

# ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

## AULA 9 – LISTAS

**WALTER TRAVASSOS SARINHO**

@WALTEROPROFESSOR

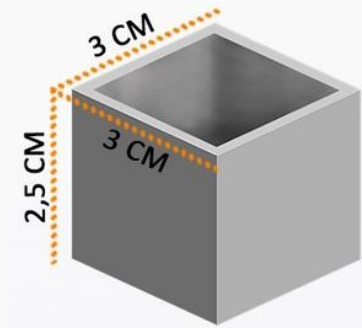
WALTER.TRAVASSOS@SECTRAS.EDU.BR



# Aula de Hoje

- Introduzir o conceito de Listas.
- Aprender a preencher e exibir o conteúdo das Listas.
- O método append.
- Slicing.
- Muitos exercícios.

# Variável x Listas



# Listas

- Uma lista é um conjunto ordenado de valores, onde cada valor é identificado por um **índice**.
- Os valores que compõem uma lista são chamados de **elementos**.

# Exemplo de lista:

```
lista = ['Python', 'Academy']  
print(lista)
```

#Saída: ['Python', 'Academy']

# Existem várias maneiras de se criar uma lista

- A maneira mais simples é envolver os elementos da lista por colchetes, por exemplo:

# Lista com apenas um elemento

```
lista = ["Python"]
```

- Também podemos criar uma lista vazia:

```
lista = []
```

- Para criar uma lista com diversos itens, podemos fazer:

```
lista = ['Python', 'Linguagem', 2023]
```

# Listas

- Sintaxe:

```
variavel = [valor1, valor2, ..., valorn]
```

- Exemplos:

```
numeros = [10, 20, 30]  
nomes = ["José", "Maria", "Carlos"]  
dados = ["alo", 2.0, 5, nomes]
```

# Exemplo 1

Faça um programa que receba do usuário a idade de 3 pessoas salvando-as dentro de uma lista e, em seguida, exiba todas as idades.

# Exemplo 1 - Resposta

```
idades = [int(input("Digite a primeira idade: ")),  
          int(input("Digite a segunda idade: ")),  
          int(input("Digite a terceira idade: "))]  
print(idades)
```

# Acessando dados da lista

- Todos os itens de uma lista são indexados, ou seja para cada item da lista, um índice é atribuído da seguinte forma: `lista[índice]`

- Exemplo com itens:

```
frutas = ['Maça', 'Banana', 'Jaca', 'Melão', 'Abacaxi']
```

E assim ficaria a sequência de índices:

Índice	0	1	2	3	4
Valores	Maça	Banana	Jaca	Melão	Abacaxi

# Exemplo 2

Faça um programa que receba do usuário a idade de 3 pessoas salvando-as dentro de uma lista e, em seguida, exiba a idade da segunda pessoa.

# Exemplo 2 - Resposta

```
idades = [int(input("Digite a primeira idade: ")),  
          int(input("Digite a segunda idade: ")),  
          int(input("Digite a terceira idade: "))]  
  
print(idades[1])
```

# Exemplo 3

Faça um programa que receba do usuário a idade de 3 pessoas salvando-as dentro de uma lista e, em seguida, mostre a maior idade digitada.

# Exemplo 3 - Resposta

```
idades = [int(input("Digite a primeira idade: ")),  
          int(input("Digite a segunda idade: ")),  
          int(input("Digite a terceira idade: "))]  
maior = 0  
  
for idade in idades:  
    if idade > maior:  
        maior = idade  
  
print(maior)
```

Usando a função **max**  
Será que tem uma função **min**?

```
1 idades = [10, 30, 50, 70]
2 maior = max(idades)
3 print(maior)
```

O método **append**

# O método append

- Método que permite adicionar dados no final de uma lista.
- Permite a adição de dados em uma lista de forma dinâmica (sem a necessidade de definir um conjunto inicial de valores).

Sintaxe:

```
idades.append(idade)
```

# Exemplo 4

Altere o programa anterior para receber dentro de um **for** as 3 idades digitadas pelo usuário e, em seguida, adicione-as na **lista** com o método **append**.

Por fim, escreva a **maior idade** digitada.

# Exemplo 4 - Resposta

```
idades = []
maior = 0

for x in range(3):
    idade = int(input('Digite uma idade: '))
    idades.append(idade)

for idade in idades:
    if idade > maior:
        maior = idade

print(maior)
```

# Fatiando uma lista (slicing)

O fatiamento de listas é a extração de um conjunto de elementos contidos numa lista. Ele é feito da seguinte forma:

`lista[ início : fim : passo ]`

Explicando cada elemento:

- **início** se refere ao índice de início do fatiamento.
- **fim** se refere ao índice final do fatiamento. A lista final não vai conter esse elemento.
- **passo** é um parâmetro opcional e é utilizado para se pular elementos da lista original

# Por exemplo...

- Se quisermos criar uma fatia de uma lista do índice 2 ao 4, podemos fazer da seguinte forma:

```
lista = [10, 20, 30, 40, 50, 60]
```

```
print(lista[2:5])
```

- O slicing conta a partir do índice 2 até o índice 5 (mas não o utiliza), pegando os índices 2, 3, 4.
- Sua saída será: [30, 40, 50]

# Questão

# Resposta: C

Analise o código Python a seguir.

```
L = []  
for x in range(10, 1, -2):  
    L.append(x)  
print(L[2:4])
```

Assinale a opção que indica os valores exibidos na execução desse código.

- a) [4, 6]      b) [6, 2]      c) [6, 4]      d) [8, 4]      e) [8, 6, 4, 2]

# Exercícios

# Exercício 1

Faça um programa que receba 6 números inteiros do usuário em uma lista e mostra apenas os números positivos.

# Exercício 1 - Resposta

```
numeros = [int(input("Digite o primeiro número: ")),  
           int(input("Digite o segundo número: ")),  
           int(input("Digite o terceiro número: ")),  
           int(input("Digite o quarto número: ")),  
           int(input("Digite o quinto número: ")),  
           int(input("Digite o sexto número: "))]  
  
for numero in numeros:  
    if numero >= 0:  
        print(numero)
```

# Exercício 2

Faça um programa que lê o sexo de 3 pessoas em uma lista, calcula e exibe a quantidade de pessoas de cada sexo.

Considere que as entradas do usuário serão apenas M ou F.

# Exercício 2 - Resposta

```
sexos = [input("Digite o sexo da primeira pessoa: "),  
         input("Digite o sexo da segunda pessoa: "),  
         input("Digite o sexo da terceira pessoa: ")]
```

```
quant_masculino = 0  
quant_feminino = 0
```

```
for sexo in sexos:  
    if sexo == "M":  
        quant_masculino = quant_masculino + 1  
    elif sexo == "F":  
        quant_feminino = quant_feminino + 1
```

```
print(quant_masculino)  
print(quant_feminino)
```

# Exercício 3

Faça um programa que leia uma lista de 10 números reais e mostre-os na ordem inversa.

# Exercício 3 - Resposta

```
1  numeros = []
2  ✓ for n in range(10):
3      n = int(input("Digite um número: "))
4      numeros.append(n)
5
6  ✓ for n in range(9,-1,-1):
7      print(numeros[n])
```

```
1  idades = [10, 30, 50, 70]
2  idades.reverse()
3  print(idades)
```

# Exercício 4

Cada espectador de um cinema respondeu a um questionário no qual constava sua opinião em relação ao filme: ótimo (3), bom (2), regular (1).

Faça um programa que receba a opinião de 5 espectadores em uma lista, calcule e mostre:

1. A quantidade de pessoas que responderam **ótimo**.
2. A quantidade de pessoas que responderam **bom**.
3. A quantidade de pessoas que responderam **regular**.

```
>_ Console x Shell x +
Digite seu voto:
1 - regular; 2 - bom; 3 - ótimo
Digite seu voto: 1
Digite seu voto: 2
Digite seu voto: 4
Digite seu voto: 5
Digite seu voto: 1
Resultado:
Total regular: 2
Total bom: 1
Total ótimo: 0
Total de votos inválidos: 2
> []
```

# Exercício 4 - Resposta

```
1 votos = []
2 print("Digite seu voto:")
3 print("1 - regular; 2 - bom; 3 - ótimo")
4 for n in range(5):
5     n = int(input("Digite seu voto: "))
6     votos.append(n)
7
8 cont1 = 0; cont2 = 0; cont3 = 0; continv = 0
9
10 for voto in votos:
11     if voto==1:
12         cont1+=1
13     elif voto==2:
14         cont2+=1
15     elif voto==3:
16         cont3+=1
17     else:
18         continv+=1
19
20 print("Resultado:")
21 print("Total regular: ", cont1)
22 print("Total bom: ", cont2)
23 print("Total ótimo: ", cont3)
24 print("Total de votos inválidos: ", continv)
```

```
>_ Console x Shell x +
Digite seu voto:
1 - regular; 2 - bom; 3 - ótimo
Digite seu voto: 1
Digite seu voto: 2
Digite seu voto: 4
Digite seu voto: 5
Digite seu voto: 1
Resultado:
Total regular: 2
Total bom: 1
Total ótimo: 0
Total de votos inválidos: 2
> []
```

# Exercício 5

Faça um programa que lê 4 números reais em uma lista, calcula e exibe a **quantidade de números negativos** e a **soma dos números positivos** dessa mesma lista.

# Exercício 6

Faça um programa que receba o preço de cinco produtos em uma lista, calcula e exiba:

1. A quantidade de produtos com preço inferior a R\$ 50,00.
2. A quantidade de produtos com preço entre R\$ 50,00 e 80,00.
3. A quantidade de produtos com preço acima de R\$ 80,00.
4. A média de preço dos produtos.

```
Menores que 50: 4  
Entre 50 e 80: 3  
Maiores que 80: 2  
A média dos valores é: 50
```

# Exercício 6 - Resposta

```
1  from statistics import mean
2  produtos=[10, 20, 30, 40, 50, 60, 70, 80, 90]
3  cont1=0; cont2=0; cont3=0
4  for preço in produtos:
5      if preço<50:
6          cont1 = cont1+1
7      elif preço <80:
8          cont2 = cont2+1
9      else:
10         cont3=cont3+1
11     print("Menores que 50: ", cont1)
12     print("Entre 50 e 80: ", cont2)
13     print("Maiores que 80: ", cont3)
14     media = mean(produtos)
15     print("A média dos valores é:", media)
```

# Exercício 7

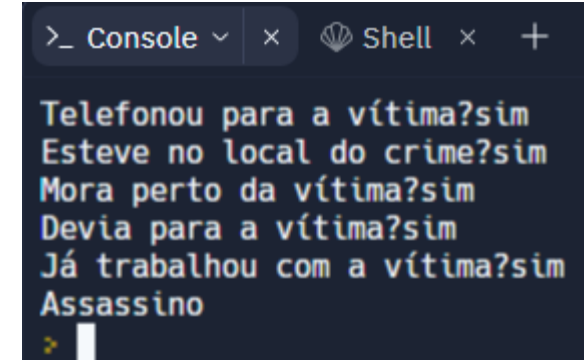
Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:

1. "Telefonou para a vítima?"
2. "Esteve no local do crime?"
3. "Mora perto da vítima?"
4. "Devia para a vítima?"
5. "Já trabalhou com a vítima?"

O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".

# Exercício 7 - Resposta

```
1 perguntas = [input("Telefonou para a vítima?"), input("Esteve no local do
crime?"), input("Mora perto da vítima?"), input("Devia para a vítima?"),
input("Já trabalhou com a vítima?")]
2
3 contsim=0; contnao=0
4 for pergunta in perguntas:
5     if pergunta=="sim":
6         contsim+=1
7     elif pergunta=="nao":
8         contnao+=1
9
10 if contsim>=5:
11     print("Assassino")
12 elif contsim>=3:
13     print("Cúmplice")
14 elif contsim>=2:
15     print("Suspeito")
16 else:
17     print("Inocente")
```



```
>_ Console × Shell × +
Telefonou para a vítima?sim
Esteve no local do crime?sim
Mora perto da vítima?sim
Devia para a vítima?sim
Já trabalhou com a vítima?sim
Assassino
>
```

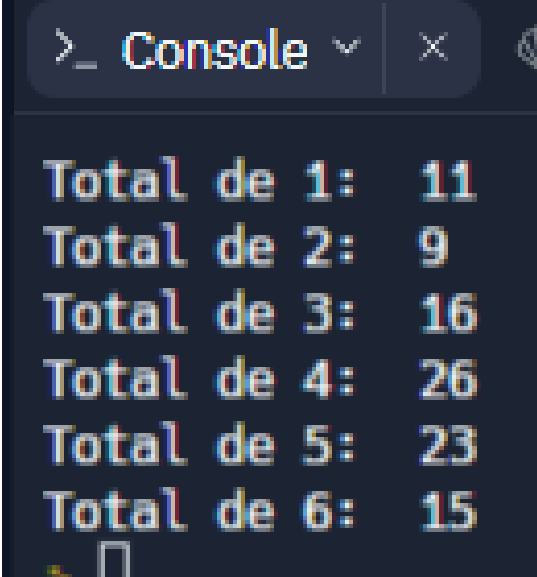
# Exercício 8



- Faça um programa que simule um lançamento de dados com valores entre 1 e 6. Use uma função para gerar números aleatórios. Pesquise sobre essa função.
- O programa deve simular o lançamento de forma randômica de um dado para 100 jogadas. Em seguida deve armazenar os resultados dessas 100 jogadas numa lista.
- Depois, o programa deve mostrar quantas vezes cada valor de 1 a 6 apareceu na lista de resultados. Use uma lista de contadores (1-6) e ao final mostre o resultado da lista de resultados e de contadores.

# Exercício 8 - Resposta

```
1 import random
2 dados = []
3 cont1=0; cont2=0; cont3=0; cont4=0; cont5=0; cont6=0
4 for x in range(100):
5     n = random.randint(1,6)
6     dados.append(n)
7     if n==1:
8         cont1+=1
9     elif n==2:
10        cont2+=1
11    elif n==3:
12        cont3+=1
13    elif n==4:
14        cont4+=1
15    elif n==5:
16        cont5+=1
17    elif n==6:
18        cont6+=1
19    print("Total de 1: ", cont1)
20    print("Total de 2: ", cont2)
21    print("Total de 3: ", cont3)
22    print("Total de 4: ", cont4)
23    print("Total de 5: ", cont5)
24    print("Total de 6: ", cont6)
```



The screenshot shows a console window titled "Console" with a dark background and light-colored text. The output consists of six lines, each showing the total count for a specific number from 1 to 6. The counts are: 11 for 1, 9 for 2, 16 for 3, 26 for 4, 23 for 5, and 15 for 6.

Total de 1:	11
Total de 2:	9
Total de 3:	16
Total de 4:	26
Total de 5:	23
Total de 6:	15

# ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 9 – LISTAS

**WALTER TRAVASSOS SARINHO**

@WALTEROPROFESSOR

WALTER.TRAVASSOS@SECTRAS.EDU.BR

