



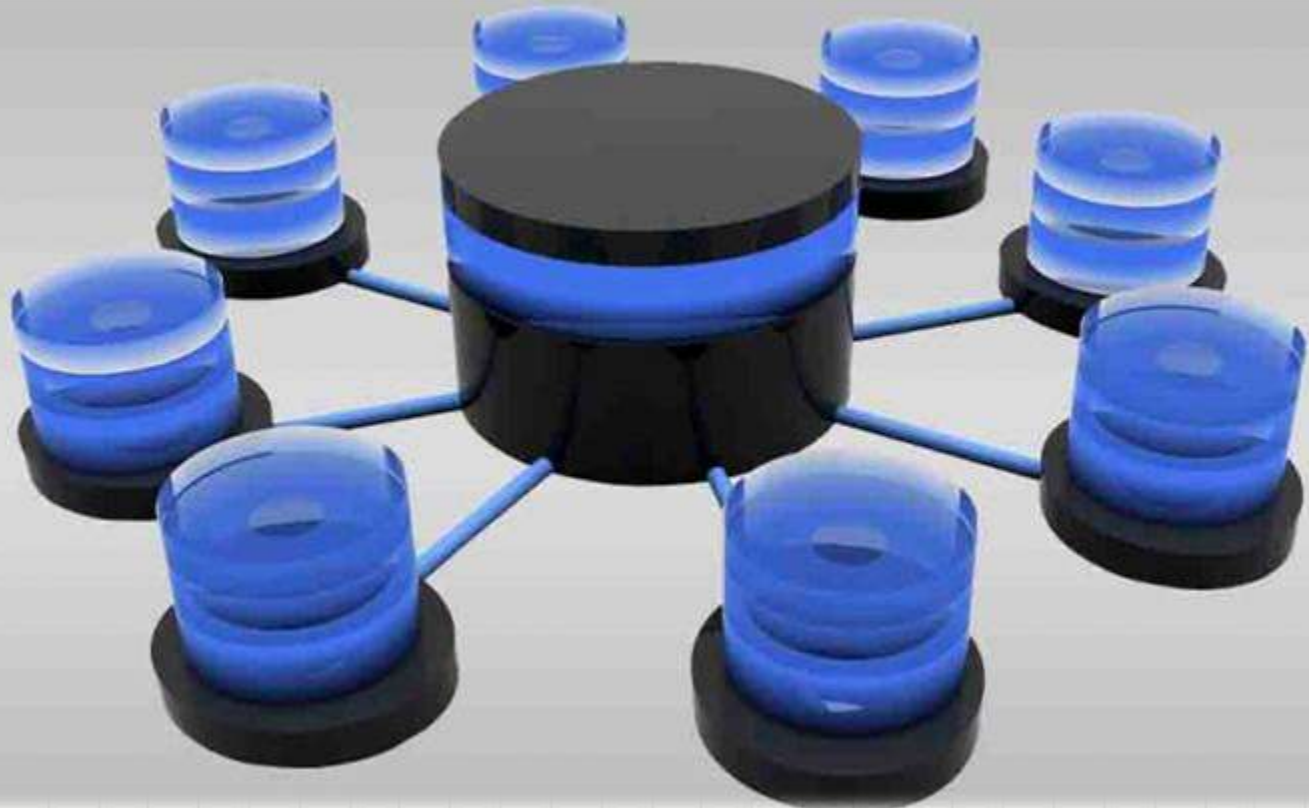
# **INFORMÁTICA PARA ENGENHARIA**

## **LISTAS E VETORES**

Prof. Dr. Daniel Caetano

2019 - 1

# Necessidade comum: guardar dados relacionados juntos



# Em uma lista de contatos, por exemplo...



# Como implementar?



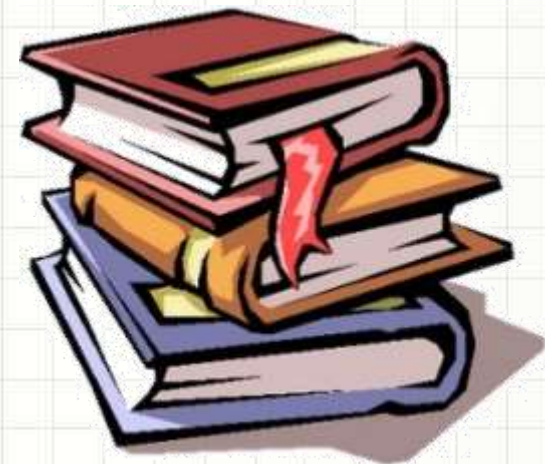
# Objetivos

- Compreender a ideia das listas
- Compreender o uso das listas unidimensionais: os vetores

- **Exercícios Aula 13!**



# 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.

Material Adicional

[https://www.ime.usp.br/~leo/mac2166/2017-1/introducao\\_vetores\\_python.html](https://www.ime.usp.br/~leo/mac2166/2017-1/introducao_vetores_python.html)

---

**LEMBRETE: CONSULTAR O “DEPOIS” DA AULA 13 NO SAVA!**



# **COMPREENDENDO A SITUAÇÃO PRÁTICA**

# Quando usar uma lista?

- Problema: ler 3 nomes e listar aqueles cujo número de letras é maior que a média

```
# Lista nomes maiores que a média
NOME1 = input("Digite 1º nome: ")
NOME2 = input("Digite 2º nome: ")
NOME3 = input("Digite 3º nome: ")
MED = (len(NOME1) + len(NOME2) + len(NOME2))/3
if len(NOME1) > MED :
    print (NOME1)
if len(NOME2) > MED :
    print (NOME2)
if len(NOME3) > MED :
    print (NOME3)
```

- Qual o problema dessa solução?
  - E se fossem 50 nomes?

**Solução: Listas**





# O QUE É UMA LISTA PARA O PYTHON?

# O que é uma lista?

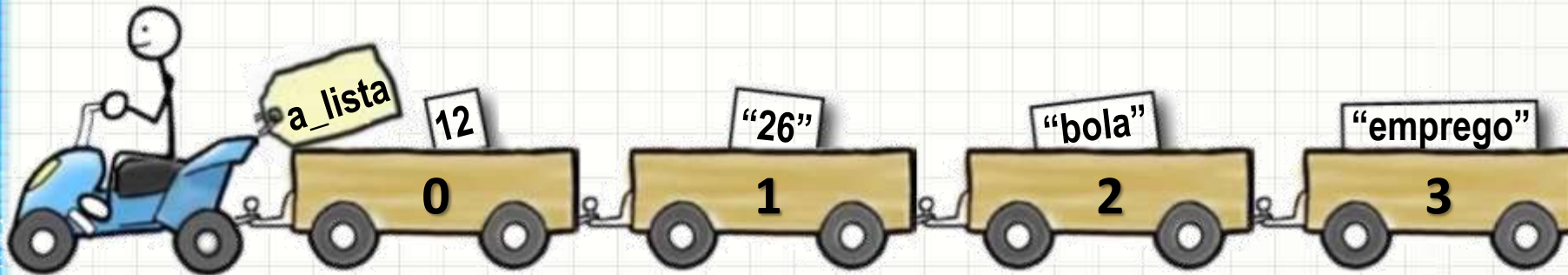
- É um conjunto de dados encadeados
  - Uma lista é como um trem de dados:

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
```

As listas são também chamadas de **vetores**

# 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 juntar listas

## Console

```
In [16]: print ( a_lista )
```

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

```
In [17]: b_lista = [ "Mais", "Elementos" ]
```

```
In [18]: a_lista = a_lista + b_lista
```

```
In [19]: print ( a_lista )
```

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

# Usando uma lista no console

- Podemos percorrer os elementos da lista

Console

```
In [20]: for N in range(0,5):  
         print( a_lista[N] )
```

**O que  
aconteceu?**

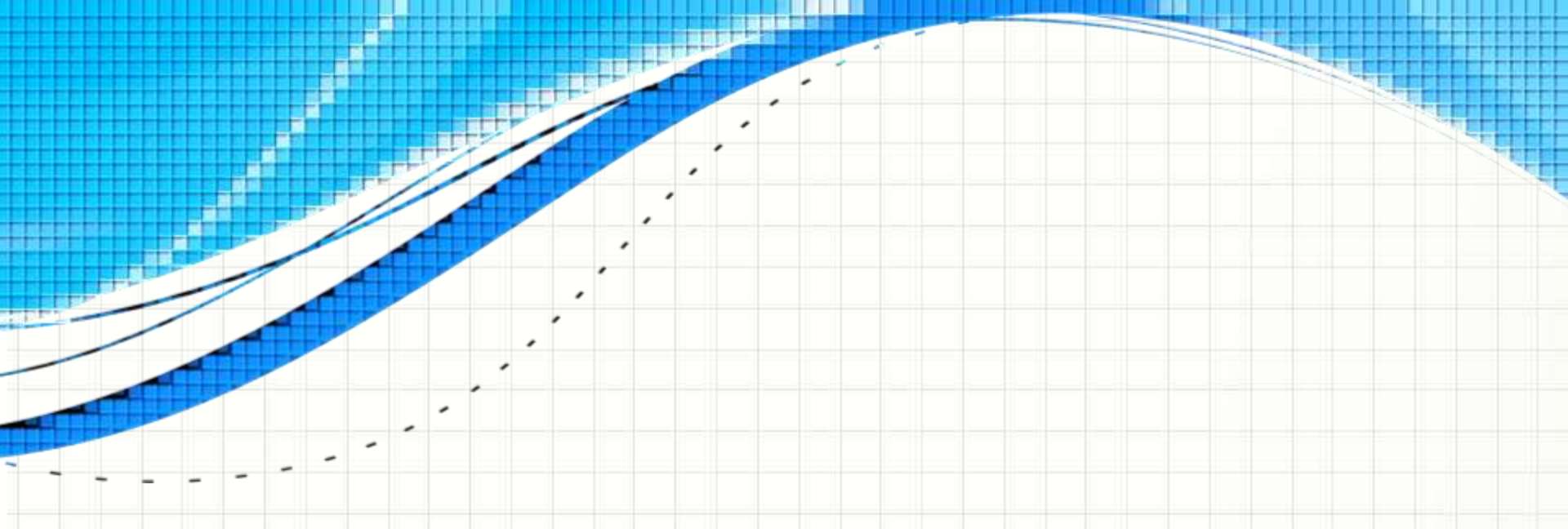
# Usando uma lista no console

- Podemos percorrer os elementos da lista

Console

```
In [21]: for EL in a_lista:  
         print(EL)
```

**O que  
aconteceu?**



# EXERCÍCIOS

# 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}.".format(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}.".format(MEDIA))
```

# Exercício

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

```
aula13ex02.py
```

```
# Calcula a média das notas

NOTAS = []
for N 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}.".format(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 N 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}.".format(MEDIA))
```

# Exercício

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

aula13ex03.py

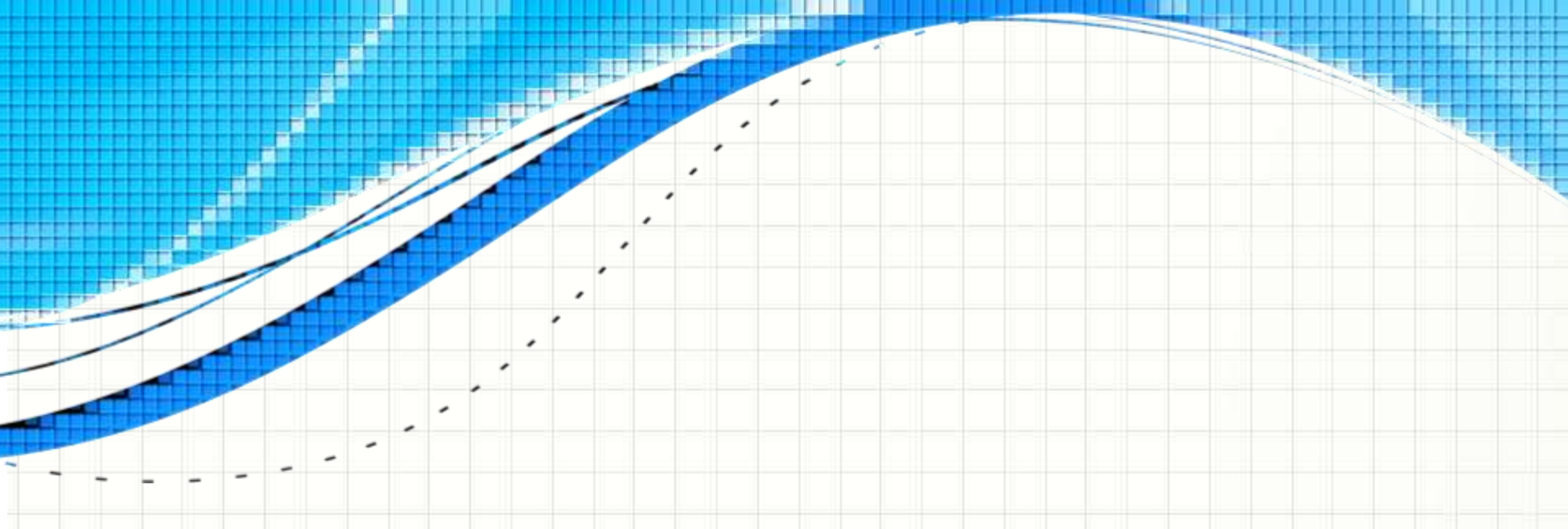
```
# Calcula a média das notas

NOTAS = []
for N in range(5):
    NOTA = float( input("Digite nota: ") )
    NOTAS.append( NOTA )

SOMA = 0
for NOTA in NOTAS:
    SOMA = SOMA + NOTA
MEDIA = SOMA / len(NOTAS)
print( "Média: {:.1f}.".format(MEDIA))
```

aula13ex03.py (cont)

```
for N in range(4, -1, -1):
    print( NOTAS[N] )
```



**VOLTANDO AO  
PROBLEMA ORIGINAL**

# Problema Original

- Problema: ler 3 nomes e listar aqueles cujo número de letras é maior que a média

```
# Lista nomes maiores que a média
NOME1 = input("Digite 1º nome: ")
NOME2 = input("Digite 2º nome: ")
NOME3 = input("Digite 3º nome: ")
MEDIA = (len(NOME1) + len(NOME2) + len(NOME2))/3
if len(NOME1) > MEDIA :
    print (NOME1)
if len(NOME2) > MEDIA :
    print (NOME2)
if len(NOME3) > MEDIA :
    print (NOME3)
```

- Como resolver usando listas?

# Problema Original – Com Lista

```
# Lista nomes maiores que a média
NOMES = []
# Lê os nomes
for N in range(3) :
    NOME = input("Digite um nome: ")
    NOMES.append( NOME )
# Computa o tamanho médio dos nomes
TOTAL = 0
for NOME in NOMES :
    TOTAL = TOTAL + len(NOME)
MEDIA = TOTAL/len(NOMES)
# Imprime aqueles cujo tamanho é maior que a média
for NOME in NOMES :
    if len(NOME) > MEDIA :
        print (NOME)
```

## Melhorando?



# Problema Original – Com Lista

```
# Lista nomes maiores que a média
NOMES = []
# Lê os nomes
for N in range(3) :
    NOME = input("Digite o " + str(N+1) + "º nome: ")
    NOMES.append( NOME )
# Computa o tamanho médio dos nomes
TOTAL = 0
for NOME in NOMES :
    TOTAL = TOTAL + len(NOME)
MEDIA = TOTAL/len(NOMES)
# Imprime aqueles cujo tamanho é maior que a média
for NOME in NOMES :
    if len(NOME) > MEDIA :
        print (NOME)
```

**E para 10 nomes?**

# Problema Original – Com Lista

```
# Lista nomes maiores que a média
NOMES = []
# Lê os nomes
for N in range(10) :
    NOME = input("Digite o " + str(N+1) + "º nome: ")
    NOMES.append( NOME )
# Computa o tamanho médio dos nomes
TOTAL = 0
for NOME in NOMES :
    TOTAL = TOTAL + len(NOME)
MEDIA = TOTAL/len(NOMES)
# Imprime aqueles cujo tamanho é maior que a média
for NOME in NOMES :
    if len(NOME) > MEDIA :
        print (NOME)
```

# Problema Alternativo

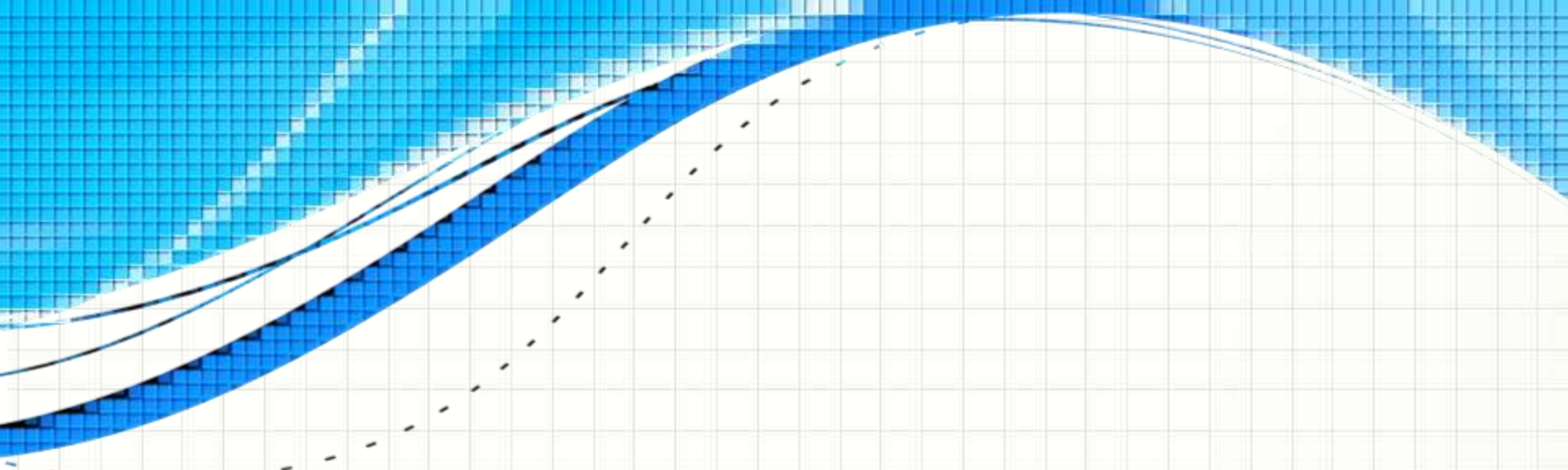
- Problema: ler 5 nomes e listá-los em ordem alfabética:

```
# Lista em ordem alfabética
NOMES = []
# Lê os nomes
for N in range(5) :
    NOME = input("Digite o " + str(N+1) + "º nome: ")
    NOMES.append( NOME )
# Mostra os nomes em ordem alfabética...?
```

# Problema Alternativo

- Problema: ler 5 nomes e listá-los em ordem alfabética:

```
# Lista em ordem alfabética
NOMES = []
# Lê os nomes
for N in range(5) :
    NOME = input("Digite o " + str(N+1) + "º nome: ")
    NOMES.append( NOME )
# Ordena os nomes
NOMES.sort()
# Mostra os nomes
for NOME in NOMES :
    print (NOME)
```



# EXERCÍCIO

# Exercício

- Faça um programa que leia os nomes e as médias de 10 alunos e, **depois**, liste os nomes dos alunos aprovados (média  $\geq 6.0$ )

# Exercício

- Faça um programa que leia os nomes e as médias de 10 alunos e, **depois**, liste os nomes dos alunos aprovados (média  $\geq 6.0$ )

aula13ex04.py

```
# Lê nomes e médias
```

```
NOMES = []
```

```
MEDIAS = []
```

```
for N in range(10) :
```

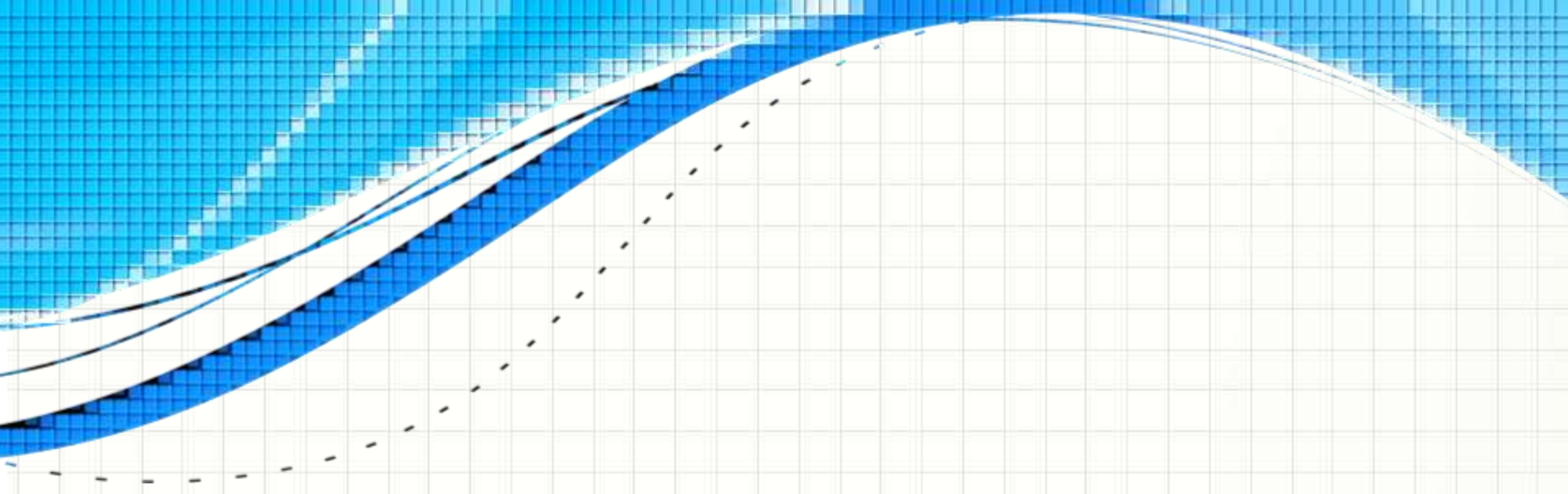
```
    NOMES.append(input("Digite o " + str(N+1) + "º nome: "))
```

```
    MEDIAS.append(input("Digite a " + str(N+1) + "ª média: "))
```

```
for N in range(10) :
```

```
    if MEDIAS[N]  $\geq$  6.0 :
```

```
        print(NOMES[N])
```



# CONCLUSÕES



# Resumo

- Listas...
  - Pode guardar muitos elementos!
  - Podemos usar para representar vetores!
  - Agrupam diferentes dados em única variável!

**SAVA13!**

- 
- Estruturas e tipos abstratos

- 
- Matrizes...
    - Como usá-las em Python?



**PERGUNTAS?**

# Exercício para Casa

- Faça um programa que leia 5 valores inteiros em uma lista, pergunte um outro valor para o usuário e, finalmente, indique a primeira posição em que o valor aparece na lista. Se o valor não aparecer, deve imprimir “Valor não aparece na lista!”
- Exemplo: para a lista [ 1, 10, 7, 28, 45 ]
- Se o usuário informar o valor 7
- O programa deve apresentar a posição “2”