



INFORMÁTICA PARA ENGENHARIA

LISTAS EM PYTHON

UMA FORMA DE IMPLEMENTAR
VETORES E MATRIZES

Prof. Dr. Daniel Caetano

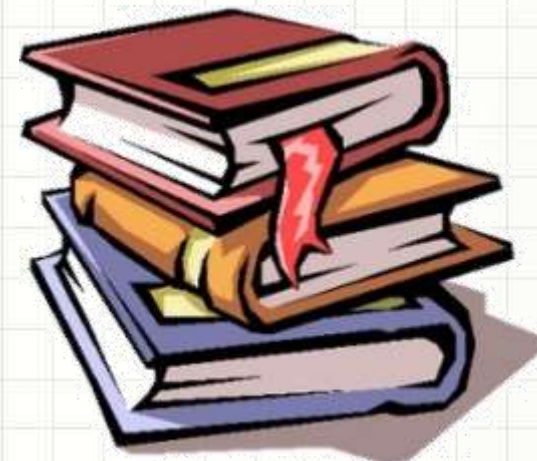
2018 - 2

Objetivos

- Compreender a ideia das listas
- Compreender o uso das listas unidimensionais: os vetores
- Compreender o uso das listas multidimensionais: as matrizes
- **Estudar para a prova!**



Material de Estudo



Material

Acesso ao Material

Notas de Aula e
Apresentação

<http://www.caetano.eng.br/>
(Informática para Engenharia – Aula 13)

Material Didático

Lógica de Programação, págs 151 a 172.

Biblioteca Virtual

“Lógica de Programação – Fundamentos da
Programação de Computadores”, págs 93 a 144.



COMPREENDENDO A SITUAÇÃO PRÁTICA

Quando usar uma lista?

- Problema: indicar empresas de um shopping

3º Piso	Ponto Trio
2º Piso	Garrefour
1º Piso	Mestlé
Térreo	Somy

```
# Lojas do Shopping
shopping_terreo = "Somy"
shopping_1o_piso = "Mestlé"
shopping_2o_piso = "Garrefour"
shopping_3o_piso = "Ponto Trio"
```

- Qual o problema dessa solução?
 - É possível usar o while/for com ela?

Quando usar uma lista?

- Problema: indicar empresas de um shopping
 - Solução: usar uma lista

3º Piso	Ponto Trio
2º Piso	Garrefour
1º Piso	Mestlé
Térreo	Somy

```
# Lojas do Shopping  
shopping = [ "Somy", "Mestlé",  
             "Garrefour", "Ponto Trio" ]
```

- Como usar isso?

Quando usar uma lista?

- Problema: indicar empresas de um shopping
 - Solução: usar uma lista

3º Piso	Ponto Trio
2º Piso	Garrefour
1º Piso	Mestlé
Térreo	Somy

```
# Lojas do Shopping
shopping = [ "Somy", "Mestlé",
             "Garrefour", "Ponto Trio" ]

print( shopping[0] )
print( shopping[1] )
print( shopping[2] )
print( shopping[3] )
```

- Como usar isso?

Quando usar uma lista?

- Problema: indicar empresas de um shopping
 - Solução: usar uma lista

3º Piso	Ponto Trio
2º Piso	Garrefour
1º Piso	Mestlé
Térreo	Somy

```
# Lojas do Shopping
shopping = [ "Somy", "Mestlé",
             "Garrefour", "Ponto Trio" ]

For X in shopping:
    print(X)
```

- Como usar isso?



O QUE É UMA LISTA PARA O PYTHON?

O que é uma lista?

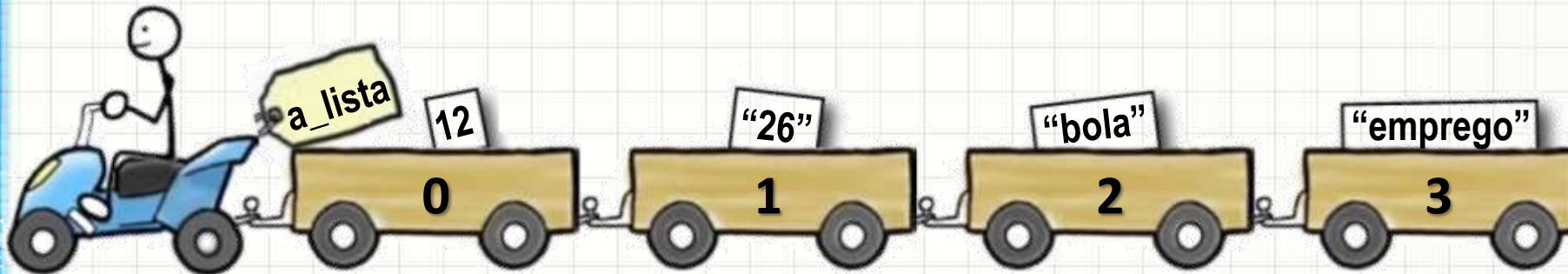
- É um conjunto de dados encadeados
 - Uma lista é como um trem de dados:

Cada vagão guarda uma informação!



O que é uma lista?

- **a_lista** é uma única variável, com vários dados
 - Os vagões são numerados



```
In[1]: print( a_lista[1] )
```

```
26
```

```
In[2]: print( a_lista[3] )
```

```
emprego
```

Criando uma lista no console

- Podemos criar uma lista vazia:

Console

```
In [1]: a_lista = [ ]
```

**Observe o
explorador de
variáveis!**

a_lista

list

0

[]

Criando uma lista no console

- Podemos criar uma lista já preenchida:

Console

```
In [2]: a_lista = [12, "26", "bola", "emprego"]
```

**Observe o
explorador de
variáveis!**

a_lista	list	4	[12, '26', 'bola', 'emprego']
---------	------	---	-------------------------------

Criando uma lista no console

- Podemos mostrar o conteúdo da lista:

Console

```
In [3]: print( a_lista )
```

**O que
aconteceu?**

Usando uma lista no console

- Podemos obter o tamanho da lista

Console

```
In [4]: N = len( a_lista )
```

```
In [5]: print( N )
```

**O que
aconteceu?**

Usando uma lista no console

- Podemos acrescentar um item na lista

Console

```
In [6]: a_lista.append( 33 )
```

**Observe o
explorador de
variáveis!**

a_lista

list

5

[12, '26', 'bola', 'emprego', 33]

Usando uma lista no console

- Podemos mudar um valor da lista

Console

```
In [7]: print ( a_lista[3] )
```

```
emprego
```

```
In [8]: a_lista[3] = "praia"
```

```
In [9]: print ( a_lista[3] )
```

```
praia
```

Usando uma lista no console

- Podemos remover um elemento da lista

Console

```
In [10]: print ( a_lista )  
[12, '26', 'bola', 'praia', 33]  
In [11]: a_lista.remove( "26" )  
In [12]: print ( a_lista )  
[12, 'bola', 'praia', 33]
```

Usando uma lista no console

- Podemos remover um elemento da lista

Console

```
In [13]: print ( a_lista )
```

```
[12, 'bola', 'praia', 33]
```

```
In [14]: a_lista.pop( 1 )
```

```
In [15]: print ( a_lista )
```

```
[12, 'praia', 33]
```

Usando uma lista no console

- Podemos percorrer os elementos da lista

Console

```
In [16]: for X in range(0,3):  
         print( a_lista[X] )
```

**O que
aconteceu?**

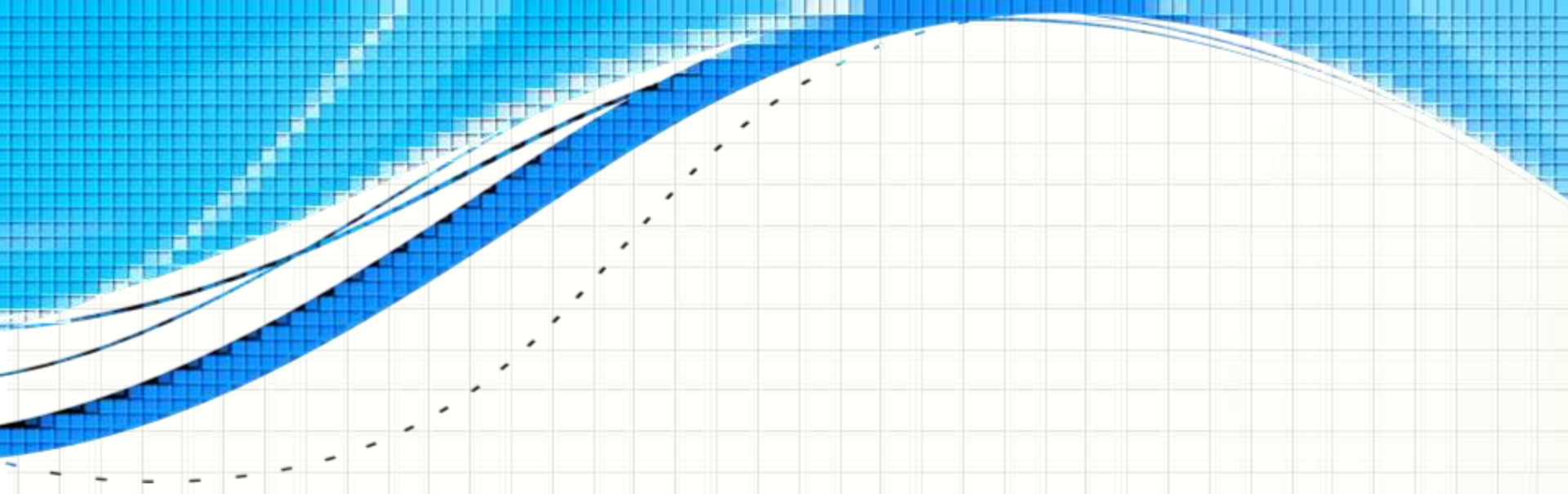
Usando uma lista no console

- Podemos percorrer os elementos da lista

Console

```
In [16]: for X in a_lista:  
         print(X)
```

**O que
aconteceu?**



ATIVIDADE

Exercício

- Complete o programa abaixo para que ele faça a média das notas da lista

```
aula13ex01.py
```

```
# Calcula a média das notas
```

```
notas = [ 8.0, 5.5, 7.5, 6.0, 3.2 ]
```

Exercício

- Complete o programa abaixo para que ele faça a média das notas da lista

```
aula13ex01.py
```

```
# Calcula a média das notas

notas = [ 8.0, 5.5, 7.5, 6.0, 3.2 ]
soma = 0
for nota in notas:
    soma = soma + nota
media = soma / len(notas)
print( "A média é %.1f." %media)
```


Exercício

- Modifique o programa abaixo para que o usuário possa digitar as 5 notas

```
aula13ex01.py
```

```
# Calcula a média das notas

notas = [ 8.0, 5.5, 7.5, 6.0, 3.2 ]
soma = 0
for nota in notas:
    soma = soma + nota
media = soma / len(notas)
print( "A média é %.1f." %media)
```

Exercício

- Modifique o programa abaixo para que o usuário possa digitar as 5 notas

aula13ex02.py

```
# Calcula a média das notas

notas = []
for nota in range(5):
    nota = float( input("Digite uma nota: ") )
    notas.append( nota )

soma = 0
for nota in notas:
    soma = soma + nota
media = soma / len(notas)
print( "A média é %.1f." %media)
```

Exercício

- Modifique o programa abaixo para que, após imprimir a média, imprima as notas em ordem inversa de digitação

aula13ex02.py

```
# Calcula a média das notas

notas = []
for nota in range(5):
    N = int( input("Digite uma nota: ") )
    notas.append( nota )

soma = 0
for nota in notas:
    soma = soma + nota
media = soma / len(notas)
print( "A média é %.1f" %media)
```

Exercício

- Modifique o programa abaixo para que, após imprimir a média, imprima as notas em ordem inversa de digitação

aula13ex03.py

```
# Calcula a média das notas

notas = []
for nota in range(5):
    N = int( input("Digite uma nota: ") )
    notas.append( nota )

soma = 0
for nota in notas:
    soma = soma + nota
media = soma / len(notas)
print( "A média é %.1f" %media)
```

aula13ex03.py (cont)

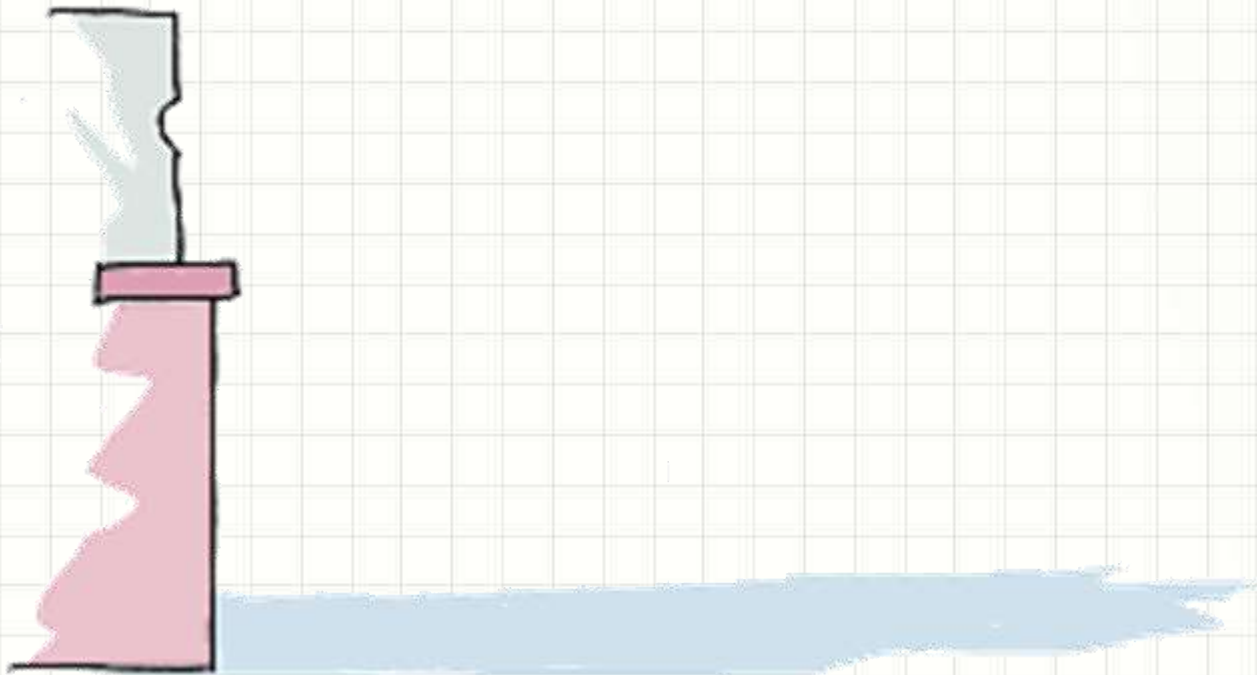
```
for X in range(4, -1, -1):
    print( notas[X] )
```



LISTAS: FILAS E PILHAS

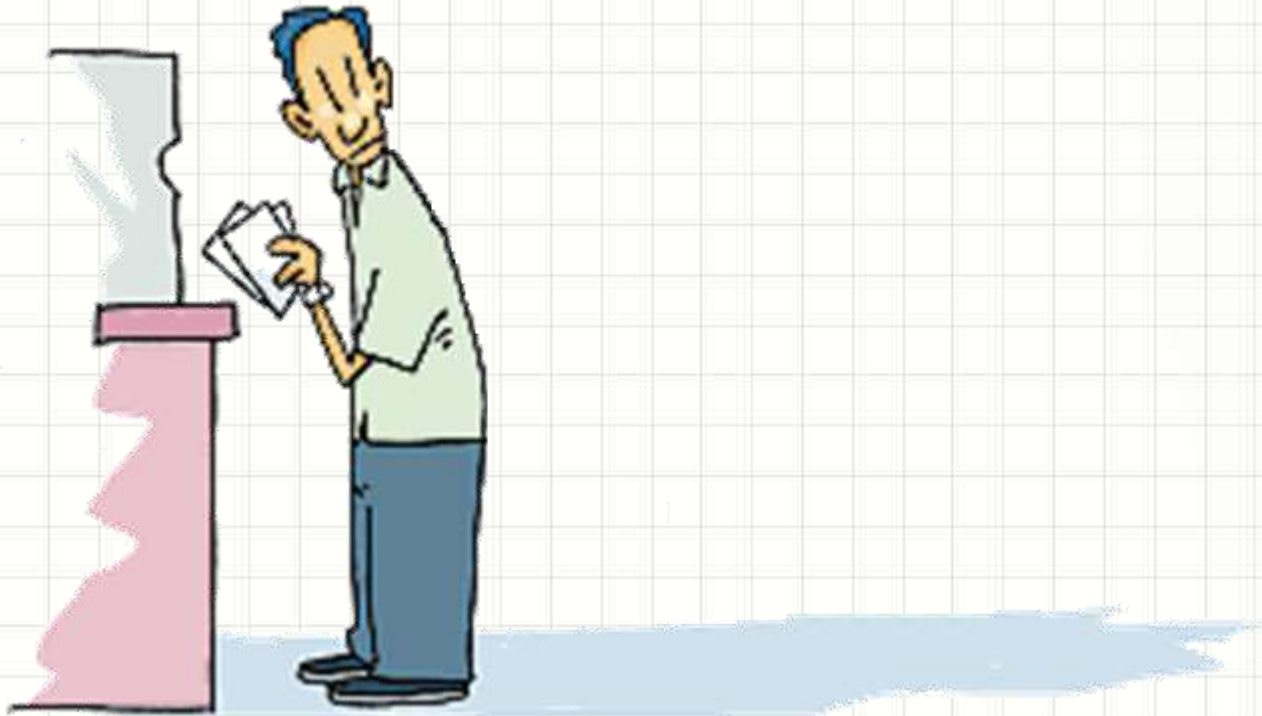
Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



FIFO:
First In
First Out

Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplos: banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila de atendimento



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do ba



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Muito usada em simulações

Implementando Filas

- Comandos `append()` e `pop(0)`
 - Observe o uso:

`aula13ex04.py`

```
# Pessoas chegando na fila e sendo atendidas
fila = []
fila.append("Alberto") # Chegou Alberto
fila.append("Fernanda") # Chegou Fernanda
print( fila.pop(0) ) # Alberto foi atendido
fila.append("Renato") # Chegou Renato
print( fila.pop(0) ) # Fernanda foi atendida
print( fila.pop(0) ) # Renato foi atendido
```

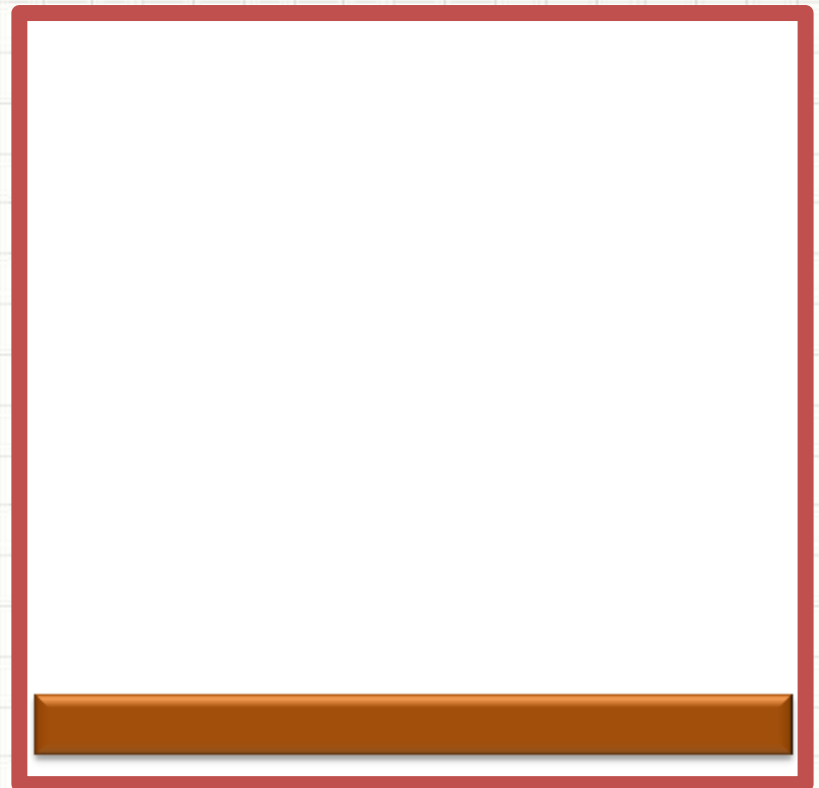
Pilha: Último a Entrar é 1º a Sair (LIFO)

- Comum: ordem de remoção → armazenamento
- Exemplo: organizando entregas de Pizza



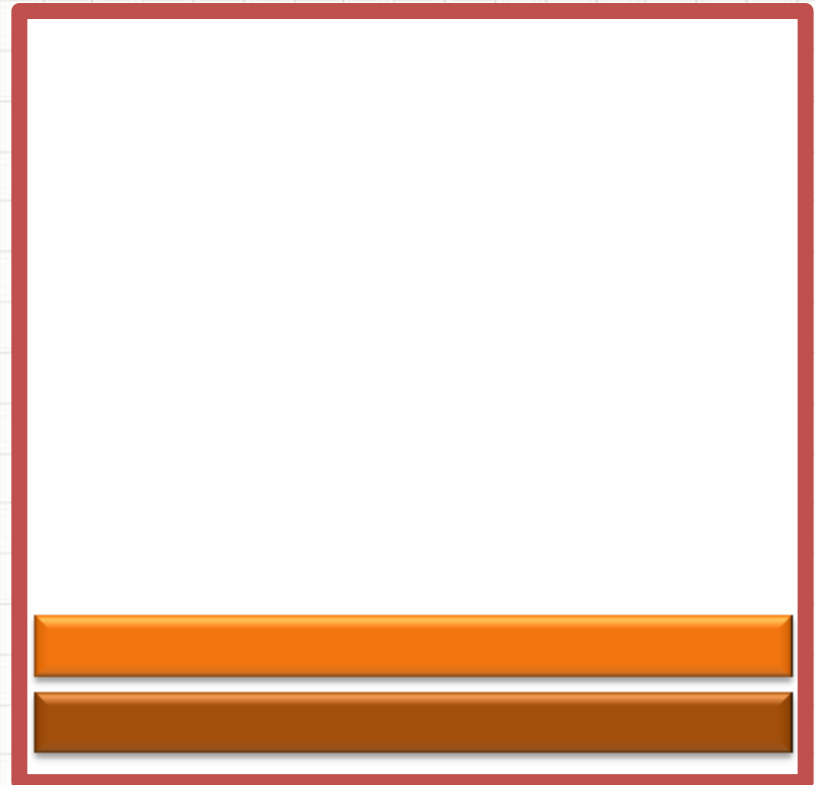
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



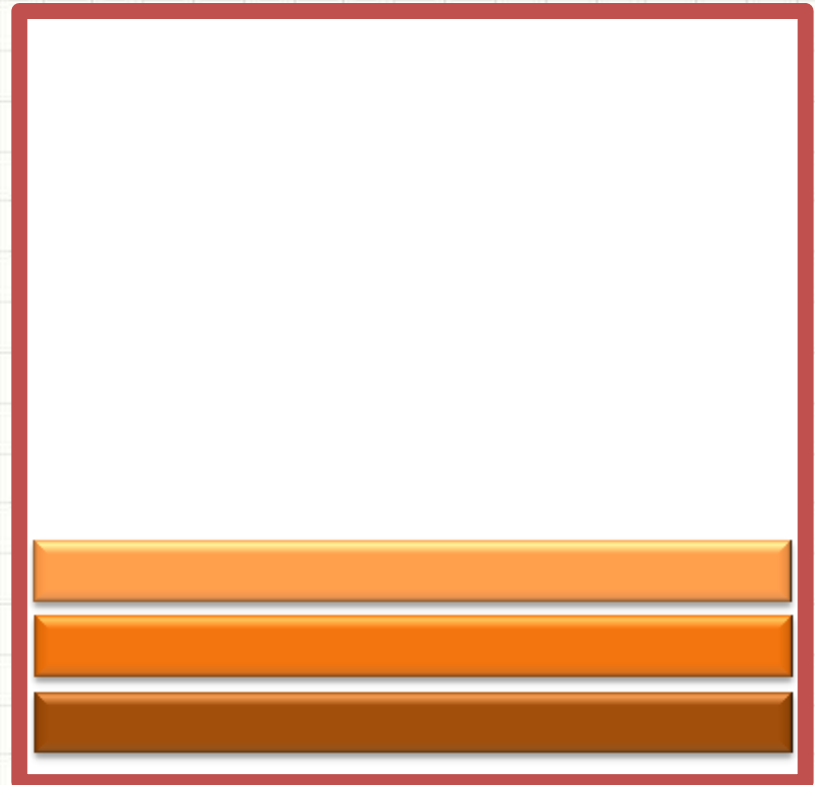
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



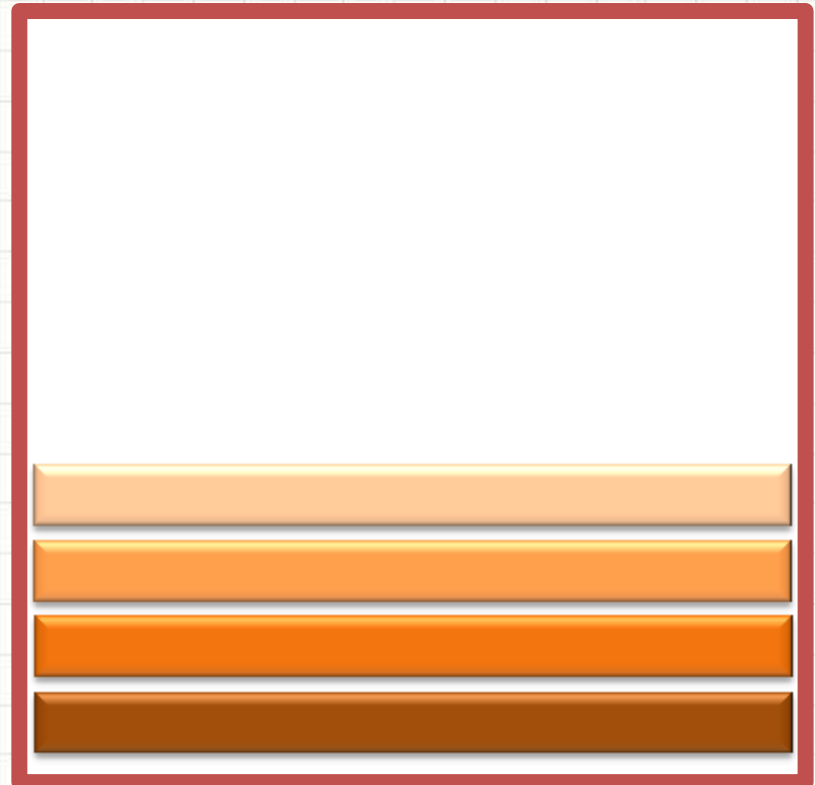
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



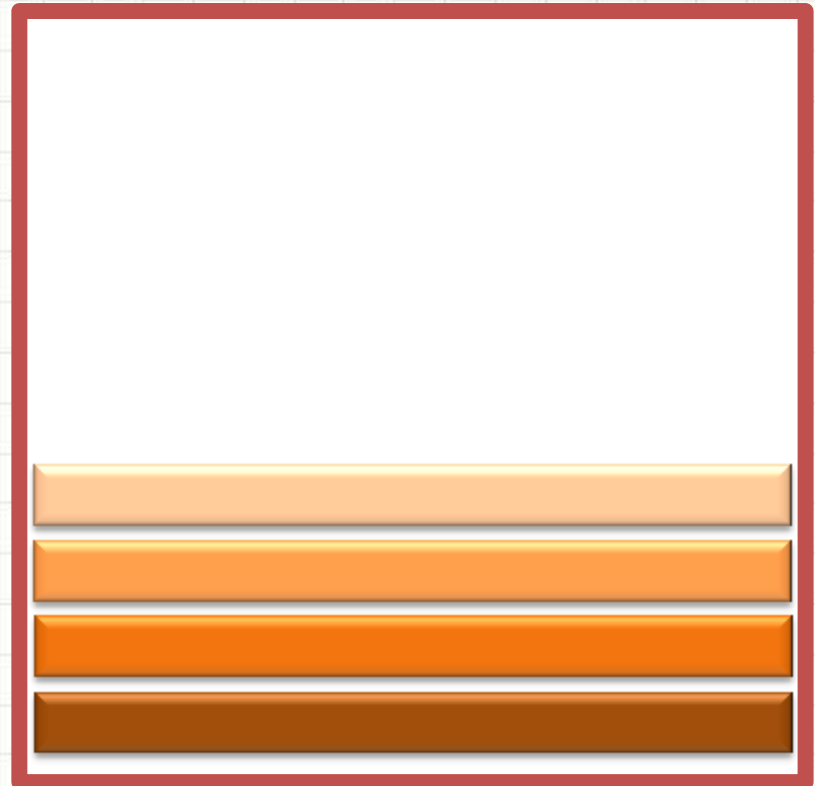
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



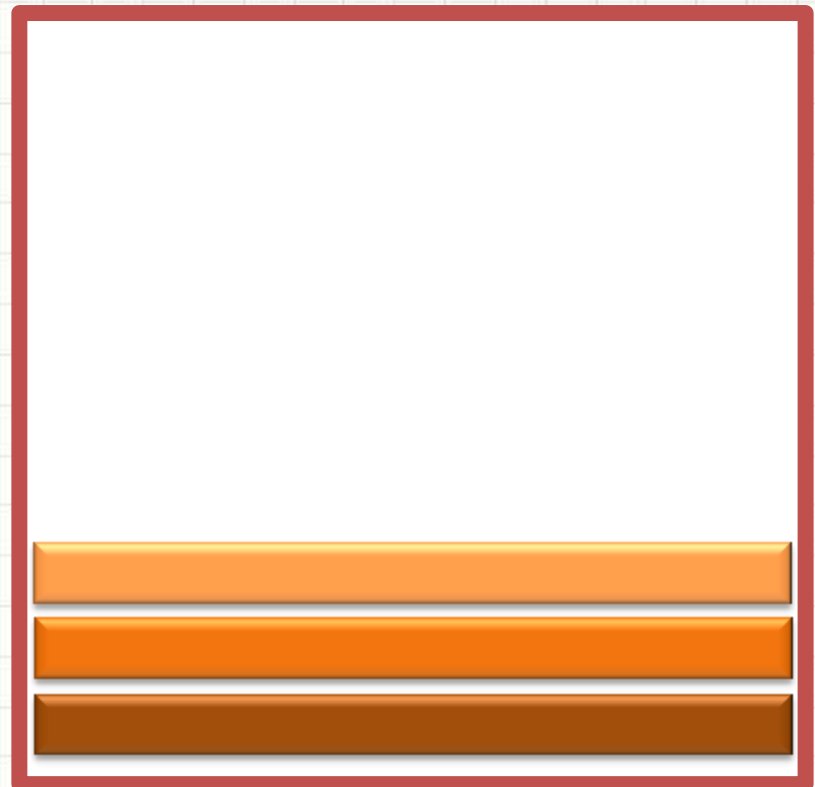
Pilha: Último a Entrar é 1º a Sair (LIFO)

- E nas entregas?



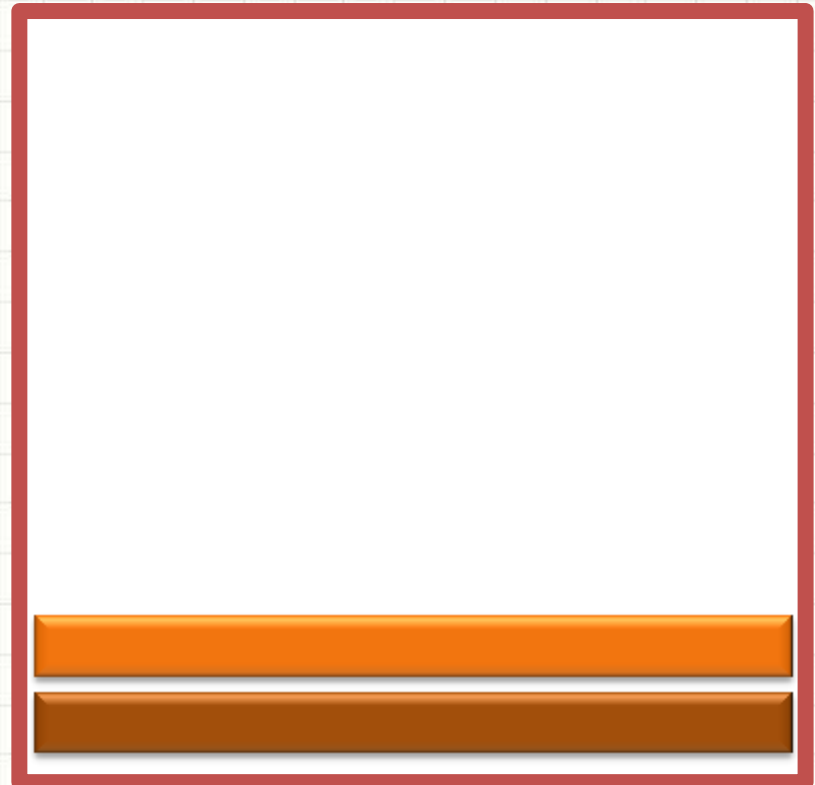
Pilha: Último a Entrar é 1º a Sair (LIFO)

- E nas entregas?



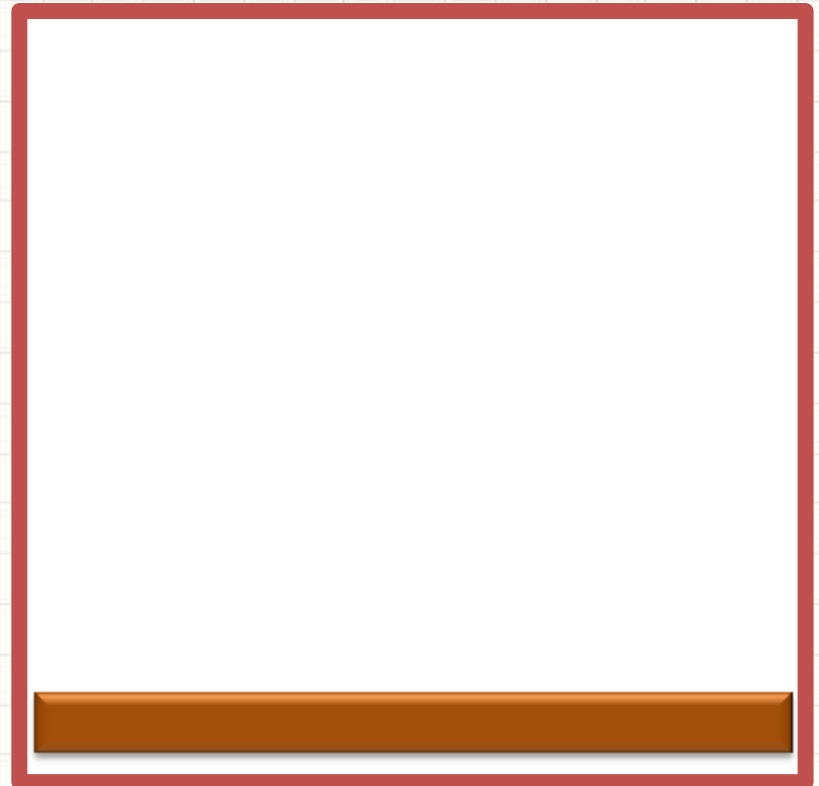
Pilha: Último a Entrar é 1º a Sair (LIFO)

- E nas entregas?



Pilha: Último a Entrar é 1º a Sair (LIFO)

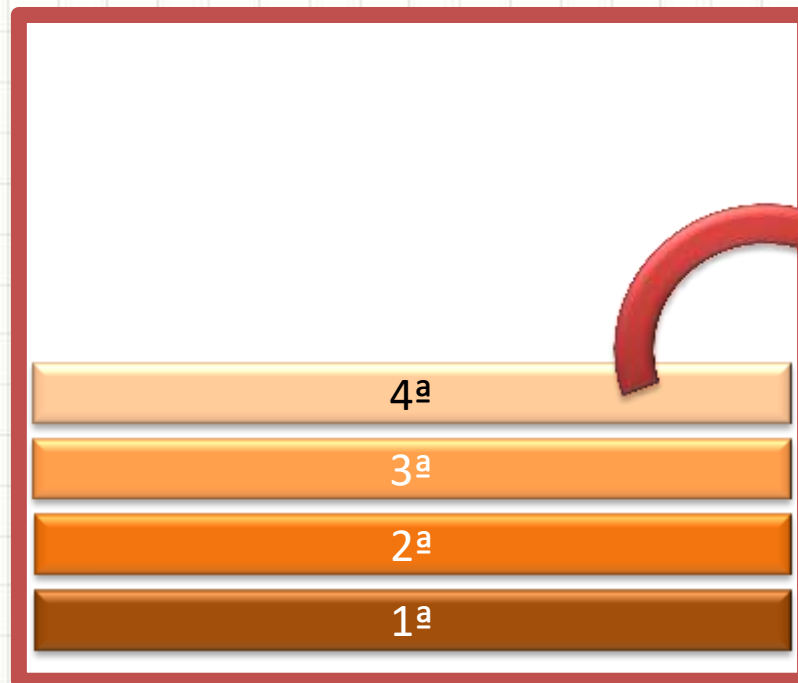
- E nas entregas?



Pilha: Último a Entrar é 1º a Sair (LIFO)

- Observe...
 - A última pizza a entrar...
 - Será a primeira a sair...

LIFO:
Last In
First Out

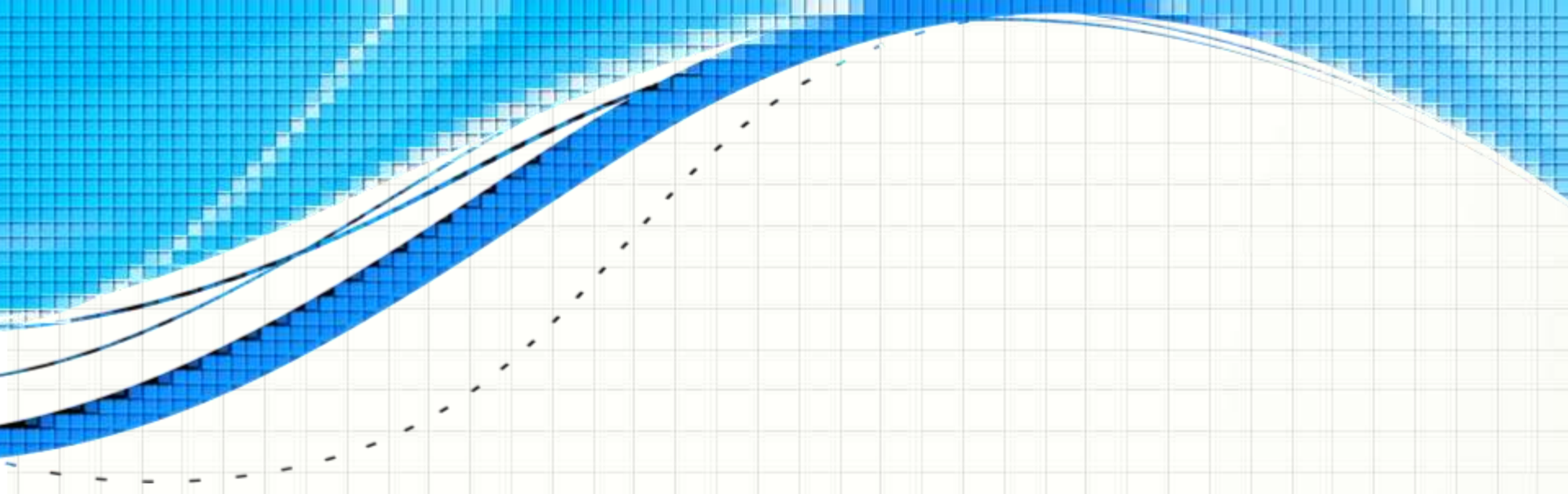


Implementando Pilhas

- Comandos `append()` e `pop(-1)`
 - Observe o uso:

`aula13ex05.py`

```
# Colocando Pizzas na Pilha
pilha = []
pilha.append("Calabresa")
pilha.append("Mussarela")
pilha.append("Pepperoni")
print( pilha.pop(-1) )
print( pilha.pop(-1) )
print( pilha.pop(-1) )
```



LISTAS DE LISTAS

TAMBÉM CONHECIDAS COMO MATRIZES

Listas de Listas

- Podemos ter uma lista de listas
 - Cada elemento da lista... É outra lista

Console

```
In [17]: matriz = [ [ 0, 1, 2 ],  
                   [ 3, 4, 5 ],  
                   [ 6, 7, 8 ] ]
```

```
In [18]: print( matriz )
```

O que
aconteceu?

Listas de Listas

- Podemos imprimir uma linha da lista

Console

```
In [19]: print( matriz[0] )
```

**O que
aconteceu?**

Listas de Listas

- Podemos imprimir um elemento da lista

Console

```
In [20]: print( matriz[0][2] )
```

**O que
aconteceu?**

Listas de Listas

- Podemos percorrer as linhas da matriz

Console

```
In [21]: for X in matriz:  
         print( X )
```

**O que
aconteceu?**

Listas de Listas

- Podemos percorrer todos os elementos da matriz

Console

```
In [22]: for linha in matriz:  
         for coluna in linha:  
             print( coluna )
```

**O que
aconteceu?**

Listas de Listas

- Melhorando o visual...
 - Teste essa variação!

Console

```
In [23]: for linha in matriz:  
         for coluna in linha:  
             print( coluna )  
         print("----")
```

**O que
aconteceu?**

Exemplo

- Crie a agenda abaixo e faça com que ela seja impressa formatada:

Nome: Daniel

Telefone: 11-5555-1234

Data Nasc.: 10/02/1973

aula13ex04.py

```
# Imprime agenda
```

```
agenda = [ ["Daniel", "11-5555-1234", "10/02/1973"],  
            ["João", "11-5555-2345", "02/10/1980"],  
            ["Alberto", "11-5555-3456", "11/01/1987"]]
```

Exemplo

- Crie a agenda abaixo e faça com que ela seja impressa formatada:

Nome: Daniel
Telefone: 11-5555-1234
Data Nasc.: 10/02/1973

aula13ex04.py

```
# Imprime agenda
```

```
agenda = [ ["Daniel", "11-5555-1234", "10/02/1973"],  
           ["João", "11-5555-2345", "02/10/1980"],  
           ["Alberto", "11-5555-3456", "11/01/1987"]]
```

```
for contato in agenda:
```

```
    print("%12s %s" %("Nome:", contato[0]))  
    print("%12s %s" %("Telefone:", contato[1]))  
    print("%12s %s" %("Data Nasc.:", contato[2]))  
    print("-----")
```

Exemplo – Alternativa de Impressão

- Crie a agenda abaixo e faça com que ela seja impressa formatada:

Nome: Daniel
Telefone: 11-5555-1234
Data Nasc.: 10/02/1973

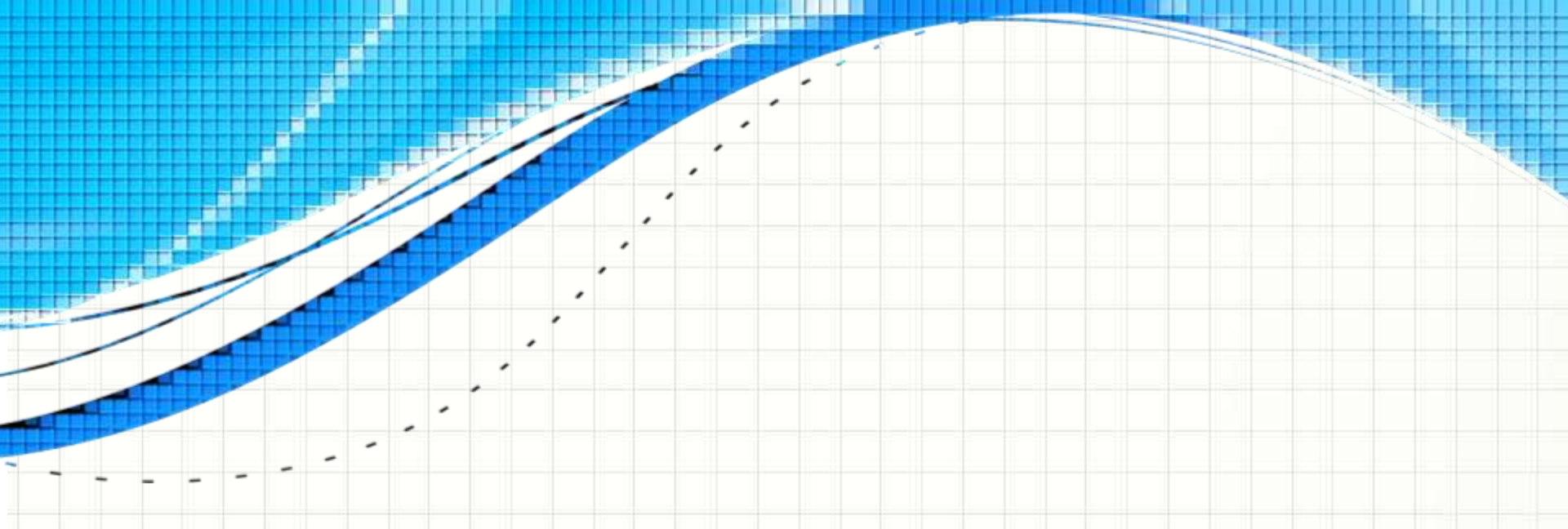
aula13ex06a.py

```
# Imprime agenda

agenda = [ ["Daniel", "11-5555-1234", "10/02/1973"],
           ["João", "11-5555-2345", "02/10/1980"],
           ["Alberto", "11-5555-3456", "11/01/1987"]]

for contato in agenda:
    print("{:>12} {}".format("Nome:", contato[0]))
    print("{:>12} {}".format("Telefone:", contato[1]))
    print("{:>12} {}".format("Data Nasc.:", contato[2]))
    print("-----")
```

Não precisa indicar os tipos!



MATRIZES USANDO NUMPY

Matrizes com Numpy

- Matrizes padrão não permitem contas diretas
 - Observe:

`aula13ex07.py`

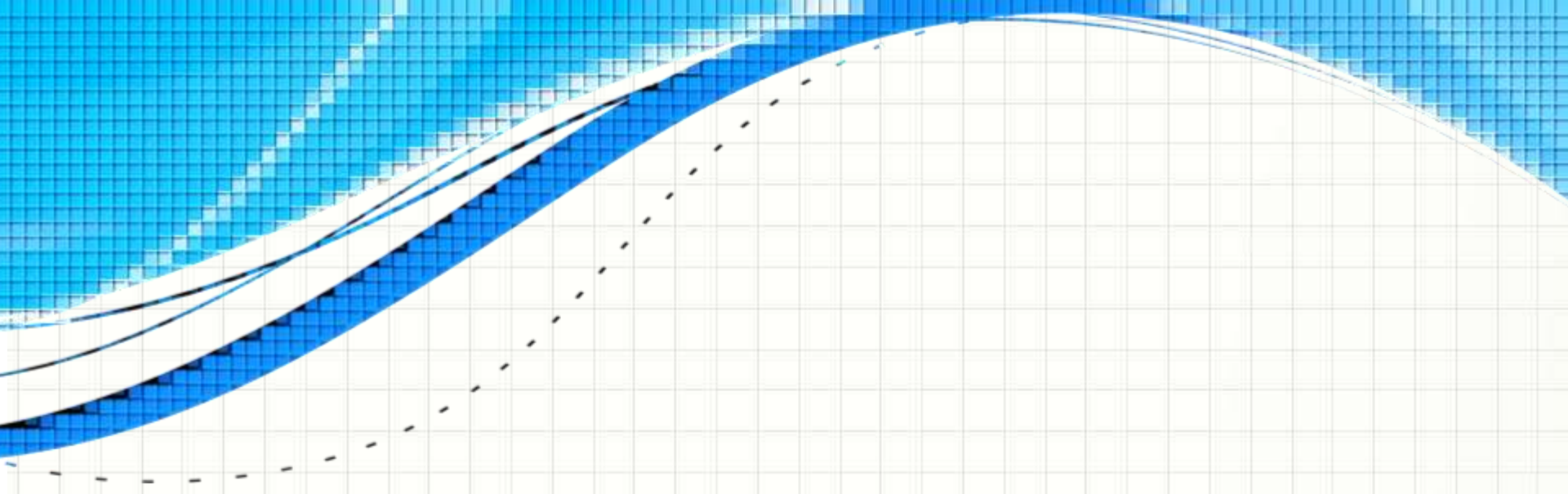
```
# Operações com matrizes
x = [[1,2] , [3,4]]
y = [[5,6] , [7,8]]
k = 2 * x
print(k)
z = k + y
print(z)
```

Matrizes com Numpy

- Numpy: biblioteca para operação com matrizes
 - Observe o uso:

`aula13ex08.py`

```
# Operações com matrizes com Numpy
import numpy
x = numpy.array([[1,2] , [3,4]])
y = numpy.array([[5,6] , [7,8]])
k = 2 * x
print(k)
z = k + y
print(z)
```



CONCLUSÕES

Resumo

- Listas...
 - Podem ser usadas para vetores ou matrizes
 - São muito úteis!
 - Agrupar diferentes dados em única variável!
 - Não deixe de praticar!
-
- Prepare-se para a prova!



PERGUNTAS?