

# RACIOCÍNIO ALGORÍTMICO

## LISTAS E VETORES

Professora:  
Lisiane Reips

(slides desenvolvidos pelos Profs. Giulio e  
Holisses e adaptado e atualizado por Lisiane)

# LISTAS

## Contextualizando...

1) Escreva um programa que registre o nome dos alunos de uma turma. A turma tem 3 alunos. O programa deve ler o nome de cada aluno e registrar.

```
aluno1 = input("Informe seu nome: ")
aluno2 = input("Informe seu nome: ")
aluno3 = input("Informe seu nome: ")

print("Os alunos da turma são: \n %s \n %s \n %s" %(aluno1,aluno2,aluno3))
```

# LISTAS

```
lista.py  X
C: > Users > User > Desktop > lista.py
1  # Lista para armazenar os nomes dos alunos
2  nomes_alunos = []
3
4  # Loop para ler o nome de 3 alunos
5  for i in range(3):
6      nome = input(f"Digite o nome do aluno {i + 1}: ")
7      nomes_alunos.append(nome)
8
9  # Exibindo os nomes registrados
10 print("\nNomes dos alunos registrados:")
11 for nome in nomes_alunos:
12     print(nome)
```

# LISTAS

`range(3)` gera uma sequência de números: 0, 1, 2 (três números no total).

O `.append()` é um método das listas que adiciona um elemento no final da lista.

# LISTAS

```
for i in range(3):  
    print(i)
```

# LISTAS

```
lista.py  X
C: > Users > User > Desktop > lista.py > ...
1  # Lista para armazenar os nomes dos alunos
2  nomes_alunos = []
3
4  # Loop para ler o nome de 3 alunos
5  for i in range(3):
6      nome = input(f"Digite o nome do aluno {i + 1}: ")
7      nomes_alunos.append(nome)
8
9  # Exibindo os nomes registrados
10 print("\nNomes dos alunos registrados:")
11 for nome in nomes_alunos:
12     print(nome)
13
14 # Mostrando o segundo nome da lista
15 print("\nSegundo nome da lista:", nomes_alunos[1])
```

# LISTAS

## Contextualizando...

Variando o requisito...

1) Construa um programa que registre o nome dos alunos de uma turma. A turma pode variar a quantidade de alunos. O programa deve ler a quantidade de alunos da turma e ler o nome de cada aluno e registrar.

# LISTAS

## *O que são Listas?*

1. Uma lista é um conjunto de valores ordenados.
2. Os valores são identificados por índices e chamamos de elementos.
3. Uma lista pode conter zero ou mais elementos de qualquer tipo (em Python usa-se tipagem dinâmica)
4. A medida que novos elementos são adicionados ou removidos o tamanho da lista cresce ou diminui.

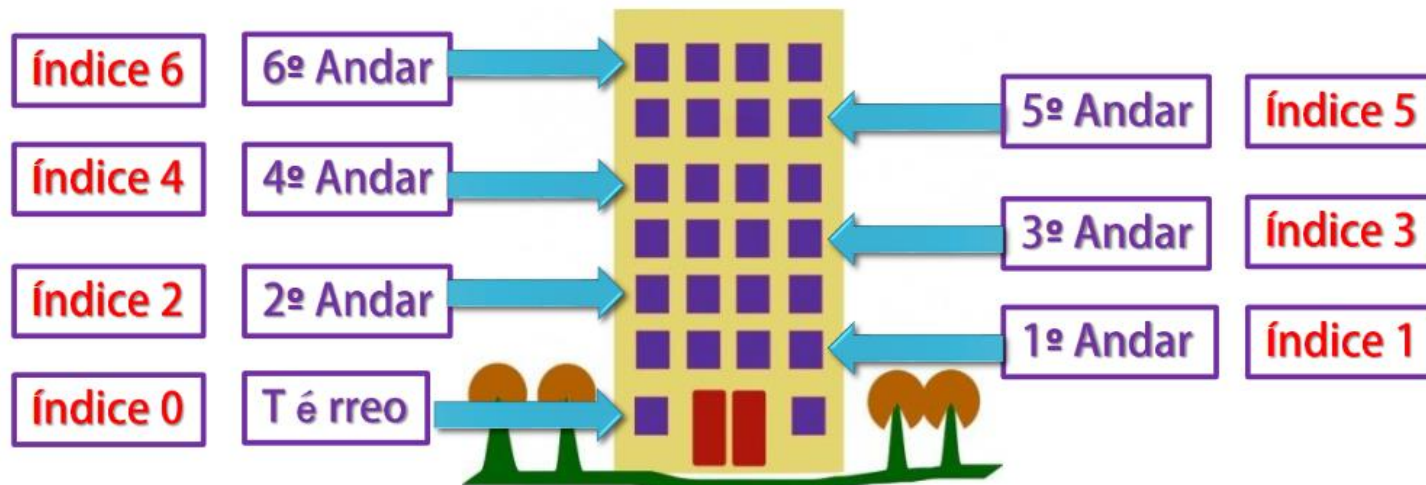
Exemplo: [10,20,30,40,50,60,70,80,90,100]



# LISTAS

## *O que são Índices?*

1. Índice é uma referência a posição do elemento dentro da lista.
2. Os índices se iniciam em zero.
3. Imagine uma lista como um edifício de apartamentos, onde o térreo é o andar zero, o primeiro andar é o 1, conforme o exemplo abaixo:



# LISTAS


## ***Declaração de Listas***

1. Elementos envolvidos por [ ] (Colchetes).
2. Declaração e inicialização de uma **LISTA VAZIA**:
  1. exemplo = []
3. Declaração e inicialização de uma lista com 5 elementos:
  1. exemplo = [10,50,5,2,100]
  2. exemplo = ["algoritmo","é","muito","bom","e fácil"]
4. Declaração e inicialização de uma **lista sequencial com o range**:
  1. minhaLista = range(10)
5. Declaração e inicialização de uma lista **ANINHADA**:
  1. suaLista = [10,50,5,[2,100],"texto",3.0]

# LISTAS

## *Acesso aos elementos de uma lista*

1. A forma de acesso a cada elemento de uma lista é através do índices (endereços) de cada elemento:




```
suaLista = [10,50,5,2,100]
print(suaLista[0])
print(suaLista[1])
print(suaLista[2])
print(suaLista[3])
print(suaLista[4])
```

```
>>> ===== RESTART =====
>>>
10
50
5
2
100
>>> |
```

# LISTAS

## *Acesso aos elementos de uma lista*

1. O acesso também pode se dar através de um loop (while):



```
suaLista = [10,50,5,2,100]
i=0
while i<5:
    print(suaLista[i])
    i=i+1
```

```
>>> ===== RESTART =====
>>>
10
50
5
2
100
>>> |
```

# LISTAS

## *Acesso aos elementos de uma lista*

1. O acesso também pode se dar através de um loop (for):

- **# Uma nova lista: lista de frutas**

```
lista = ['Caju', 'Laranja', "Banana", 'Uva']
```

- **# Varrendo a lista inteira**

```
for fruta in lista:  
    print(fruta)
```

```
Caju  
Laranja  
Banana  
Uva
```

# LISTAS

## ***Alguns pontos importantes:***

Temos aspas simples e duplas na mesma lista, você pode decidir quando usar uma ou outras, mas:

1. Use aspas simples se a string tiver aspas duplas dentro:
  - o mensagem = 'Ela disse: "Oi!"'
2. Use aspas duplas se a string tiver aspas simples dentro:
  - o mensagem = "A caixa d'água está vazia."

# LISTAS


## ***Alguns pontos importantes:***

O que significa *for fruta in lista*?

Fruta é uma variável temporária criada dentro do for para armazenar cada item da lista enquanto ela é percorrida.

# LISTAS

## *Alterando o valor dos elementos pelo índice*



```
suaLista = [10,50,5,2,100]
suaLista[3]=200
i=0
while i<5:
    print(suaLista[i])
    i=i+1
```

```
>>> ===== RESTART =====
>>>
10
50
5
200
100
>>> |
```



# LISTAS

## *Adicionando elementos a lista*

1. Uma das maiores vantagens da utilização de listas é a possibilidade de adicionar elementos durante a execução do programa.
2. Na prática o programador não precisa se preocupar com a quantidade de variáveis para os alunos da turma.
3. O método `append()`:

```
umaLista = []  
umaLista.append("Joãozinho")  
umaLista.append("Chico de José")  
umaLista.append("Antônio")  
umaLista.append("Zefinha")  
i=0  
while i<4:  
    print(umaLista[i])  
    i=i+1
```

# LISTAS

## *Removendo elementos da lista*

1. Uma das maiores vantagens da utilização de listas é a possibilidade de remover elementos durante a execução do programa. **(POR VALOR)**

```
lista.remove('Melancia')  
for fruta in lista:  
    print(fruta)
```

Caju  
Laranja  
Uva  
Laranja

# LISTAS

## *Removendo elementos da lista*

1. Uma das maiores vantagens da utilização de listas é a possibilidade de remover elementos durante a execução do programa. **(POR ÍNDICE) – COMANDO “DEL”**

```
suaLista = [10,50,5,2,100]
del suaLista[2]

i=0
while i<4:
    print(suaLista[i])
    i=i+1
```

# LISTAS

## *Removendo elementos da lista*

1. Uma das maiores vantagens da utilização de listas é a possibilidade de remover elementos durante a execução do programa. **(POR ÍNDICE) – COMANDO “POP” – RETORNANDO O ELEMENTO REMOVIDO**

```
suaLista = [10,50,5,2,100]
elementoRemovido = suaLista.pop(2)
print("Removido: %d" %elementoRemovido)

i=0
while i<4:
    print(suaLista[i])
    i=i+1
```

# LISTAS

## *Tamanho de uma lista*

1. O tamanho de uma lista é igual à quantidade de elementos que ela contém.
2. Para descobrir o tamanho de uma lista existe a função len.
3. A função len retorna a quantidade de elementos na lista.

```
suaLista = [10,50,5,2,100]
tamanho = len(suaLista)
print("O tamanho dessa lista é: %d " %tamanho)
```

# LISTAS

## *Funções MIN e MAX*

1. As funções `min()` e `max()` são funções integradas do Python que retornam, respectivamente, o menor e o maior elemento de uma sequência, como uma lista.
2. Se a lista contiver elementos de diferentes tipos que não possam ser comparados entre si, `min()` / `max()` levanta uma exceção `TypeError`.

```
lista = [5, 2, 8, 1, 9]
menor_valor = min(lista)
print(menor_valor) # Saída: 1
```

```
lista = [5, 2, 8, 1, 9]
maior_valor = max(lista)
print(maior_valor) # Saída: 9
```

# LISTAS

## *Diferença de Lista e Vetor*

```
minha_lista = [1, 2, 3, "quatro", 5.0]
```

```
import array  
meu_vetor = array.array('i', [1, 2, 3, 4]) # 'i' indica inteiros
```

# EXERCÍCIOS DE FIXAÇÃO

1. Escreva um programa que leia 10 inteiros e armazene-os em uma lista “a”.
2. Escreva um programa que escreva todos os dados da lista “a”.
3. Escreva um programa que leia 10 inteiros e armazene-os em uma lista “a”. O programa deve informar qual o maior e o menor número digitado.
4. Escreva um programa que leia as notas dos alunos de uma disciplina (A quantidade de alunos deve ser informada pelo usuário) e informe quantos alunos estão abaixo da média e quantos estão na média. (Considere a nota sendo um inteiro de 0 a 100 e a média 60)
5. Escreva um programa que leia 5 números inteiros. Em seguida, determine e imprima na tela o maior elemento par da lista(se houver), o menor elemento ímpar da lista(se houver), o somatório dos elementos da lista e a média.