

Faculdade de Tecnologia
Fatec Mauá

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

Prof. Me. Anderson Vanin

Listas, Tuplas e Dicionários em Python

Objetivo da aula: Introduzir e explicar os conceitos de listas, tuplas e dicionários em Python, e demonstrar como iterar sobre seus elementos utilizando laços de repetição.

Listas em Python (20 minutos):

As listas em Python são uma estrutura de dados flexível e versátil que permite armazenar uma coleção ordenada de itens. Cada item em uma lista é identificado por sua posição, ou índice, na lista. As listas são definidas usando colchetes `[]`, e os elementos dentro da lista são separados por vírgulas.

Principais características das listas:

- **Mutabilidade:** As listas em Python são mutáveis, o que significa que é possível alterar, adicionar e remover elementos após a criação da lista.
- **Ordenação:** Os elementos de uma lista são mantidos em uma ordem específica, o que permite acessá-los por meio de seus índices.
- **Heterogeneidade:** As listas podem conter elementos de diferentes tipos de dados, como números, strings, outras listas, etc.
- **Iterabilidade:** É fácil iterar sobre os elementos de uma lista usando loops `for`.

Uma lista no Python pode armazenar qualquer tipo de dado primitivo (string, int, float, etc.). Também é possível criar listas dentro de listas.

Exemplo:

```
# Exemplo de criação e manipulação de uma lista
frutas = ['maçã', 'banana', 'laranja', 'morango']

# Acessando elementos da lista
print("A primeira fruta é:", frutas[0])
print("A última fruta é:", frutas[-1])

# Adicionando um elemento à lista
frutas.append('uva')

# Removendo um elemento da lista
frutas.remove('banana')

# Modificando um elemento da lista
frutas[2] = 'pêssego'

# Iterando sobre os elementos da lista
print("Todas as frutas:")
for fruta in frutas:
    print(fruta)
```

Tuplas em Python (15 minutos):

As tuplas são estruturas de dados semelhantes às listas, porém são imutáveis, o que significa que uma vez criadas, elas não podem ser modificadas. As tuplas são definidas usando parênteses **()**, e os elementos dentro da tupla são separados por vírgulas.

Principais características das tuplas:

- **Imutabilidade:** Ao contrário das listas, as tuplas não podem ser alteradas depois de criadas. Isso garante que os elementos da tupla permaneçam inalterados durante toda a execução do programa.
- **Ordenação:** Os elementos de uma tupla são mantidos em uma ordem específica, assim como em listas.

Faculdade de Tecnologia
Fatec Mauá

- **Heterogeneidade:** Assim como as listas, as tuplas podem conter elementos de diferentes tipos de dados.
- **Iterabilidade:** É possível iterar sobre os elementos de uma tupla usando loops for.

Exemplo:

```
# Exemplo de criação e manipulação de uma tupla
cores = ('vermelho', 'verde', 'azul', 'amarelo')

# Acessando elementos da tupla
print("A primeira cor é:", cores[0])
print("A última cor é:", cores[-1])

# Iterando sobre os elementos da tupla
print("Todas as cores:")
for cor in cores:
    print(cor)
```

Dicionários em Python (20 minutos):

Os dicionários são estruturas de dados que permitem armazenar pares de chave-valor, onde cada valor é associado a uma chave única. Os dicionários são definidos usando chaves `{}` e consistem em uma série de pares chave-valor separados por vírgulas.

Principais características dos dicionários:

- **Flexibilidade:** Os dicionários são uma estrutura de dados flexível e poderosa, permitindo que os valores sejam acessados e modificados de forma eficiente por meio de suas chaves.
- **Chaves únicas:** Cada chave em um dicionário deve ser única, mas os valores podem ser duplicados.
- **Não ordenação:** Os elementos de um dicionário não são mantidos em uma ordem específica. Portanto, não é possível acessar os elementos de um dicionário por meio de índices.

Faculdade de Tecnologia
Fatec Mauá

- **Iterabilidade sobre chaves:** É fácil iterar sobre as chaves de um dicionário usando loops for.

Exemplo:

```
# Exemplo de criação e manipulação de um dicionário
pessoa = {'nome': 'João', 'idade': 30, 'cidade': 'São Paulo'}

# Acessando valores do dicionário
print("Nome:", pessoa['nome'])
print("Idade:", pessoa['idade'])
print("Cidade:", pessoa['cidade'])

# Adicionando um novo par chave-valor ao dicionário
pessoa['profissão'] = 'Engenheiro'

# Modificando o valor de uma chave existente no dicionário
pessoa['idade'] = 35

# Removendo um par chave-valor do dicionário
del pessoa['cidade']

# Iterando sobre as chaves do dicionário
print("Informações da pessoa:")
for chave in pessoa:
    print(chave + ":", pessoa[chave])
```

Atividade Prática: Explorando Listas, Tuplas e Dicionários

Objetivo: Criar, manipular e iterar sobre listas, tuplas e dicionários em Python.

Criar as seguintes estruturas de dados:

- Uma **lista** de cores favoritas.
- Uma **tupla** de nomes de países.
- Um **dicionário** de informações pessoais, onde as chaves são *'nome'*, *'idade'* e *'cidade'*, escolha os valores a serem fornecidos.

Faculdade de Tecnologia
Fatec Mauá

Manipulação das Estruturas de Dados:

Realizar as seguintes operações em cada estrutura de dados:

- Adicionar um novo elemento à lista de cores favoritas.
- Modificar um elemento da tupla de nomes de países.
- Adicionar uma nova entrada ao dicionário de informações pessoais.
- Remover uma entrada do dicionário de informações pessoais.

Iteração sobre os Elementos:

Iterar sobre os elementos de cada estrutura de dados e exibir os resultados na tela. Usar um loop *for* para isso.

- Para a lista de cores favoritas, imprimir cada cor em uma linha separada.
- Para a tupla de nomes de países, imprimir cada país em uma linha separada.
- Para o dicionário de informações pessoais, imprimir cada chave e seu respectivo valor em uma linha separada.