Estrutura de dados Homogêneas (Matrizes); Tuplas; Dicionários

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I

Estruturas Homogêneas Básicas

- Matrizes
  - Conceito
  - Exemplos
  - Manipulação

Matrizes

Matrizes

“**Uma matriz generaliza o vetor.** Uma matriz é o caso **multidimensional**. Um **vetor é uma matriz com apenas uma linha (ou, uma coluna)**”

Qual a necessidade?

- Vetores: Sequência simples de dados
  - `notas_aluno1 = [80, 90, 75, 85]`
  - `notas_aluno2 = [80, 90, 75, 85]`
- Matrizes
  - ```
    notas_alunos = [
      - [80, 90, 75, 85],
      - [70, 65, 80, 90],
      - [85, 80, 95, 70]
    ]
    ```

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I
    ```

Matrizes

“Uma referência às notas do primeiro aluno (linha de índice 0 - zero) é feita por Nota[0][0] para o primeiro bimestre, Nota[0] [1]para o segundo bimestre, a reavaliação é Nota[0][2] e a final é Nota[0][3].”

	0	1	2	3
0	3.0	4.0	8.5	6.0
1	7.5	5.5	4.5	8.0
2	6.0	7.0	5.0	7.0

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I

Matrizes

```
a = [
    [3.0, 4.0, 8.5, 6.0],
    [7.5, 5.5, 4.5, 8.0]
]
```

```
print(a[0][2]) -> Linha 1, coluna 3 (Lembrando: Começamos em 0)
```

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I

Estrutura Homogênea?

- Em python não precisa ser homogênea;
- Podemos criar Matrizes Heterogêneas

```
matriz heterogenea = [
    [1, 2, 3],
    [4.5, 6.7, 8.9],
    ["a", "b", "c"]
]
```

Tuplas

Então o que são tuplas?

- Listas são poderosas, mas **são aplicáveis em todos os cenários?**

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I

Tuplas

- Imutáveis!!!!
- Não somente isso …
  - **Registro sem nomes de campos (RAMALHO; 2015)**

```
# Definindo uma tupla para representar um registro de pessoa
pessoa = ("Alice", 30, "Engenheira")

# Acessando os campos da pessoa usando índices
nome = pessoa[0]
idade = pessoa[1]
profissao = pessoa[2]

# Exibindo os valores dos campos
print("Nome:", nome)
print("Idade:", idade)
print("Profissão:", profissao)
```

Então o que são tuplas?

```
lista = (112938, 2923847)
```

```
print(lista[0])
```

```
Se tentarmos alterar o indice:
lista[0] = -1
```

```
TypeError: 'tuple' object does not support item assignment
```

Dicionários

Sumário

- Dicionários
  - Conceito
  - Exemplos

Dicionários

- Dicionário é um exemplo de tabela hash;
- Hash table é uma estrutura de dados utilizada para implementar um array **associativo**;
- Cada **valor** tem uma chave **associada** a ele

https://pt.slideshare.net/mcastrosouza/python-dicionrios

Dicionários

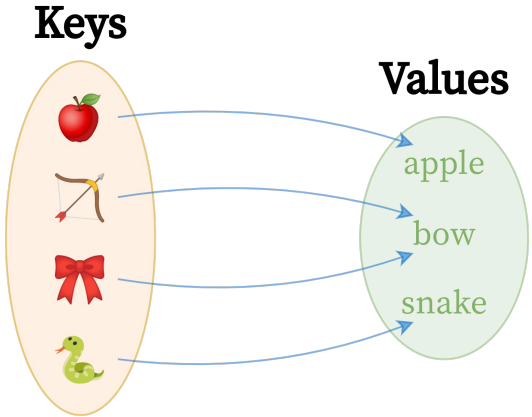
Array

Value
New York
Boston
Mexico
Kansas
Detroit
California

Hash Table

Key	Value
1	New York
2	Boston
3	Mexico
4	Kansas
5	Detroit
6	California

https://qph.cf2.quoracdn.net/main-qimg-8c7683eeaac173b3337578fadcdc521b



https://files.realpython.com/media/dict_map.6ea6c6e33b4b.png

Em python

```
dicionario = {}


print(type(dicionario))
```

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I

Quando utilizar?

Imagine que temos uma lista de contatos com nome e telefone:

nomes = [‘contato 1’, ‘contato 2’, ‘contato 3’]

telefones = [11111, 22222, 33333]

Se quisermos o nome e telefone do contato 1, como faremos?

E se eu quiser listar os nomes e telefones?

Quando utilizar?

Podemos organizar nossos dados melhor utilizando dicionários:

```
contato = {
        ‘nome’: ‘Teste’,
        ‘telefone’: ‘3434’,
        ‘email’: ‘teste@email.com’,
}
```

Praticando!

Primeiro vamos **criar** nosso dicionário:

```
contato = {
        ‘nome’: ‘Teste’,
        ‘telefone’: ‘3434’,
        ‘email’: ‘teste@email.com’,
}
```

Praticando!

Agora vamos **mostrar** o telefone do nosso contato 1:

print(contato[‘telefone’])

Praticando!

Para exibir o nome e o telefone devemos **converter** em lista com items

for chave, valor in **contatos.items()**:

```
    print(chave)
    print(valor)
```

Praticando!

Podemos utilizar em conjunto: **listas e dicionários**

```
contatos = [
    {
        "nome": "Contato 1",
        "telefone": "394837"
    },
    {
        "nome": "Contato 2",
        "telefone": "394837"
    },
    {
        "nome": "Contato 3",
        "telefone": "394837"
    },
]


print(contatos[0]['nome'])
```

Praticando!

O que vai ser listado ao chamar contato[0]? Erro?

Mutabilidade

Até agora, você tem visto dois tipos compostos: strings, que são compostos de caracteres; e listas, que são compostas de elementos de qualquer tipo. Uma das diferenças que notamos é que os elementos de uma lista podem ser modificados, mas os caracteres em uma string não. Em outras palavras, strings são **imutáveis** e listas são **mutáveis**.

Tente fazer:

```
name = “exemplo”
name[0] = “E”
```

O que irá acontecer?

https://aprendacompy.readthedocs.io/pt/latest/capitulo_09.html

Mutabilidade

Até agora, você tem visto dois tipos compostos: strings, que são compostos de caracteres; e listas, que são compostas de elementos de qualquer tipo. Uma das diferenças que notamos é que os elementos de uma lista podem ser modificados, mas os caracteres em uma string não. Em outras palavras, strings são **imutáveis** e listas são **mutáveis**.

Agora:

```
exemplo = {“a”: 1}
exemplo[“a”] = 2

print(exemplo)
```

https://aprendacompy.readthedocs.io/pt/latest/capitulo_09.html

Mutabilidade [Referenciando]

```
object1 = {“a”: 10}

object2 = object1

object2[“a”] = 20

print(object1[“a”])
```

Qual a saída? Explique.

Referências

1. https://educapes.capes.gov.br/bitstream/capes/176522/2/TextoAEDI_SI_UFAL_AiltonCruz.pdf

Slide baseado no tutorial do Professor Ailton Cruz, Algoritmo e Estrutura de Dados I