

## Aula 6

### Lógica de Programação e Algoritmos

Prof. Vinicius Pozzobon Borin

### Conversa Inicial

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

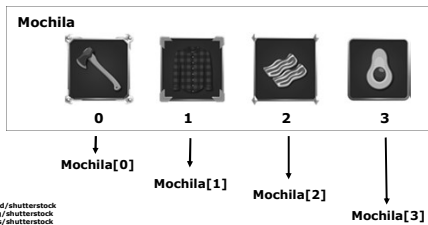
- Estruturas de dados
  - Tuplas: ()
  - Listas: []
  - Dicionários: {}
- Aprenderemos também:
  - Conceito de método
  - Métodos para *strings*

### Tuplas

### Variáveis

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

### 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'`

```
mochila = [
    [0] [1] [2] [3]
    ['Machado', 'Camisa', 'Bacon', 'Abacate']
]

Machado
0 1 2 3 4 5 6
```

### 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

- Vejamos no Python

### Listas dentro de listas

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

```
      [0]      [1]      [2]
mochila = [['Cebola',0.39], ['Tomate',0.49], ['Maçã',0.89]]
          [0] [1]      [0] [1]      [0] [1]
```

- 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

### Dicionários

### Dicionários

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

- ▀ **Vejamos um dicionário no Python**

### **Métodos para dicionários**

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

- ▀ **Vamos praticar no Python**

### **Listas com dicionários**

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

### **Dicionários com listas**

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

**Trabalhando com métodos  
em *strings***

- 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*

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

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

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

- Vejamos em Python alguns exemplos

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

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

- Vejamos em Python alguns exemplos

