



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 04 – LÓGICA DA PROGRAMAÇÃO

# O QUE IREMOS APRENDER

**01** EXERCÍCIOS DE REVISÃO

**02** INTRODUÇÃO

**03** FOR

**04** FOR + ELSE

**05** FOR + BREAK

**06** ATIVIDADES PRÁTICAS



# Exercício de revisão

Faça um programa, utilizando while, que permita o usuário fazer contas de adição enquanto quiser.

Faça um programa que imprima na tela apenas os números ímpares entre 1 e 50.

# Introdução

O comando for em Python é uma estrutura de repetição utilizada para executar um bloco de código um número específico de vezes. Ele é útil quando sabemos a quantidade de iterações desejadas.



# For

A sintaxe básica do for como estrutura de repetição é a seguinte:

```
for variavel in range(inicio, fim, passo):
```

variavel: Variável que assume o valor de cada iteração.

range(inicio, fim, passo):

Gera uma sequência numérica de inicio até fim-1 com o incremento definido por passo.

# For

Exemplo 1: Contagem de 1 a 5

```
for i in range(1, 6):  
    print(i)
```

Neste exemplo, o for gera uma sequência de números de 1 a 5 (o fim é exclusivo) e imprime cada número.



# For

Exemplo 2: Contagem Regressiva de 10 a 1

```
for num in range(10, 0, -1):  
    print(num)
```

Aqui, o for gera uma sequência decrescente de 10 a 1 com um passo de -1.



# For

Exemplo 3: Soma dos Números de 1 a 100

```
soma = 0

for num in range(1, 101):
    soma += num

print(f"A soma dos números de 1 a 100 é: {soma}")
```

Neste exemplo, o for é usado para somar os números de 1 a 100.



# Função range()

range( ) - Criando uma Sequência Numérica

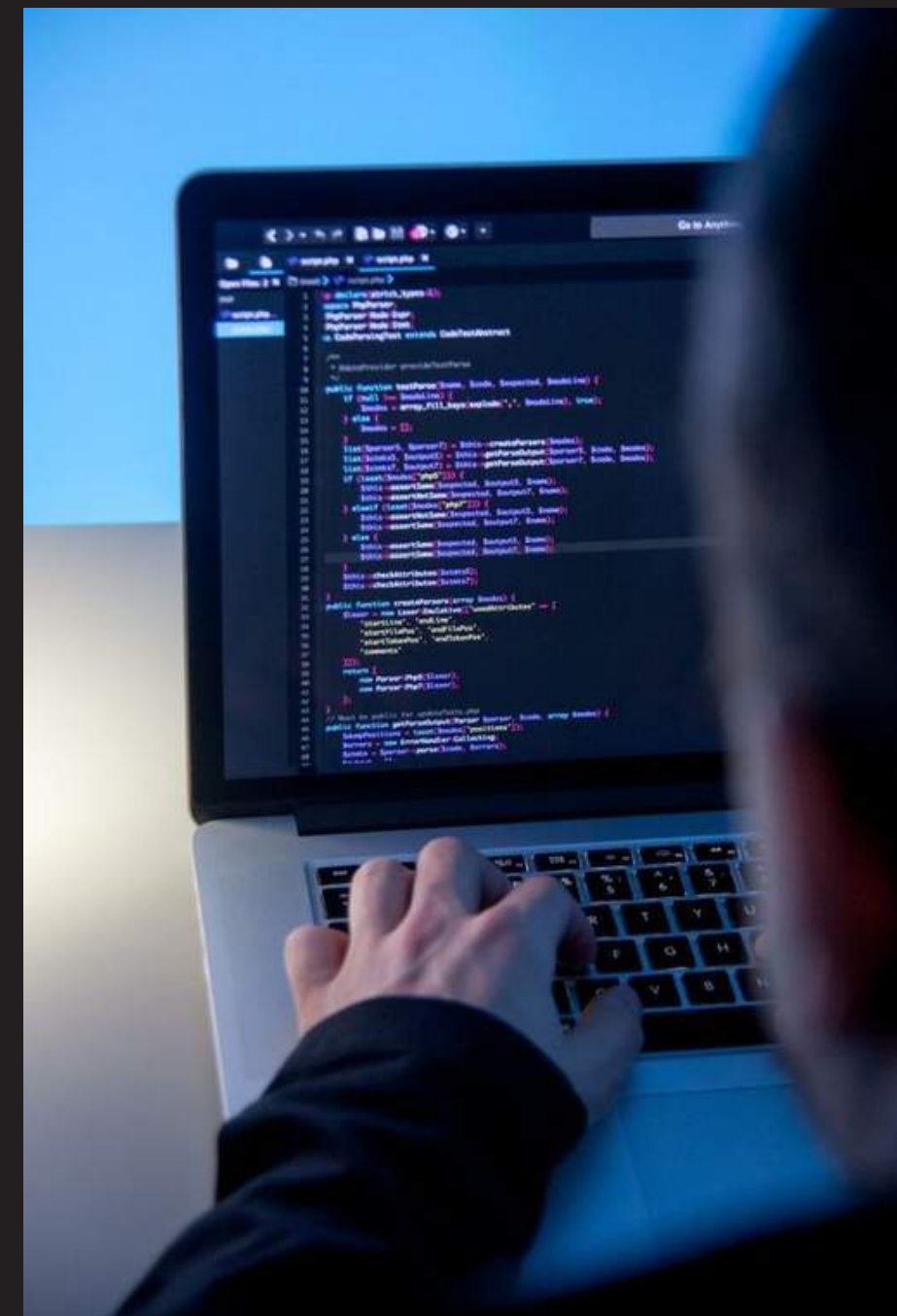
```
for i in range(5):  
    print(i)
```

A função range( ) é uma ferramenta útil para gerar sequências numéricas em um for.

# For

## Observações importantes:

- O for é usado para repetir um bloco de código um número específico de vezes.
- Utilize range(inicio, fim, passo) para controlar o número e a direção das iterações.

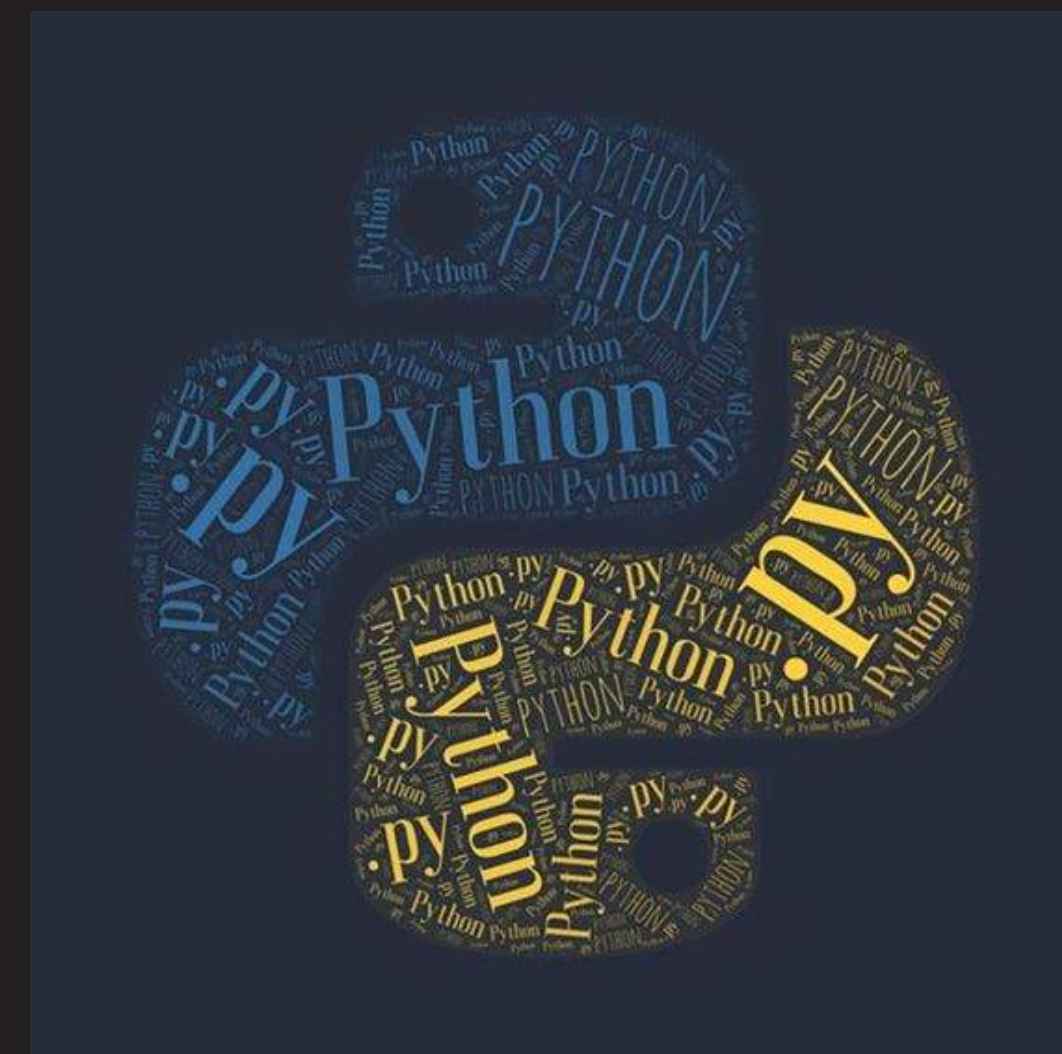


# ATIVIDADE PRÁTICA

1. Crie um programa que imprima os números pares de 2 a 10.
2. Utilize um loop for para calcular o produto dos números de 1 a 5.

# Break no for

A instrução break é utilizada para interromper um loop antes que ele atinja o final, dando assim um controle maior sobre seu código. Quando o break é encontrado, o loop é encerrado imediatamente, independentemente de ter percorrido todos os itens na sequência.



# Break no for

```
numero_procurado = 7

for i in range(1, 11):
    if i == numero_procurado:
        print(f"Número {numero_procurado} encontrado!")
        break
    print(i)
```

Neste exemplo, o loop for é interrompido assim que o número procurado é encontrado. Se o número 7 estiver presente na sequência de 1 a 10, a mensagem será exibida, e o loop será interrompido.

# Continue no for

A instrução continue é usada para pular a iteração atual de um loop e passar para a próxima iteração. Ou seja, o código abaixo da instrução continue dentro do loop não será executado na iteração atual.



# Continue no for

Exemplo: Pular Números Ímpares

```
for i in range(1, 11):  
    if i % 2 != 0:  
        continue  
    print(i)
```

Neste exemplo, o continue é usado para pular os números ímpares. Se a condição  $i \% 2 \neq 0$  for verdadeira, o continue é acionado, e o código abaixo não é executado nessa iteração.

# Break e continue

Exemplo: Parar o Loop quando Encontrar um Número Primo

```
for num in range(2, 20):  
    for i in range(2, num):  
        if num % i == 0:  
            # Não é um número primo, pular para o próximo  
            break  
    else:  
        # Chegou aqui apenas se não houve break no loop interno  
        print(f"{num} é um número primo!")
```

Neste exemplo, o loop externo percorre números de 2 a 19. O loop interno verifica se o número é primo. Se encontrar um divisor, o break é acionado, e o loop interno é encerrado. Se nenhum divisor é encontrado, a mensagem é impressa.



# Break e continue

Observações importantes:

- O break é usado para sair imediatamente de um loop.
- O continue é usado para pular para a próxima iteração do loop.
- Ambos podem ser úteis para controlar o fluxo do código dentro de loops.



# ATIVIDADE PRÁTICA

1. Crie uma programa que solicite 10 números para o usuário. O programa deve somar todos os números múltiplos de 6 digitados. Se a soma for igual ou maior que 30, o programa dever parar e mostrar o resultado da soma.
2. Utilize um loop for para imprimir os números de 1 a 20, pulando os múltiplos de 3.

# PASSO 3 PROJETO

- Neste módulo você construirá um simulador de cadastro de senha e inicialização de celular.

Use o aprendizado desta aula para implementar o for ao projeto.

Após implementar o sistema de cadastro de senha, chegou a hora de entrar no sistema operacional.

Após ligar o celular, é necessário inserir a senha cadastrada.

São 3 tentativas até o telefone bloquear.

Se o usuário acertar a senha, escreva a mensagem: "bem vindo."

Se o usuário errar a senha, escreva a mensagem:

"Senha incorretar. Você tem x tentativas até o bloqueio."

Pense em todas as possibilidades e faça um sistema completo.



# SEU MÓDULO TERMINA AQUI!

RESPONDA NOSSA  
PESQUISA DE SATISFAÇÃO



IN

**INFINITY SCHOOL**  
VISUAL ART CREATIVE CENTER





# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 02 - LÓGICA DE PROGRAMAÇÃO