

## Aula 6

### Lógica de Programação e Algoritmos

Prof. Vinicius Pozzobon Borin

1

### Conversa Inicial

2

- O objetivo desta aula é aprendermos a manipular estruturas de dados em Python

3

- Estruturas de dados
  - Tuplas: ()
  - Listas: []
  - Dicionários: {}
- Aprenderemos também:
  - Conceito de método
  - Métodos para *strings*

4

### Tuplas

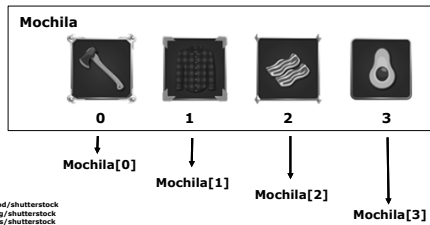
5

### Variáveis

- Simples: armazenam somente um dado
- Compostas: armazenam um conjunto de dados

6

### Exemplo lúdico



### Estrutura de dados

- É um conjunto de dados organizados de uma maneira específica na memória do programa
- A maneira como os dados estão organizados na memória, como podem ser buscados, manipulados e acessados, é o que define e diferencia as estruturas de dados

### Tupla

- Estrutura de dados estática
- A tupla é imutável
- Representada em Python por parênteses ()

- Vejamos tuplas no Python

### Desempacotamento de parâmetros em funções

- Suponha que você quer realizar o somatório de diversos valores, porém, não sabe quantos valores serão somados. Pode ser que sejam somente 2, ou então 10, ou mesmo 100 números
- Como criar uma função capaz de receber um número tão variável de parâmetros?

- Vejamos o exemplo no Python

## Listas

## Listas

- Estrutura de dados dinâmica
- Podemos alterar dados e tamanho
- Indexadas por valores numéricos inteiros
- Representadas em Python por colchetes []

- Vejamos listas no Python

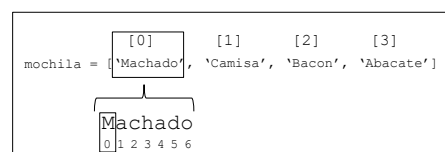
## O que são métodos?

- Uma lista é um objeto de uma classe dentro do Python
- Paradigmas de programação orientada a objetos (POO)
- Método é equivalente à função
  - `mochila.append('Ovos')`
  - `variável.função(parâmetro)`

## Strings e listas dentro de listas

## Strings dentro de listas

- `Mochila[0][0] → 'M'`



### Dupla indexação

- O primeiro índice é referente a cada item da lista
- O segundo índice é referente a cada caractere da *string*
- Assim, podemos acessar não só cada dado dentro da lista, mas também cada caractere das *strings* de um índice da lista

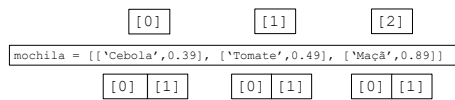
19

- Vejamos no Python

20

### Listas dentro de listas

- `Mochila[0][0] → 'Cebola'`
- `Mochila[2][1] → 0.89`



21

- Imagine uma situação na qual você deve realizar o cadastro de uma lista de compras em um sistema. Cada produto comprado deverá ser registrado com seu nome, quantidade e valor unitário
- Vejamos no Python

22

### Dicionários

### Dicionários

- Estrutura de dados dinâmica
- Podemos alterar dados e tamanho
- Indexados por chaves (palavras-chave)
- Representados em Python por chaves `{}`

23

24

- ▀ **Vejamos um dicionário no Python**

25

### **Métodos para dicionários**

- ▀ ***values***: obtém somente os dados
- ▀ ***keys***: obtém somente as chaves
- ▀ ***items***: obtém o par chave:dado

26

- ▀ **Vamos praticar no Python**

27

### **Listas com dicionários**

- ▀ **Uma lista contendo, em cada índice, um dicionário**
- ▀ **Vejamos em Python**

28

### **Dicionários com listas**

- ▀ **Um dicionário contendo, em cada índice, uma lista**
- ▀ **Vejamos em Python**

29

### **Trabalhando com métodos em *strings***

30

- Manipular *strings* é um assunto bastante vasto
- Nesta aula, você aprendeu os conceitos de lista e de método, o que nos permite estudar mais a fundo as *strings*

31

- Uma *string* é imutável
- Mas, com listas, podemos alterá-la
- Vejamos em Python

32

#### Relação de métodos para uso com *strings*

Função/método	Objetivo
<code>startswith</code>	Verifica se caracteres existem no início da <i>string</i>
<code>endswith</code>	Verifica se caracteres existem no final da <i>string</i>
<code>lower</code>	Converte <i>string</i> para minúscula
<code>upper</code>	Converte <i>string</i> para maiúscula
<code>find</code>	Busca a primeira ocorrência de um padrão de caracteres em uma <i>string</i>
<code>rfind</code>	Idêntico ao <code>find</code> , mas inicia a busca da direita para a esquerda
<code>center</code>	Centraliza uma <i>string</i>
<code>ljust, rjust</code>	Ajustam uma <i>string</i> com alinhamentos à esquerda ou à direita, respectivamente
<code>split</code>	Divide uma <i>string</i>
<code>replace</code>	Substitui caracteres em uma <i>string</i>
<code>lstrip, rstrip</code>	Removem espaços em branco à esquerda ou à direita, respectivamente
<code>strip</code>	Remove espaços em branco das extremidades

33

- Vejamos em Python alguns exemplos

34

#### Relação de métodos para validação de dados em *strings*

Função/método	Retorna True para uma <i>string</i> com...
<code>isalnum</code>	Somente letras e números; acentos são aceitos
<code>isalpha</code>	Somente letras; acentos são aceitos
<code>isdigit</code>	Somente números
<code>isnumeric</code>	Somente números; aceita também caracteres matemáticos, como frações
<code>isupper</code>	Somente caracteres maiúsculos
<code>islower</code>	Somente caracteres minúsculos
<code>isspace</code>	Somente espaços. Inclui TAB, quebra de linha, retorno etc.
<code>isprintable</code>	Somente caracteres possíveis de serem impressos na tela

35

- Vejamos em Python alguns exemplos

36