

AULA 18

- Avisos importantes!

- Abertura do formulário de entregas pelas monitoras

- Erika: divulgação do link para as turmas 03 e 10
 - Vanessa: divulgação do link para a turma 07

- Inclusão de item no regulamento

- Utilização do Bootstrap

- As alunas têm permissão para utilizar o Bootstrap no desenvolvimento da One Page, caso desejem

- Mostrar os regulamentos atualizados

- Datas e horários

- Projeto final

- Turma 03

- até 03/10/2023

- até 22h

- Turma 07

- até 04/10/2023
- até 22h

- Turma 10

- até 05/10/2023
- até 22h

- Depois da entrega, não realizar modificações

- Conclusão da trilha

- até 13/10/2023
- até 23h59

- Formatura

- 26/10/2023
- Divulgação das ganhadoras
- 1 ganhadora entre as turmas 03, 07 e 10

- Revisão da aula 17

- Lógica de Programação
 - Estrutura condicional if else
 - Estrutura de repetição for
 - Estrutura de repetição while

■ Diferença entre for e while

● Funções

○ O que é uma função?

- Um bloco de códigos que realiza uma tarefa específica
- Pode ser reutilizada em várias partes de um programa

○ Qual a sintaxe da função em Python?

■ Explicar sintaxe

```
def nome_da_funcao(parametro1, parametro2, ...):  
    # Corpo da função  
    # (código a ser executado quando a função é chamada)  
    # Pode incluir declarações, operações, condicionais, loops, etc.  
    return valor_de_retorno
```

○ Exemplo 01

■ Programa para somar 2 números

```
def somar(a, b):  
    resultado = a + b  
    return resultado  
  
num1 = int(input('Digite o primeiro número para somar: '))  
num2 = int(input('Digite o segundo número para somar: '))
```

```
soma = somar(num1, num2)
print(f'A soma entre {num1} e {num2} é {soma}.')
```

- O que é um parâmetro?

- São variáveis definidas na declaração da função
- Representam os valores que a função espera receber quando é chamada
- "a" e "b" são parâmetros da função "somar"

```
def somar(a, b):
    resultado = a + b
    return resultado
```

- O que é um argumento?

- Valores reais que são passados para a função quando ela é chamada

- Correspondem aos parâmetros definidos na declaração da função
- Os valores das variáveis num1 e num2 são argumentos passados para a função "somar", correspondendo aos parâmetros "a" e "b" na definição da função

```
soma = somar(num1, num2)
```

○ O que é o return?

- Serve para comunicar um resultado ou um valor de saída de volta ao código que a chamou
- Permite que o resultado da função seja armazenado em uma variável
- Quando uma função não possui uma instrução return, ela retorna None

■ LIMPAR TERMINAL

```
def somar(a, b):  
    resultado = a + b  
    print(resultado)
```

```

num1 = int(input('Digite o primeiro número para somar:
'))
num2 = int(input('Digite o segundo número para somar: '))

soma = somar(num1, num2)

print(f'A soma entre {num1} e {num2} é {soma}.')

```

○ Exemplo 02

■ Programa para calcular a média entre 2 números

```

def media(num1, num2):
    media = (num1 + num2) / 2
    return media

print(30*'-')
print('PROGRAMA PARA CÁLCULO DA MÉDIA')
print('-----\n')

num1 = float(input('Digite a primeira nota: '))
num2 = float(input('Digite a segunda nota: '))

calculo_media = media(num1, num2)

print(f'\n> A média dos números {num1} e {num2} é
{calculo_media:.1f}')

```

- \n utilizado para pular a linha

- %.1f

- : formato especial para a saída
- 1 número de casas decimais
- f número de ponto flutuante (decimal)

- LIMPAR TERMINAL

● Estrutura de dados

- O que é?

- Se refere à forma como os dados são organizados, armazenados e gerenciados em um programa

- Existem várias estruturas de dados

- Listas
- Pilhas
- Filas
- Árvores
- Grafos

- Cada estrutura mencionada, organiza os dados da sua maneira

○ O que são listas?

- Coleção de elementos que podem ser do mesmo tipo ou de tipos diferentes
- As listas podem crescer ou encolher dinamicamente, à medida que os elementos são adicionados ou removidos

○ Exemplo 01

- Criar uma lista e exibir seus itens
- ['ameixa', 'kiwi', 'pera', 'melão', 'amora']

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
  
print(frutas)
```

○ Exemplo 02

- Mostrar um item específico da lista
- Indicar o índice
- pera

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
  
print(frutas[2])
```


- Exemplo 03

- Substituir um item específico da lista

- Indicar o índice e o item

- ['ameixa', 'kiwi', 'pera', 'uva', 'amora']

- LIMPAR TERMINAL

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
frutas [3] = 'uva'  
print(frutas)
```

- Funções dentro de listas

- Append

- Adicionar um item no final da lista

- Indicar o item

- ['ameixa', 'kiwi', 'pera', 'melão', 'amora', 11]

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
frutas.append(11)  
print(frutas)
```

- Insert*

- Inserir elemento em uma posição específica
- Indicar o índice e o item
- ['ameixa', 2, 'kiwi', 'pera', 'melão', 'amora', 11]

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
  
frutas.append(11)  
frutas.insert(1, 2)  
print(frutas)
```

○ Remove

- Remover um item específico da lista
- Indicar o item
- ['ameixa', 2, 'kiwi', 'melão', 'amora', 11]

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
  
frutas.append(11)  
frutas.insert(1, 2)  
frutas.remove('pera')  
print(frutas)
```

○ Pop*

- Retirar item em uma posição específica

- Indicar o índice
- ['ameixa', 2, 'kiwi', 'amora', 11]
- LIMPAR TERMINAL

```
frutas = ['ameixa', 'kiwi', 'pera', 'melão', 'amora']  
  
frutas.append(11)  
frutas.insert(1,2)  
frutas.remove('pera')  
frutas.pop(3)  
print(frutas)
```

DESAFIO

Postar a parte 4 do projeto
final

TAREFA: SEÇÕES 45 A
56

Programação Python do Zero ao Avançado + 32 Projetos
Reais