemplo abaixo.

Exemplo de Entrada	Exemplo de Saida
1 2 3 4 5 6 7 8 9 10 20 30 40 50 40 30 20 10	[[180, 180, 140], [420, 450, 380], [660, 720, 620]]

Referências Bibliográficas

Referências

- [1] Matemática Básica *Matrizes: Definições e Operações.*
<https://matematicabasica.net/matrizes/>