UNIFCV
CENTRO UNIVERSITÁRIO CIDADE VERDE

Introdução a linguagem Python

Listas, arrays, dicionários e tuplas

Prof. Me. Leonardo G. Catharin

prof_leonardo@unifcv.edu.br

Listas

- Conjunto sequencial de valores, em que cada valor é identificado por um índice.

- Em Python, uma lista pode valores de diferentes tipos.

- Em Python...

  lista = ['Faculdade' , 2 , [2, 3, 10] , (3, 'teste')] /* lista de diferentes tipos de valores */

- Para acessar um valor pelo seu índice,

  > **lista[0]**;

  'Faculdade'

  > **lista[0][1];**

  3

Manipulação de listas

- Funções.

Função	Descrição	Aplicação
len	Retorna o tamanho da lista	**len**(lista)
min	Retorna o menor valor da lista	**min**(lista)
max	Retorna o maior valor da lista	**max**(lista)
sum	Soma os valores da lista	**sum**(lista)
append	Adiciona um novo valor no final da lista	lista.**append**(x)
extend	Insere uma lista no final de outra lista	lista.**extend**([1, 2, 3])
del	Remove um elemento	**del** lista[1]
in	Verifica se um valor pertence à lista	2 **in** lista
sort	Ordena os elementos	lista.**sort**()
reverse	Inverte os elementos	lista.**reverse**()

Manipulação de listas

- Operações.

Função	Descrição	Aplicação
+	Concatenação	lista_a **+** lista_b
*	Retorna o menor valor da lista	lista_a ***** 3

- Fatiamento.

lista = [‘Faculdade’ , 2 , [2, 3, 10] , (3, ‘teste’)]

> lista[1:3]

[2, [2,3,10]]

Manipulação de listas

- Função range()
  - Definição de intervalo de valores inteiros
  - Pode ser utilizado para criar listas
  - Pode ter de 1 a 3 parâmetros:
    - **range(n)**: gera um intervalo de 0 a n-1;
    - **range(i, n)**: gera um intervalo de i a n-1;
    - **range(i, n, p)**: gera um intervalo de i a n-1 com intervalo p entre os números;

- Em Python...

```
lista1 = list(range(2, 11, 3))
> print(lista1)
[2, 5, 8]
```

Arrays

- Listas nem sempre é a melhor opção.
  - Pouco eficiente para armazenamento de grandes quantidades de registros (por exemplo, 10 milhões de valores de ponto flutuante)

- Nesses casos, **arrays** são mais eficientes, pois armazenam os bytes compactos que representam seus valores de máquina e não o objeto float inteiro.

- Arrays utilizam as mesmas operações de listas (pop, insert, extend e etc.)

- Em Python...

```
from array import array
from random import Random

floats = array('d', (random() for i in range(10**7)))
```

Tuplas

- Conjunto sequencial de valores, porém são **imutáveis**.

- Tuplas também são usadas como registros sem nomes de campos.
  - A quantidade de itens, geralmente, será fixa e sua ordem é relevante.

- Em Python...

  coordenadas = (33.9425, -118)

  cidade = (‘Maringá’, 2021, ...)

Desempacotamento de tuplas

- Funciona com qualquer objeto **iterável**.

- Em Python...

```
cidade, ano, area = (‘Maringá’, 2021, 8014)

coordenadas = (33.9425, -118)
latitude, longitude = coordenadas

_ , nome = (‘teste’, ‘nomedoarquivo.txt’)
```

Desempacotamento de tuplas

- Funciona com qualquer objeto **iterável**.

- Em Python...

```
cidade, ano, area = (‘Maringá’, 2021, 8014)

coordenadas = (33.9425, -118)
latitude, longitude = coordenadas

_ , nome = (‘teste’, ‘nomedoarquivo.txt’) # _ é utilizado com variável descartável
```

- Captura de itens excedentes (*****).

```
a, b, *rest = range(5)
```

Dicionários

- Conjunto de valores, em que cada valor é associado a uma chave (key).

- Em Python...

```
pessoa = {
    'nome' : 'João da Silva',
    'idade': 24,
    'rg'   : '541561111',
    'cpf'  : '09877701947'
}
```

- É possível acrescentar valores ou modificá-los.

Manipulação de dicionários

- Funções.

Função	Descrição	Aplicação
del	Exclui um item informando a chave	**del** dict[‘nome’]
in	Verifica se uma chave existe no dicionário	‘nome’ **in** dict
keys	Retorna as chaves de um dicionário	dict**.keys**()
values	Retorna os valores de um dicionário	dict**.values**()

Exemplos

- Exemplo 1: escreva um programa que leia n nomes digitados pelo usuário e retorne uma lista com os nomes em ordem alfabética no final da execução. [*arquivo: ordemalfabetica.py*]

- Exemplo 2: gere uma lista de todos os números pares entre 0 e 100 e armazene em uma lista os múltiplos de 3. [*arquivo: numeros-pares-multiplos-3.py*]

- Exemplo 3: leia o nome e a nota de n alunos e depois calcule a média aritmética das notas. [*arquivo: notas.py*]

Exercícios

- Exercício 1: crie um programa para armazenar e acessar dados de uma lista de pessoas. [*arquivo: crud.py*]
  - Considere como atributos: nome, rg, cpf, idade e cidade.
  - O programa deve ter 4 funções: **cadastrar** nova pessoa, **listar** pessoas, **remover** pessoas pelo índice e **fechar**.
  - Ao fechar o programa, a lista das pessoas cadastradas deverá ser impressa.

Referências

- Ramalho, L. Python Fluente: Programação clara, concisa e eficaz. Novatec Editora, 2015.
- PET - ADS. Introdução à Programação com Python. IFSP – São Carlos.