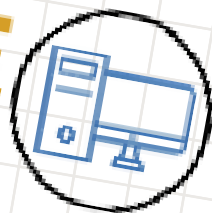


CODE  
LAB TEEN



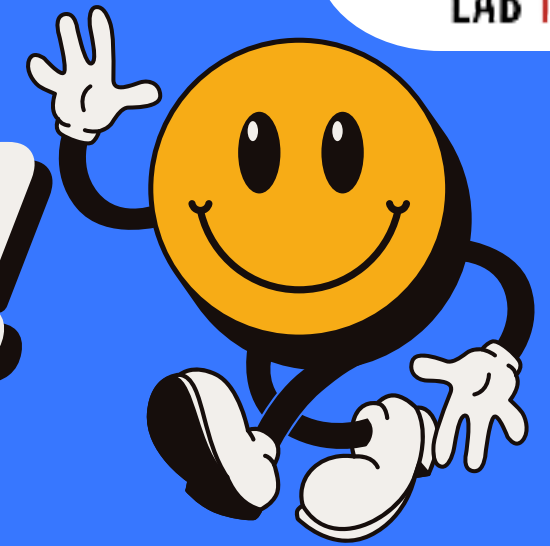
# PYTHON

## AULA 15





# ***BEM - VINDOS!***



## **AGENDA**

**Hoje vamos revisar listas e vamos dar  
início ao aprendizado de funções.**

# REVISÃO DE LISTAS

Listas em Python são como caixinhas onde você pode guardar várias coisas, uma do lado da outra. Inclusive com tipos diversos, diferente dos vetores

```
lista = ["abc", 34, True, 40, "não"]
print(lista)

['abc', 34, True, 40, 'não']
```

Você pode acessar, tirar ou colocar novos itens nessas caixinhas quando quiser.

```
lista = [1,2,3,4]

for i in range(4):
    print(f"indice: {i} valor: {lista[i]}")

indice: 0 valor: 1
indice: 1 valor: 2
indice: 2 valor: 3
indice: 3 valor: 4
```

# REVISÃO DE LISTAS

Como vimos na ultima aula, esse programa calcula a média da sua idade e da de alguns colegas. Para isso, ele armazena as idades em uma lista e utiliza um LOOP para completar o código.

```
idades = [12, 14, 15, 10]
soma = 0

for i in range(4):
    soma += idades[i]
    print(soma)
    media = soma/(i+1)

print(f"Media das idades: {media}")
```

```
51
Media das idades: 12.75
```

Esse programa é um bom exemplo de como podemos usar listas para armazenar várias informações de uma vez e depois fazer cálculos com elas. Além disso, ele mostra como o loop facilita o trabalho ao repetir a mesma ação várias vezes, somando cada idade da lista para depois calcular a média.

# ***LISTAS - ADICIONAR***

A função `append` é usada para adicionar um novo item ao final de uma lista.

```
1  frutas = ['maçã', 'banana', 'laranja']
2
3  frutas.append('uva')
4
5  print(frutas)
6
7  # Saída: ['maçã', 'banana', 'laranja', 'uva']
8
```

# ***LISTAS - REMOVE***

Remove e retorna o item na posição especificada da lista. Se nenhum índice for especificado, remove e retorna o último item da lista.

```
1  frutas = ['maçã', 'banana', 'laranja', 'uva']
2
3  # Remove e retorna o item no índice 1
4  item_removido = frutas.pop(1)
5
6  print(frutas)  # ['maçã', 'laranja', 'uva']
7  print(item_removido)  # 'banana'
```

# ***EXERCÍCIO LISTAS***

1. Crie uma lista chamada favoritos onde você vai armazenar: seu número favorito, sua cor favorita, o nome do seu filme favorito
2. Imprima a lista completa.
3. Troque o filme ou jogo favorito por outro e imprima a lista novamente.
4. Adicione mais um item à lista, como seu dia da semana favorito. Imprima a lista final.
5. Criar uma frase usando todos os itens da lista.

# EXERCÍCIO LISTAS

```
1 favoritos = [  
2     7,  
3     'azul',  
4     'O Senhor dos Anéis'  
5 ]  
6  
7 print("Lista completa inicial:", favoritos)  
8  
9 favoritos[2] = 'Inception'  
10  
11 print("Lista após alteração do filme ou jogo favorito:", favoritos)  
12  
13 favoritos.append('sexta-feira')  
14  
15 print("Lista final:", favoritos)  
16  
17 frase = f"Meu número favorito é {favoritos[0]}, minha cor favorita é {favoritos[1]}, e meu  
    filme favorito é {favoritos[2]}. Além disso, meu dia da semana favorito é {favoritos[3]}."  
18 print("Frase com todos os itens da lista:", frase)
```



# FUNÇÕES

Funções são pequenas "máquinas" de código que recebem informações (chamadas de parâmetros), realizam um processo com essas informações e devolvem um resultado. Elas ajudam a organizar o código, tornando-o mais simples e fácil de entender, além de permitir que você reutilize o mesmo código em várias partes do seu programa sem precisar reescrevê-lo.

```
1 def diga_olá(nome):  
2     return "Olá, " + nome + "!"  
3  
4 mensagem = diga_olá("Ana")  
5 print(mensagem)
```

Definindo a função

Usando a função

# FUNÇÕES

Aqui, a função `area_quadrado` recebe o comprimento do lado de um quadrado e retorna a área. É útil para calcular a área de quadrados com diferentes tamanhos.

```
1 def area_quadrado(lado):  
2     return lado * lado  
3  
4 area = area_quadrado(4)  
5 print(area)  
6 # Saída: 16
```

Neste outro exemplo, a função `soma` recebe dois números e retorna a soma deles. Você pode usar essa função sempre que precisar somar dois valores.

```
1 def soma(a, b):  
2     return a + b  
3  
4 resultado = soma(5, 3)  
5 print(resultado)  
6 # Saída: 8
```

# FUNÇÕES

Neste exemplo, a função `calcular_media` recebe uma lista de números (por exemplo, notas escolares) e calcula a média deles. A função soma todos os números da lista e divide pelo número total de itens na lista para obter a média. Esse exemplo mostra como você pode usar funções para lidar com listas e realizar cálculos mais complexos.

```
1  def calcular_media(lista):
2      soma = 0
3      quantidade = len(lista)
4      for numero in lista:
5          soma += numero
6      return soma / quantidade
7
8  notas = [8, 7, 9, 10]
9  media = calcular_media(notas)
10 print("A média das notas é:", media)
11 # Saída: A média das notas é: 8.5
```

# ***EXERCÍCIO FUNÇÕES***

Crie uma função chamada cumprimentar que recebe o nome de uma pessoa e a idade dela como parâmetros. A função deve retornar uma mensagem de cumprimento que inclua o nome e a idade da pessoa.

Por exemplo: "Olá, [nome]! Você tem [idade] anos e está ficando cada dia melhor em Python!"

# ***EXERCÍCIO FUNÇÕES***

```
1 def cumprimentar(nome, idade):  
2     return "Olá, " + nome + "! Você tem " + str(idade) + "  
   anos e está ficando cada vez mais experiente!"  
3  
4 mensagem = cumprimentar("Ana", 12)  
5 print(mensagem)  
6  
7 # Saída: Olá, Ana! Você tem 12 anos e está ficando cada dia  
   mais experiente em Python!
```

# OBRIGADO!

Contem para gente o que você achou da aula de hoje:



<https://forms.gle/GwxAyZpy1Ev4g94B7>