LABORATÓRIOS DE PROGRAMAÇÃO

Estruturas de Dados

Aula 5

Profª. Kamilla Dória

TÓPICOS

- Tuplas
- Dicionários
- Matriz

TUPLAS

ESTRUTURA DE DADOS

- Estruturas nativas do Python as quais podem ser utilizadas em importar bibliotecas:
  - Tupla
  - Dicionário
  - Matriz

TUPLA

- É semelhante à lista, mas com uma diferença: seus elementos não podem ser mudados;
    - É um elemento imutável;
    - Não é possível adicionar novos elementos, remover ou substituir elementos existentes;
        - Caso um dos elementos da tupla seja uma lista, a lista não pode ser removida nem substituída, mas seus elementos podem sofrer alterações.

TUPLA

- Pode ter qualquer quantidade de valores e qualquer tipo de valor;
- É escrita sempre entre parênteses:

```
x = (1, 2, 3)
```

- Os valores não podem ser alterados mas a variável que contém a tupla pode receber uma nova atribuição:

```
x = (1, 2, 3, 4)
```

TUPLA

- Exemplos:
  - Representar a temperatura em Celsius e Farenheint:

    temp1 = (32, ‘C’)
    temp2 = (45, ‘F’)

  - Representar coordenadas:

    coord = (-7374498, 9887767)

TUPLA

- O acesso aos itens de uma tupla é feito da mesma forma que o acesso a itens de uma lista, ou seja, por meio de índices e colchetes:

```
temp1 = (32, 'C')
temp2 = (45, 'F')

#verificar se a temperatura é Celsius
if temp1[1] == 'C':
    print(f'Temperatura em Celsius: {temp1[0]}')
else:
    print(f'Temperatura em Celsius: {temp2[0]}')
```

EXERCÍCIO 1

Crie um programa modularizado com uma função que recebe duas temperaturas, em Celsius ou Farenheint, e informa qual a mais alta. Caso as temperaturas sejam de escalas diferentes, a função deve fazer uma conversão para verificar qual a temperatura mais alta.

Fórmulas:

$$(0\ °C \times 9/5) + 32 = 32\ °F$$

$$(32\ °F - 32) \times 5/9 = 0\ °C$$

DICIONÁRIO

DICIONÁRIO

- É um tipo de mapeamento nativo do Python;
- Um mapa é uma coleção associativa desordenada;
- A associação, ou mapeamento, é feita a partir de uma **chave**, que pode ser qualquer tipo imutável, para um **valor**, que pode ser qualquer objeto de dados do Python.

DICIONÁRIO

- Exemplo:
  - Criar um dicionário para traduzir palavras em Inglês para Espanhol.
  - Para este dicionário, as chaves são strings.

Chave	Valor
"one"	"uno"
"two"	"dos"
"three"	"tres"

DICIONÁRIO

- Uma maneira de criar um dicionário é começar com o dicionário vazio e adicionar pares chave-valor.
  - O dicionário vazio é denotado por {}
  - Cada elemento é acessado e atribuído por meio de []

```
dic = {}
dic['one'] = 'uno'
dic['two'] = 'dos'
dic['three'] = 'tres'
print(dic)
```

```
"Dicionario.py"
{'one': 'uno', 'two': 'dos', 'three': 'tres'}

Process finished with exit code 0
```

DICIONÁRIO

- Outra forma de criar o dicionário:

```
dic = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
print(dic)
```

```
"Dicionario.py"
{'one': 'uno', 'two': 'dos', 'three': 'tres'}

Process finished with exit code 0
```

OPERAÇÕES COM DICIONÁRIOS

- A exclusão de elementos é feita por meio do comando *del*:

```
dic = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
print(dic)
del dic['two']
print(dic)
```

"Dicionario.py"
{'one': 'uno', 'two': 'dos', 'three': 'tres'}
{'one': 'uno', 'three': 'tres'}

OPERAÇÕES COM DICIONÁRIOS

- A alteração de elementos é feita por meio do comando de atribuição:

```
dic = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
print(dic)
dic['two'] = 'dos dos'
print(dic)
```

"Dicionario.py"
{'one': 'uno', 'two': 'dos', 'three': 'tres'}
{'one': 'uno', 'two': 'dos dos', 'three': 'tres'}

Process finished with exit code 0

OPERAÇÕES COM DICIONÁRIOS

- A verificação da quantidade de elementos é feita por meio do comando *len*:

```
dic = {'one': 1, 'two': 2, 'three': 3}
print(len(dic))
```

```
"Dicionario.py"
{'one': 'uno', 'two': 'dos dos', 'three': 'tres'}
3

Process finished with exit code 0
```

OPERAÇÕES COM DICIONÁRIOS

- Outros métodos:

keys() : Retorna as chaves do dicionário

values() : Retorna os valores do dicionário

items() : Retorna os pares chave-valor do dicionário

get(índice) : Retorna o valor associado com a chave; ou None

OPERAÇÕES COM DICIONÁRIOS

- Exemplo:

```
dic = {'one': 1, 'two': 2, 'three': 3}
#transformando a lista de chaves em lista
for key in dic.keys():
   print('Elemento: ', dic[key])
#outra forma de fazer
ks = list(dic.keys())
print(ks)
```

```
"Dicionario.py"
Elemento:  1
Elemento:  2
Elemento:  3
['one', 'two', 'three']

Process finished with exit code 0
```

OPERAÇÕES COM DICIONÁRIOS

- Os operadores *in* e *not* podem ser utilizados para saber se uma chave existe ou não no dicionário:

```
dic = {'one': 1, 'two': 2, 'three': 3}
if 'one' in dic:
    print('Existe!')
else:
    print('Não existe!')
```

```
"Dicionario.py"
Existe!

Process finished with exit code 0
```

OPERAÇÕES COM DICIONÁRIOS

- Outra forma de fazer:

```
dic = {'one': 1, 'two': 2, 'three': 3}
if dic.get('one') == 1:
    print('Existe!')
else:
    print('Não existe!')
```

```
"Dicionario.py"
Existe!

Process finished with exit code 0
```

EXERCÍCIO 2

Utilizando o conceito de modularização, crie um programa com um dicionário que mostre a probabilidade de detectar certas partículas subatômicas.

Chave: nomes das partículas | Valor: probabilidades

Chave: nêutron | Valor: 0.55

O programa deverá ter os inputs para criar o dicionário e deverá ter uma função que receba um dicionário como entrada e que retorne uma lista contendo o nome da partícula menos provável de ser observada. Em caso de empate, a lista deve conter todos os empatados.

Trate o código para que não permita inserir valores de tipo incorreto no dicionário.

Trate também a possibilidade de a soma de todas as probabilidades ultrapassarem 1.

MATRIZ

MATRIZES

- São estruturas bidimensionais (tabelas) com m linhas por n colunas.

- Python não possui o elemento matriz mas é possível simular uma colocando lista dentro de lista;

  - Exemplo: [[2, 3], [4, 5]]

$$\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$$

MATRIZES

- O acesso aos elementos das matrizes é semelhante ao das listas: deve-se indicar os índices por meio de colchetes;

- Exemplo: matriz[[2, 3], [4, 5]]

  matriz[0] -> [2, 3]
  matriz[0][1] -> 3

- Assume-se que o primeiro índice é a linha e o segundo é a coluna.

MATRIZES

- Da mesma forma valores são alterados:
  - Exemplo: matriz[[2, 3], [4, 5]]

matriz[0][1] = 7

Matriz -> [[2, 7], [4, 5]]

MATRIZ

- Para criar uma matriz dinâmica, utiliza-se dois laços:

```python
#matriz 20x20
DIMENSAO = 20
matriz = []
for i in range(DIMENSAO):
    linha = []
    for j in range(DIMENSAO):
        linha.append(0)
    matriz.append(linha)

#imprimir as linhas
for i in range(DIMENSAO):
    print(matriz[i])
```

```
"Matriz.py"
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Process finished with exit code 0
```

MATRIZ

- Exemplo: marcar como verdadeira uma posição aleatória em uma matriz 5x5 onde todas as posições estão como falsa.

EXERCÍCIO 3

Utilizando os conceitos de modularização, crie um sistema simples de reserva de assentos para um teatro, o qual possui 25 filas com 40 cadeiras cada uma.

Quando o programa é iniciado todas as cadeiras ficam desocupadas. À medida que o usuário vai solicitando as reservas, as posições devem ser marcadas para não serem mais utilizadas por outras pessoas. Dessa forma, se alguém tentar reservar uma cadeira já reservada, o sistema não deve permitir.

OBRIGADA!