PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
INTELIGÊNCIA ARTIFICIAL APLICADA
ADS1233
PROF. MSC. ANIBAL SANTOS JUKEMURA

PYTHON

PUC GOIÁS

Agenda:

- Strings
- Sequências e listas
- Estruturas de repetição

PUC GOIÁS

Strings

Prompt de Comando - python

```
C:\>python
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> "Bom dia"
'Bom dia'
>>> "Apostrofos: \'"
"Apostrofos: '"
>>> a = "Darth "
>>> b = "Japa"
>>> a+b
'Darth Japa'
>>> a= "Idade: "
>>> b=str(47)
>>> a+b
'Idade: 47'
>>>
```

Sequencias e Listas

- Uma sequência é uma coleção de itens ordenada posicionalmente. E você pode se referir a qualquer item na sequência usando seu número de índice, por exemplo, s[0] e s[1].

- Em Python, o índice de sequência começa em 0, não em 1. Portanto, o primeiro elemento é s[0] e o segundo elemento é s[1]. Se a sequência s tiver n itens, o último item será s[n-1].

- Python possui os seguintes tipos de sequência integrados: listas, bytearrays, strings, tuplas, intervalo e bytes. Python classifica os tipos de sequência como mutáveis e imutáveis.

PUC GOIÁS

Sequencias e Listas

- Os **tipos de sequência mutáveis são listas e bytearrays**, enquanto os **tipos de sequência imutáveis** são strings, tuplas, intervalo e bytes.

- Uma sequência pode ser **homogênea** ou **heterogênea**. Numa sequência homogênea, todos os elementos têm o mesmo tipo. Por exemplo, strings são sequências homogêneas onde cada elemento é do mesmo tipo.

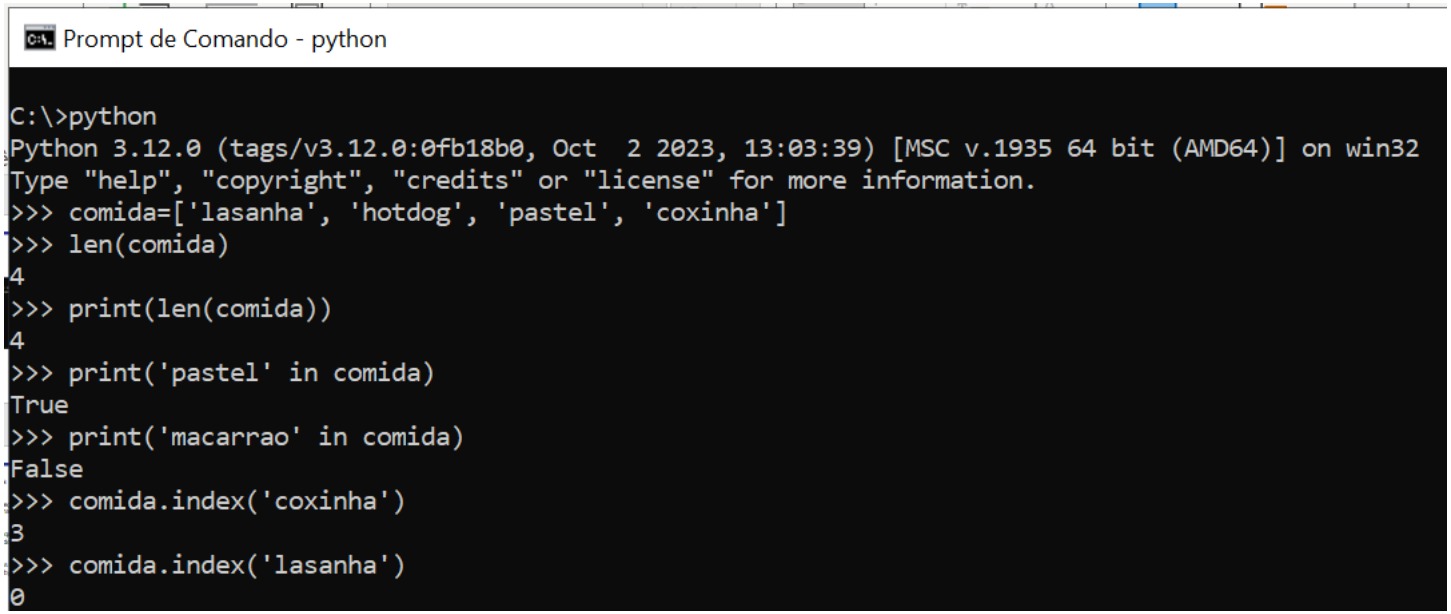
- As **listas**, entretanto, **são sequências heterogêneas** onde você pode armazenar elementos de diferentes tipos, incluindo números inteiros, strings, objetos, etc.

- Em geral, os tipos de sequências homogêneas são mais eficientes que os heterogêneos em termos de armazenamento e operações.

PUC GOIÁS

Sequencias e Listas

- Operações: contagem, busca , indexação



PUC GOIÁS

Sequencias e Listas

- Operações: impressão, listagem, concatenação

Prompt de Comando - python

```
C:\>python
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935
Type "help", "copyright", "credits" or "license" for more information.
>>> inteiros = [10,20,30,40,50,60,70,80,90]
>>> inteiros
[10, 20, 30, 40, 50, 60, 70, 80, 90]
>>> ad = [100,110]
>>> inteiros = inteiros + ad
>>> inteiros
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]
>>> inteiros.index(60)
5
>>> ad2 = [50, 90]
>>> inteiros = inteiros + ad2
>>> inteiros
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 50, 90]
>>> inteiros.index(50, 4)
4
>>> inteiros.index(50, 5)
11
>>>
```

PUC GOIÁS

Sequencias e Listas

- Operações: particionamento, min, max, “apendar”, repetição

Prompt de Comando - python

```
C:\>python
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license" for more information.
>>> inteiros = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 50, 90]
>>> print(inteiros[2:7])
[30, 40, 50, 60, 70]
>>> print(inteiros[2:7:2])
[30, 50, 70]
>>> min(inteiros)
10
>>> max(inteiros)
110
>>> inteiros.append(300)
>>> inteiros
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 50, 90, 300]
>>> print(inteiros*3)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 50, 90, 300, 10, 20, 30, 40, 50, 60, 70, 80, 90,
 100, 110, 50, 90, 300, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 50, 90, 300]
>>>
```

PUC GOIÁS

EXERCÍCIOS

1. Escreva um programa em Python que verifique se existe a letra "S" na frase "My name is Superman“

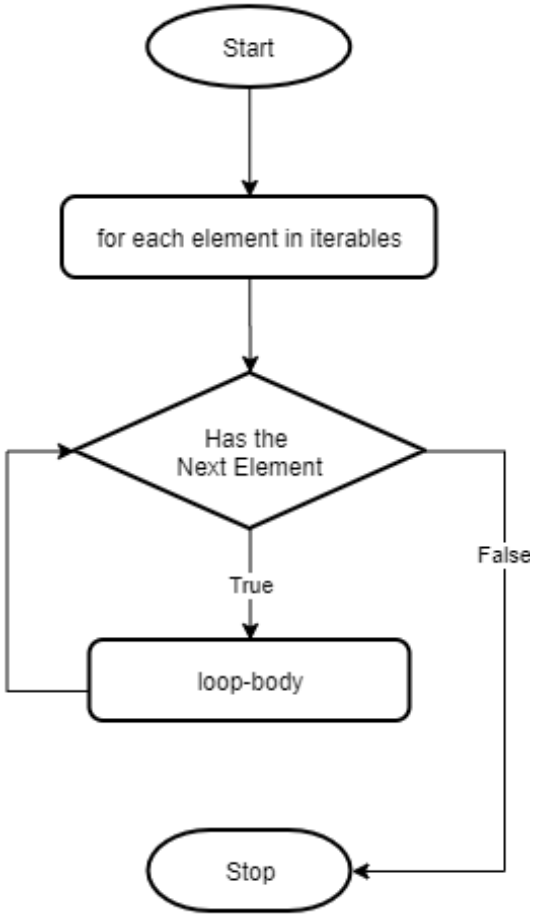
2. Escreva um programa em Python que peça uma frase para o usuário e, em seguida, imprima-a em ordem reversa.

3. Escreva um programa em Python que: Inicie a sequência numeros = [100,200,300,400,500,400,600]. Mostre a lista normalmente e em ordem inversa. Mostre quantas vezes aparece o número 400. Imprima o maior e o menor número da sequência. Adicione o valor 900 no final da sequência. Imprima a quantidade total de elementos. Imprima a sequência de forma ordenada.

PUC GOIÁS

Estrutura de repetição: FOR

```
for index in range(n):
    statement
```



Exemplos:

```
for index in range(5):
print(index + 1)
```

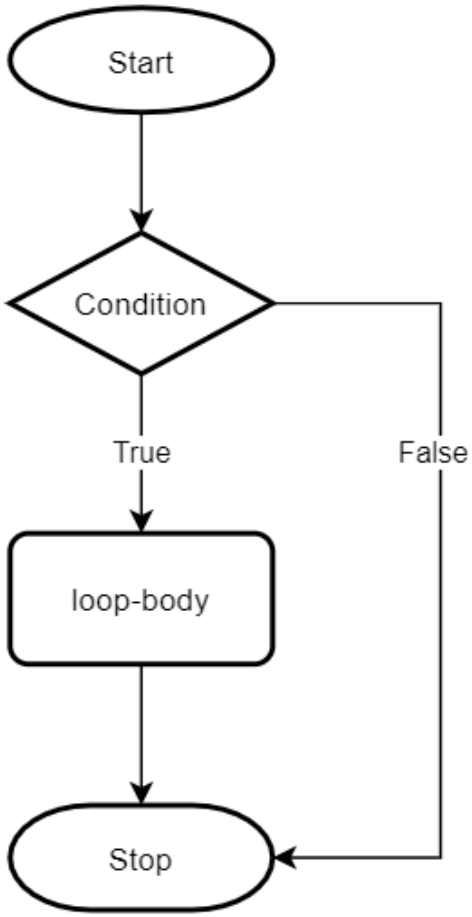
```
for index in range(1, 6):
print(index)
```

```
for index in range(0, 11, 2):
print(index)
```

PUC GOIÁS

Estrutura de repetição: WHILE

```
while condition:
    body
```



Exemplo:

```
max = 5
counter = 0
while counter < max:
    print(counter)
    counter += 1
```

PUC GOIÁS

EXERCÍCIOS

1. Faça um programa que peça uma nota, entre zero e dez. Mostre uma mensagem caso o valor seja inválido e continue pedindo até que o usuário informe um valor válido.

2. Faça um programa que leia um nome de usuário e a sua senha e não aceite a senha igual ao nome do usuário, mostrando uma mensagem de erro e voltando a pedir as informações.

3. Supondo que a população de um país A seja da ordem de 80000 habitantes com uma taxa anual de crescimento de 3% e que a população de B seja 200000 habitantes com uma taxa de crescimento de 1.5%. Faça um programa que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.

PUC GOIÁS

BIBLIOGRAFIA COMPLEMENTAR

- GRUS, J. **Data Science do zero: noções fundamentais com Python**. Rio de Janeiro: Alta Books, 2021.

- https://www.pythontutorial.net/