

Listas em Python

Dia 3

Prof Flávio Sousa Silva

Recode Pro

Agenda

- Porque usar Python?
- Um pouco mais de listas
- Dicionários
- Tuplas

- Porque usar Python?
- Por que listas no Python é importante?
- Qual sua relação com a atividade de Cientista de Dados?
- Quais as linguagens mais utilizadas para análise de dados?

Métodos para listas (OO)

- seus métodos:

- append

- pop

- insert(a,b)

-

Duplicando listas

- `id()`
- `copy`

Dicionários

- Dicionários consistem em uma estrutura de dados similar às listas.
- Entretanto, as propriedades de acesso são diferentes.
- Um dicionário é composto por um conjunto de chaves e valores.
- O dicionário em si consiste em relacionar uma chave a um valor específico.
- Criamos dicionários em Python utilizando chaves
(`{ }`)

Dicionários

- Vejamos como exemplo a tabela:

Produto	Preço
Alface	R\$ 0,45
Batata	R\$ 1,20
Tomate	R\$ 2,30
Feijão	R\$ 1,50

Dicionários

- Exemplo de criação de um dicionário

```
>>> Tabela = {"Alface": 0.45,  
              "Batata": 1.20,  
              "Tomate": 2.30,  
              "Feijão": 1.50}
```


Dicionários

- Diferente de listas, dicionários utilizam suas chaves como índice.
- Quando atribuímos um valor a uma chave, duas coisas podem ocorrer.
 - Se a chave já existe: o valor associado é alterado para um novo valor.
 - Se a chave não existe: uma nova chave será adicionada ao dicionário.

```
>>> tabela = { "Alface": 0.45,  
...           "Batata": 1.20,  
...           "Tomate": 2.30,  
...           "Feijão": 1.50 }  
>>> print(tabela["Tomate"]) ❶  
2.3  
>>> print(tabela)  
{'Batata': 1.2, 'Alface': 0.45, 'Tomate': 2.3, 'Feijão': 1.5}  
>>> tabela["Tomate"] = 2.50 ❷  
>>> print(tabela["Tomate"])  
2.5  
>>> tabela["Cebola"] = 1.20 ❸  
>>> print(tabela)  
{'Batata': 1.2, 'Alface': 0.45, 'Tomate': 2.5, 'Cebola': 1.2, 'Feijão': 1.5}
```

Tuplas

- Tuplas podem ser vistas como listas em Python, com a grande diferença de serem imutáveis.
- Tuplas são ideais para representar listas de valores constantes
- Para criar tuplas utilizamos parênteses em vez de colchetes

```
>>> Tupla = ("a", "b", "c")  
>>> Tupla  
( 'a', 'b', 'c')
```