Data Science & Big Data

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Listas e Tuplas

Prof. André Grégio

DSBD

Listas

Listas

Lista de compras

Precisamos comprar: maçãs, leite, arroz, frango

Listas

Lista de compras

Precisamos comprar: maçãs, leite, arroz, frango

Isso é uma lista!

Em Python

Python possui **nativamente** uma **estrutura de dados** para listas

Para criar uma lista em Python:

```
nome_lista = [item1, item2, item3, …]
```

Questão de Ordem!

Uma lista é uma coleção de elementos **em uma ordem particular.**

Os itens vão se manter na ordem que você definiu.

Exceto se você explicitamente pedir para que a ordem seja alterada

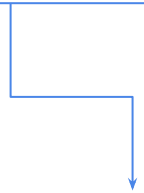
Exemplo

```
lista_compras = ["maçãs", "leite", "arroz", "frango"]
```

Exemplo

Cada item da lista é uma string. Então não esqueça das aspas.

```
lista_compras = ["maçãs", "leite", "arroz", "frango"]
```



Boa prática: como listas possuem múltiplos itens, utilize um nome no plural!

Iterando

Existem **várias formas** para acessar (iterar) os itens da lista

Uma simples é usando uma versão especial do loop *for*

```
for nome_variavel in nome_lista:
    faz algo com nome_variavel
```

Exemplo

```
lista_compras = ["maçãs", "leite", "arroz", "frango"]
for item in lista_compras:
    print("Preciso comprar:", item)
```

Índices

Uma lista nada mais é do que um vetor indexado

E um nome melhor para as listas do Python talvez fosse vetor

Veja uma breve explicação aqui: docs.python.org/3/faq/design.html#how-are-lists-implemented-in-cpython

Toda lista é indexada.

O primeiro elemento é o 0

O segundo elemento é o 1

O terceiro elemento é o 2...

Acesso por índice

Você pode acessar a lista pelos seus índices. Para isso utilize o operador []

```
lista_compras = ["maçãs", "leite", "arroz", "frango"]

print("O primeiro item que preciso comprar é:",
lista_compras[0])
```

Qual o tamanho da lista?

A função len() retorna o tamanho de uma lista

```
lista_compras = ["maçãs", "leite", "arroz", "frango"]

tamanho = len(lista_compras)
print("A lista de compras possui:", tamanho, "itens")
```

Outra forma de iterar

Dessa forma, podemos iterar também de outras formas em uma lista:

```
for i in range(len(lista_compras)):
    print("O elemento", i, "da lista é", lista_compras[i])
```

Ou usando um loop while:

```
i = 0
while i < len(lista_compras):
    print("O elemento", i, "da lista é", lista_compras[i])
    i = i + 1
```

Intervalos

Podemos acessar um intervalo da lista utilizando [inicio:fim] onde

Início é o índice do início da sublista, que é incluído na sub lista
Fim é o índice de fim da sub lista, que não é incluído na sub lista

Intervalos

Podemos acessar um intervalo da lista utilizando [inicio:fim] onde

Início é o índice do início da sublista, que é incluído na sub lista
Fim é o índice de fim da sub lista, que não é incluído na sub lista

Exemplo

```
lista_compras = ["maçãs", "leite", "arroz", "frango", "macarrão"]
sublista = lista_compras[1:4]
for item in sublista:
    print(item)

print("Outra forma")
for item in lista_compras[2:6]:
    print(item)
```

Alterando

Você pode alterar o valor de um elemento da lista, acessando-o via []. Veja um exemplo:

```
lista_compras = ["maçãs", "leite", "arroz", "frango"]
lista_compras[1] = "açúcar"

for item in lista_compras:
    print(item)
```

Funções de listas

Funções de listas

Existem diversas funções que podem ser usadas com listas

Vamos ver apenas algumas

Veja uma lista completa em
docs.python.org/pt-br/3/tutorial/datastructures.html

Índices e contagens

```
lista_compras = ["maçãs", "leite", "arroz", "frango", "leite", "trigo"]

qtde_vezes = lista_compras.count("leite")
idx = lista_compras.index("leite")

print("Quantidade: ", qtde_vezes)
print("Primeiro idx: ", idx)
```

Índices e contagens

```
lista_compras = ["maçãs", "leite", "arroz", "frango", "leite", "trigo"]

qtde_vezes = lista_compras.count("leite")
idx = lista_compras.index("leite")

print("Quantidade: ", qtde_vezes)
print("Primeiro idx: ", idx)
```

Quantas vezes “leite” aparece na lista?

Qual o primeiro índice de “leite” na lista?

Ordenando

Você pode ordenar os elementos da lista usando sort()

```
lista_compras = ["maçãs", "leite", "arroz", "frango", "trigo"]
for item in lista_compras:
    print(item)

print("Ordenado")
lista_compras.sort()
for item in lista_compras:
    print(item)

print("Ordenado Decrescente")
lista_compras.sort(reverse = True)
for item in lista_compras:
    print(item)
```

Excluindo por índice

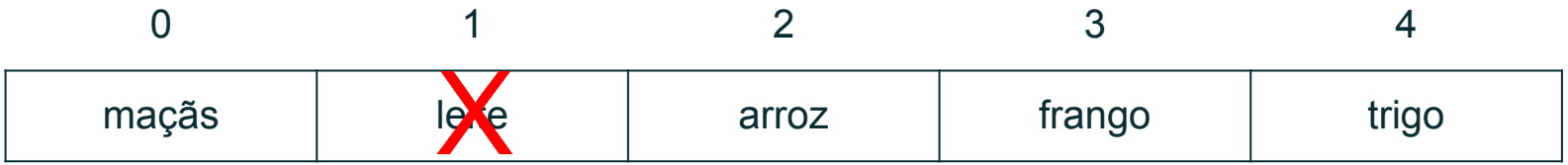
Para excluir um elemento da lista por índice, utilize

`del nome_lista[idx]`

```
lista_compras = ["maçãs", "leite", "arroz", "frango", "trigo"]

del lista_compras[1]

for item in lista_compras:
        print(item)
```



Excluindo por valor

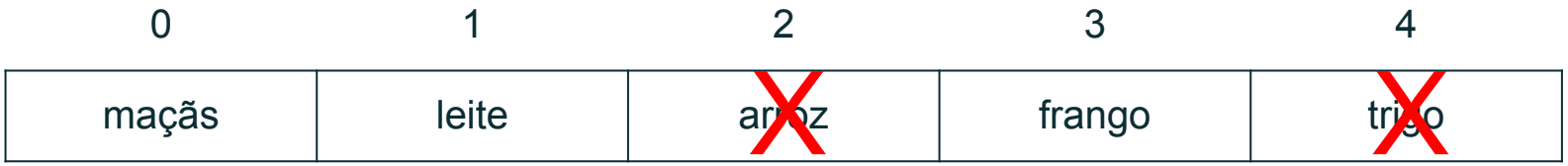
A função remove(conteúdo) remove **o primeiro** elemento com o conteúdo especificado

A função pop() remove o último elemento

```
lista_compras = ["maçãs", "leite", "arroz", "frango", "trigo"]

lista_compras.remove("arroz")
lista_compras.pop()

for item in lista_compras:
    print(item)
```



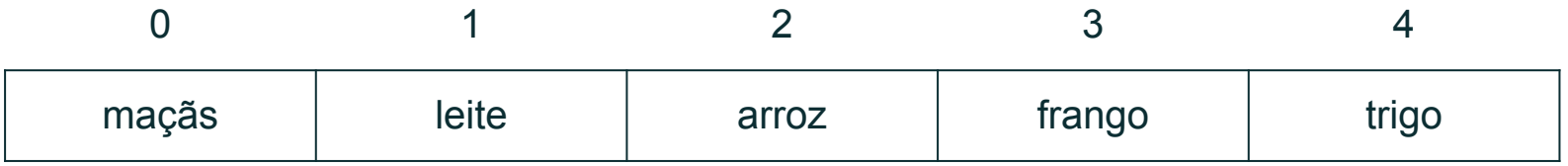
Inserindo

`insert(idx, item)` insere o item na posição

`append(item)` insere o item no final

lista_compras = ["maçãs", "leite", "arroz", "frango", "trigo"]

lista_compras.**insert**(2, "feijão")
lista_compras.**append**("tomate")



Inserindo

`insert(idx, item)` insere o item na posição

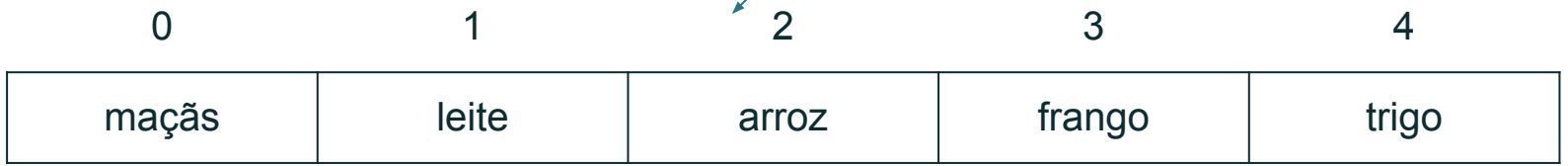
`append(item)` insere o item no final

lista_compras = ["maçãs", "leite", "arroz", "frango", "trigo"]

lista_compras.**insert**(2, "feijão")
lista_compras.**append**("tomate")

Vai inserir feijão aqui, e deslocar todos itens para a direita.

Vai inserir tomate aqui



Exemplo completo

```
lista_compras = [] #uma lista vazia
item = input("Digite um item ou sair: ")
while item != "sair":
    lista_compras.append(item)
    item = input("Digite um item ou sair: ")

for it in lista_compras:
    print(it)
```

Listas são heterogêneas

Em python, uma lista é heterogênea

Pode misturar todo tipo de item

Ter inteiros, strings, outras listas, ...

Listas são heterogêneas

Em python, uma lista é heterogênea

Pode misturar todo tipo de item

Ter inteiros, strings, outras listas, ...

Exemplo

```
lista_inteiros = [1,2,3]

elementos = ["casa", 1, "banana", lista_inteiros]

for item in elementos:
    print(item, type(item))
```

Arrays

Python provê arrays (vetores) homogêneos

Operam de maneira similar a lista mas em um **array, todos elementos precisam ser do mesmo tipo**

(Muito) Mais eficiente do ponto de vista computacional

Veja mais sobre arrays em docs.python.org/3/library/array.html

Tuplas

Tuplas

Tuplas em Python são similares a listas, no entanto tuplas são **imutáveis**

Depois de criada uma tupla, você não pode fazer modificações

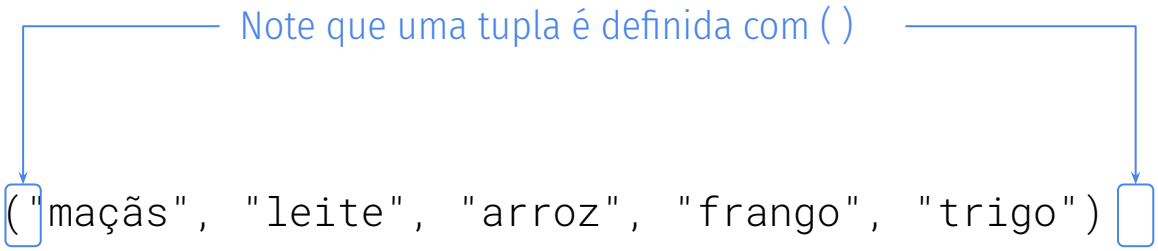
Exemplo

```
tupla_compras = ("maçãs", "leite", "arroz", "frango", "trigo")

for item in tupla_compras:
    print(item)

print("O item 1 é", tupla_compras[1])
```

Exemplo




```
tupla_compras = ("maçãs", "leite", "arroz", "frango", "trigo")

for item in tupla_compras:
    print(item)

print("O item 1 é", tupla_compras[1])
```

Exemplo

```
tupla_compras = ("maçãs", "leite", "arroz", "frango", "trigo")

for item in tupla_compras:
    print(item)

print("O item 1 é", tupla_compras[1])
```

```
tupla_compras[0] = "tomate"
```

TypeError: 'tuple' object does not support item assignment

Você não pode modificar uma tupla!

Quando usar

Sempre que você precisar de uma “lista” e garantir que ela não deve ser modificada, use tuplas.

Garantir que os itens não sejam modificados por acidente.

Tuplas são mais eficientes do que listas.

Por curiosidade...

```
import sys

lista = ["maçãs", "leite", "arroz", "frango", "trigo"]
tupla = ("maçãs", "leite", "arroz", "frango", "trigo")

print("Tamanho na memória da tupla:", sys.getsizeof(tupla))
print("Tamanho na memória da lista:", sys.getsizeof(lista))
```

Tamanho na memória da tupla: 80
Tamanho na memória da lista: 96

Mais sobre tuplas

Veja mais sobre tuplas na bibliografia recomendada, e em docs.python.org/3/tutorial/datastructures.html#tuples-and-sequences