Código em Python Complementar

Profa. Solange Kanso
solange.kanso@uni9.pro.br

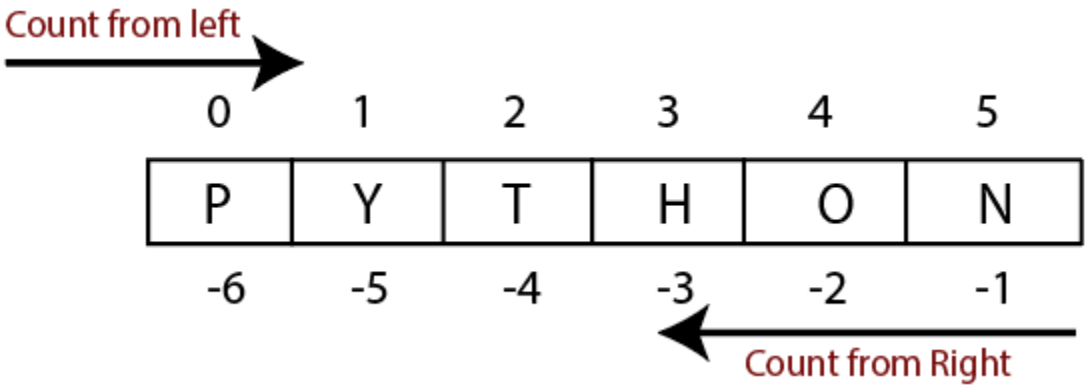
Prof. Leandro Romualdo

UNINOVE/SP
1º/2024

COMPLEMENTO – AULAS 05, 06 E 07

LISTAS = REPRESENTADA POR [ ] E OS ITENS SEPARADOS POR “,”

- É uma estrutura de dados indexados e armazenados em sequência, onde cada elemento possui uma posição que é identificada por um índice.
- O 1o elemento à esquerda da lista tem o índice 0 e seguindo a sequência, acrescente +1 (**da esquerda para a direita**). É possível realizar a indexação dos itens por valores negativos, nesse caso a indexação ocorre **da direita para esquerda** com decremento (-1).



Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

LISTAS = REPRESENTADA POR [ ] E OS ITENS SEPARADOS POR “,”

- Exemplo: lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
- Pode armazenar qualquer tipo de dado (string, int, float, etc.).
- É possível criar listas dentro de listas — os populares *nested list*
  - lista_cores = [‘verde’,[‘vermelho’, ‘azul’, ’amarelo’]]

- Identificar um item
  - lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
  - cor_vermelho= lista_cores[0]
  - print(cor_vermelho)
  - vermelho

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

LISTAS = REPRESENTADA POR [ ] E OS ITENS SEPARADOS POR “,”

↘ Alterar um item

- lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
- lista_cores[2]= “rosa”
- print(lista_cores)
- [‘vermelho’, ‘azul’, ’rosa’]

↘ Adicionar um item**: pelo append**

- lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
- lista_cores.append(”rosa”)
- print(lista_cores)
- [‘vermelho’, ‘azul’, ’amarelo’, ’rosa’]

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

LISTAS = REPRESENTADA POR [ ] E OS ITENS SEPARADOS POR “,”

↘ Adicionar um item: **pelo insert**

- lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
- lista_cores.**insert**(0, ”rosa”)
- print(lista_cores)
- [‘rosa, ‘vermelho’, ‘azul’, ’amarelo’]

↘ Remover um item: **pelo remove**

- lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
- lista_cores.**remove**(“vermelho”)
- print(lista_cores)
- [‘azul’, ’amarelo’]

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

LISTAS = REPRESENTADA POR [ ] E OS ITENS SEPARADOS POR “,”

↘ Remover um item: **pelo pop**

- lista_cores = [‘vermelho’, ‘azul’, ’amarelo’]
- lista_cores.pop(0)
- print(lista_cores)
- [‘azul’, ’amarelo’]

É recomendado a **utilização de listas** ao trabalhar com estruturas homogêneas, ou seja, quando todos os elementos são do mesmo tipo (strings, int, float, etc.), e/ou quando existe a necessidade de alterar os itens da coleção.

Outras informações:
https://docs.python.org/3/tutorial/datastructures.html#more-on-lists

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

TUPLAS = REPRESENTADA POR ( ) E OS ITENS SEPARADOS POR “,”

↘ Similar as listas, porém é **imutável**. Assim, após a criação não é possível realizar alterações no seu estado.

↘ Identificar um item

- tupla_cores = (‘vermelho’, ‘azul’, ’amarelo’)
- cor_vermelho= tupla_cores(0)
- print(cor_vermelho)
- vermelho

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

TUPLAS = REPRESENTADA POR ( ) E OS ITENS SEPARADOS POR “,”

↘ Quando uma tupla possui apenas um item, devemos adicionar uma vírgula após o elemento para que o Python reconheça o objeto como tupla. Caso contrário, é obtido um objeto do tipo String.

↘ Identificar um item

- tupla_amarelo = (’amarelo’,)
- print(type(tupla_amarelo))
- <class ‘tuple’>

- str_amarelo = (’amarelo’) **# atenção, retirei a vírgula**
- print(type(str_amarelo))
- <class ‘str’>

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

TUPLAS = REPRESENTADA POR ( ) E OS ITENS SEPARADOS POR “,”

Utilizadas quando os seus itens são pré-definidos e **geralmente não necessitam de alterações**, alguns exemplos: meses do ano, datas comemorativas, dias da semana, estações do ano, nomes de cidades, etc. Também é recomendável sua utilização quando trabalhamos com **dados heterogêneos, ou seja, com elementos de tipos diferentes**

Outras informações:

https://docs.python.org/3/tutorial/datastructures.html#tuples-and-sequences

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

DICIONÁRIOS = REPRESENTADA POR { } E OS ITENS SEPARADOS POR “,”

- Dicionários são estruturas que compreendem um conjunto de pares: chave e valor. Cada chave individual possui um valor associado. Esse tipo de associação se dá o nome de **coleção associativa desordenada**.
- Assim como as listas, são **mutáveis**

- Identificar um item
  - bandas_dicionario = {"Rock Alternativo":"Red Hot Chili Peppers", "Indie Rock": "The Strokes", "Punk":"Ramones"}
  - banda_punk = bandas_dicionario["Punk"]
  - print(banda_punk)
  - Ramones

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

DICIONÁRIOS = REPRESENTADA POR { } E OS ITENS SEPARADOS POR “,”

↘ Identificar um item

- bandas_dicionario = {"Rock Alternativo":"Red Hot Chili Peppers", "Indie Rock": "The Strokes", "Punk":"Ramones"}
- bandas_punk = bandas_dicionario.**get**("Punk", "Não foram encontradas bandas para o estilo.")
- print(bandas_punk)
- Ramones

OU

- bandas_heavy_metal = bandas_dicionario.get("Heavy Metal", "Não foram encontradas bandas para o estilo.")
- print(bandas_heavy_metal)
- Não foram encontradas bandas para o estilo.

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

DICIONÁRIOS = REPRESENTADA POR { } E OS ITENS SEPARADOS POR “,”

↘ Adicionar um item: **pelo update**

- bandas_dicionario = {"Rock Alternativo":"Red Hot Chili Peppers", "Indie Rock": "The Strokes", "Punk":"Ramones"}
- bandas_dicionario.**update**({"Heavy Metal": "Iron Maiden"})
- print(bandas_dicionario)
- {'Rock Alternativo': 'Red Hot Chili Peppers', 'Indie Rock': 'The Strokes', 'Punk': 'Ramones', 'Heavy Metal': 'Iron Maiden'}

- Adicionar um item: **pela nova chave e valor associado**
- bandas_dicionario["Heavy Metal"] = "Black Sabbath"
- print(bandas_dicionario)
- {'Rock Alternativo': 'Red Hot Chili Peppers', 'Indie Rock': 'The Strokes', 'Punk': 'Ramones', 'Heavy Metal': 'Black Sabbath'}

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

DICIONÁRIOS = REPRESENTADA POR { } E OS ITENS SEPARADOS POR “,”

↘ Remover um item: **pelo pop**

- bandas_dicionario = {"Rock Alternativo":"Red Hot Chili Peppers", "Indie Rock": "The Strokes", "Punk":"Ramones"}
- bandas_dicionario.**pop**("Heavy Metal", None)
- print(bandas_dicionario)
- {'Rock Alternativo': 'Red Hot Chili Peppers', 'Indie Rock': 'The Strokes', 'Punk': 'Ramones'}

- Remover um item: **pelo del**
- **del** bandas_dicionario["Rock Alternativo"]
- print(bandas_dicionario)
- {'Indie Rock': 'The Strokes', 'Punk': 'Ramones'}

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

COMPLEMENTO – AULAS 05, 06 E 07

DICIONÁRIOS = REPRESENTADA POR { } E OS ITENS SEPARADOS POR “,”

Utilizar quando existe a necessidade de armazenar dados de forma organizada, visto que, com o dicionário é possível mapear o valor de um elemento com uma chave única.

Outras informações:

https://docs.python.org/3/tutorial/datastructures.html#dictionaries

Fonte: https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796

↘

Obrigada!

Até a próxima aula ☺