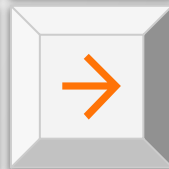
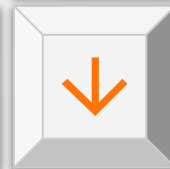
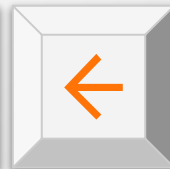
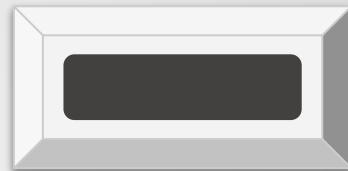


# Algoritmos e Programação

## Listas

“Transformar ideias em realidade”.

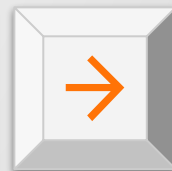




REVISÃO



LISTA



# DEFINIÇÃO DE LISTA

Em Python, uma lista é uma coleção de elementos ordenados e mutáveis. Cada elemento na lista é referenciado por um índice numérico, começando do 0 para o primeiro elemento.

As listas são criadas usando colchetes `[]` e separando os elementos com vírgulas. Por exemplo:

```
minha_lista = [1, 2, 3, 4, 5]
```

# DEFINIÇÃO DE LISTA

- Uma constante do tipo lista é escrita entre colchetes com os elementos separados por vírgula:
  - [] # lista vazia
  - [1,2] # lista com 2 elementos
- Os elementos de uma lista podem ser de qualquer tipo, inclusive listas.  
Ex.:
  - lista = [1, 'a', 'ab', 'CD']
- Os elementos de uma lista podem ser acessados por índices como strings
  - O primeiro elemento tem índice 0
  - O último elemento tem índice -1

# EXEMPLOS DE LISTAS

```
listaAlunos = ["Ana", "Beto", "Paulo"]
```

```
print(listaAlunos)
```

- ['Ana', 'Beto', 'Paulo']

```
numeros = [10, 3, -8]
```

```
print(numeros)
```

- [10, 3, -8]

```
listaAlunos = ["Ana", 10.0, "Beto", 8.5, "Paulo", 7.8]
```

```
print(listaAlunos)
```

- ['Ana', 10.0, 'Beto', 8.5, 'Paulo', 7.8]

# TAMANHO DA LISTA

```
listaAlunos = ["Ana", "Beto", "Paulo"]  
print("Tamanho da lista:", len(listaAlunos))
```

- Tamanho da lista: 3

```
listaAlunos = ["Ana", "Beto", "Paulo", "Carol"]  
print("Tamanho da lista:", len(listaAlunos))
```

- Tamanho da lista: 4

Em Python, **len()** é uma função integrada que retorna o comprimento de uma sequência, ou seja, o número de elementos na sequência.

# EXISTENCIA DE ELEMENTOS

```
listaAlunos = ["Ana", "Beto", "Paulo"]
```

```
if "Ana" in listaAlunos:  
    print("Existe na lista!")  
else:  
    print("Não existe na lista!")
```

- **Existe na lista!**

## OPERADOR "IN"

- **Permite saber se um elemento pertence a uma lista**

\*FAÇA UM TESTE ADICIONANDO MAIS PROCURE O NOME "CAROL" NA LISTA

# APRESENTANDO ELEMENTOS

Vimos que, ao executar `print(listaAlunos)` são mostrados todos os nomes da lista. Assim, o nome da lista se refere a todo seu conteúdo.

```
listaAlunos = ["Ana", "Beto", "Paulo"]  
print("Nome: ", listaAlunos[0])  
print("Nome: ", listaAlunos[1])  
print("Nome: ", listaAlunos[2])
```

- Nome: Ana
- Nome: Beto
- Nome: Paulo

Cada valor na lista é  
identificado por um índice  
(posição na lista)





# POSIÇÕES

No exemplo, quando é indicado **listaAlunos[0]** é apresentado na tela o primeiro nome da lista listaAlunos.

Posições dos elementos da lista

0

1

2

**listaAlunos**

=

[ "Ana", "Beto", "Paulo" ]

Declaração de variável

Criando uma lista

# PERCORRENDO UMA LISTA

```
listaAlunos = ["Ana", "Beto", "Paulo"]  
cont = 0
```

**Lista  
inicializada**

```
while cont < len(listaAlunos):  
    print("Posição:", cont, "Nome: ", listaAlunos[cont])  
    cont = cont + 1
```

**Tamanho da  
lista**

- **Posição: 0 Nome: Ana**
- **Posição: 1 Nome: Beto**
- **Posição: 2 Nome: Paulo**

**Elementos  
na posição**

## INSERINDO NA LISTA

```
listaAlunos = []  
print('Tamanho da lista:', len(listaAlunos), ' Elementos:', listaAlunos)
```

```
listaAlunos.append('Maria')  
print('Tamanho da lista:', len(listaAlunos), ' Elementos:', listaAlunos)
```

```
listaAlunos.append('João')  
print('Tamanho da lista:', len(listaAlunos), ' Elementos:', listaAlunos)
```

A função **append()** adiciona um novo elemento no final da lista.

# ALTERANDO VALORES

```
listaAlunos = ["Ana", "Beto", "Paulo"]  
print(listaAlunos)
```

```
listaAlunos[0]="Ana Paula"  
print(listaAlunos)
```

- ['Ana', 'Beto', 'Paulo']
- ['Ana Paula', 'Beto', 'Paulo']

**Podemos modificar um elemento da lista atribuindo um novo valor ao seu índice:**

# VALORES DE TRÁS PARA FRENTE

```
listaNum = [10, 8, 11, 5, 4, 7]
```

```
#Imprime a lista de trás para frente
```

```
print('-----')
```

```
cont = 5
```

```
while cont >= 0:
```

```
    print(listaNum[cont])
```

```
    cont = cont - 1
```

**Quando cont alcança o valor -1, o loop termina e a execução do programa continua com a próxima linha de código fora do loop.**

# VALORES DE TRÁS PARA FRENTE

```
listaPrecos = [15.00,20.00,10.00,25.00,30.00,5.00]
```

```
listaProdutos = ['Boneca', 'Camiseta', 'Caneca', 'Chaveiro', 'Caneta', 'Copo']
```

```
i=5
```

```
while i>=0:
```

```
    if listaPrecos[i-1] > listaPrecos[i]:
```

```
        print("Mais caro:", listaProdutos[i-1])
```

```
    else:
```

```
        print("Mais caro:", listaProdutos[i])
```

```
    i = i - 1
```

**Mais caro: Caneta**  
**Mais caro: Caneta**  
**Mais caro: Chaveiro**  
**Mais caro: Camiseta**  
**Mais caro: Camiseta**

A laptop is open on a dark desk. The screen shows a code editor with two files. The left file, 'index.html', contains JavaScript code for a chatbot. The right file, 'index.html', contains HTML code for a chatbot interface, including links to Google Fonts, jQuery, and a jQuery plugin. The background is a blurred indoor setting with a window and some plants.

# CORREÇÃO DOS EXERCÍCIOS

# EXERCÍCIO 11

Escreva um programa que leia as 10 respostas do gabarito de uma prova e guarde em uma lista G, sendo que o índice representa o número da questão. A seguir, leia as 10 respostas de um aluno e guarde em uma lista R. Imprima:

os números das questões que o aluno errou.

se o aluno foi aprovado ou reprovado, sendo que é aprovado o aluno que teve 6 ou mais acertos.

Exemplo:

	0	1	2	3	4	5	6	7	8	9
G	E	A	B	C	C	D	E	A	C	D
	0	1	2	3	4	5	6	7	8	9
R	A	A	B	C	D	D	E	A	C	E

Aluno acertou as questões 1, 2, 3, 5, 6, 7 e 8



## EXERCÍCIO 03 - ADICIONAL

O programa a seguir insere 10 números aleatórios, entre 1 e 50 (inclusive os dois), em uma lista.

```
import random
numeros = []
i=0
while i<10:
    numeros.append(random.randint(1,50))
    i=i+1
print("Numeros:", numeros)
```

Continue a implementação (abaixo do código apresentado), para apresentar:

- apenas os 5 primeiros elementos;
- a soma dos elementos que estão em posições pares;
- o menor entre os 3 últimos elementos.

## EXERCÍCIO 09 - ADICIONAL

Escreva um programa que leia 10 números reais e guarde em uma lista, e:

- imprima a lista;
- troque os 5 primeiros números da lista pelo seu dobro;
- troque os 5 últimos elementos da lista pela sua metade;
- Imprima a lista novamente.

## EXERCÍCIO 10- ADICIONAL

Faça um programa que leia 20 números inteiros, armazene-os em uma lista e imprima a lista. Depois, armazene os números que estão em posições pares em uma lista chamada 'listaA' e os números que estão em posições ímpares em uma lista chamada 'listaB'. Imprima as duas listas criadas.

## EXERCÍCIO 08

Faça um programa que leia 10 valores inteiros e armazene em um lista. Depois, troque o primeiro elemento com o último.

## EXERCÍCIO 09

Faça um programa que leia números inteiros e guarde em uma lista, encerrando a entrada de dados quando for digitado 0 (zero), que não deve ser armazenado.

Depois e imprima a lista criada, troque o sinal de todos os números (Ex. se o número for -5 troca por 5 ) e imprima a lista novamente.