



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 03 – LÓGICA DA PROGRAMAÇÃO

# O QUE IREMOS APRENDER

**01** EXERCÍCIOS DE REVISÃO

**02** INTRODUÇÃO

**03** WHILE

**04** WHILE + ELSE

**05** WHILE + BREAK

**06** ATIVIDADES PRÁTICAS



# Exercício de revisão

Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:

Salário bruto

Quanto pagou ao INSS.

Quanto pagou ao sindicato.

O salário líquido.

Calcule os descontos e o salário líquido, conforme a tabela abaixo:

Salário Bruto : R\$

IR (11%) : R\$

INSS (8%) : R\$

Sindicato ( 5%) : R\$

Salário Líquido : R\$



# Introdução

Estrutura de repetição em Python.

Uma estrutura de repetição em Python é um recurso para desenvolver tarefas repetitivas em um loop contínuo. O loop funciona até uma condição ser satisfeita.

É importante configurar devidamente essa condição para não cair em erros fatais em um programa.

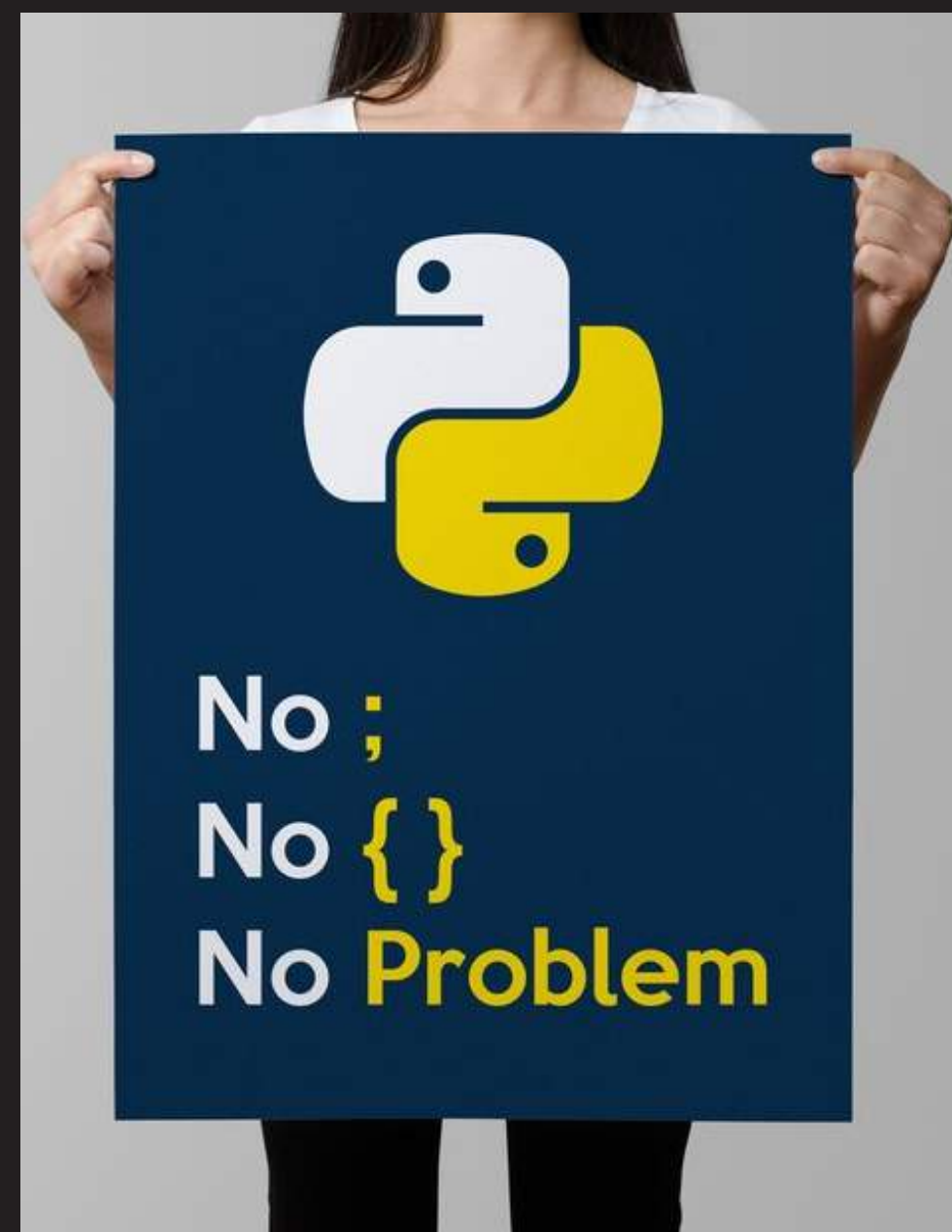
O loop faz o que chamamos em programação de iteração, uma repetição que analisa alguma estrutura.

# While

While é a estrutura de repetição que utilizamos quando queremos que determinado bloco de código seja executado ENQUANTO (do inglês while) determinada condição for satisfeita.

Faça o teste:

```
while True:  
    print('looping infinito')
```



# While

O comando while faz com que um conjunto de instruções seja executado enquanto uma condição é atendida. Quando o resultado dessa condição passa a ser falso, a execução do loop é interrompida.

Vamos ver mais um exemplo:  
Execute em seu computador.

```
contador = 0
```


```
while (contador < 5):  
    print(contador)  
    contador = contador + 1
```



# While

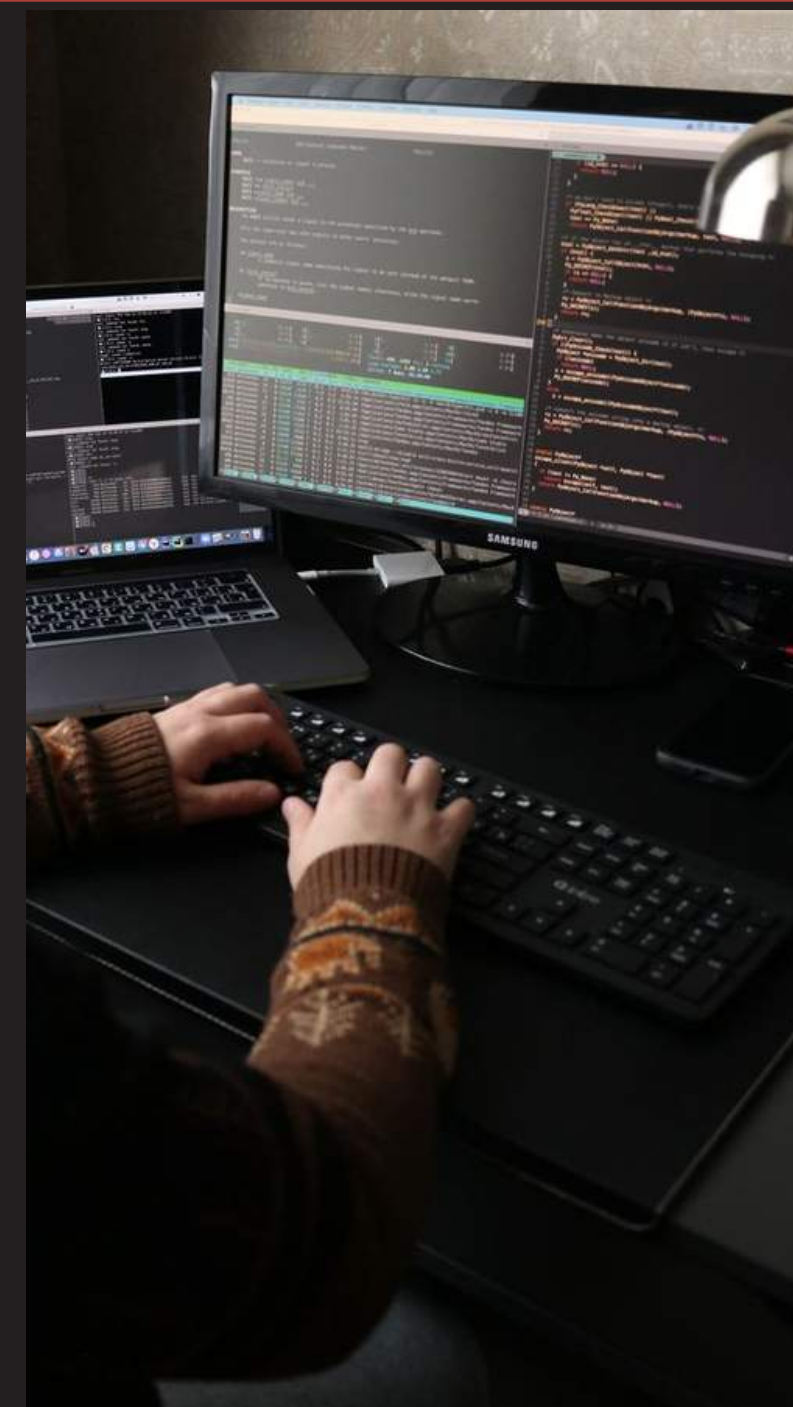
Lembre-se;

Em Python, para indicar o bloco de código pertencente ao while, devemos apenas indentar o código, conforme demonstrado no exemplo.



```
contador = 0

while contador < 5:
    print(f'Valor do contador é {contador}')
    contador += 1
```



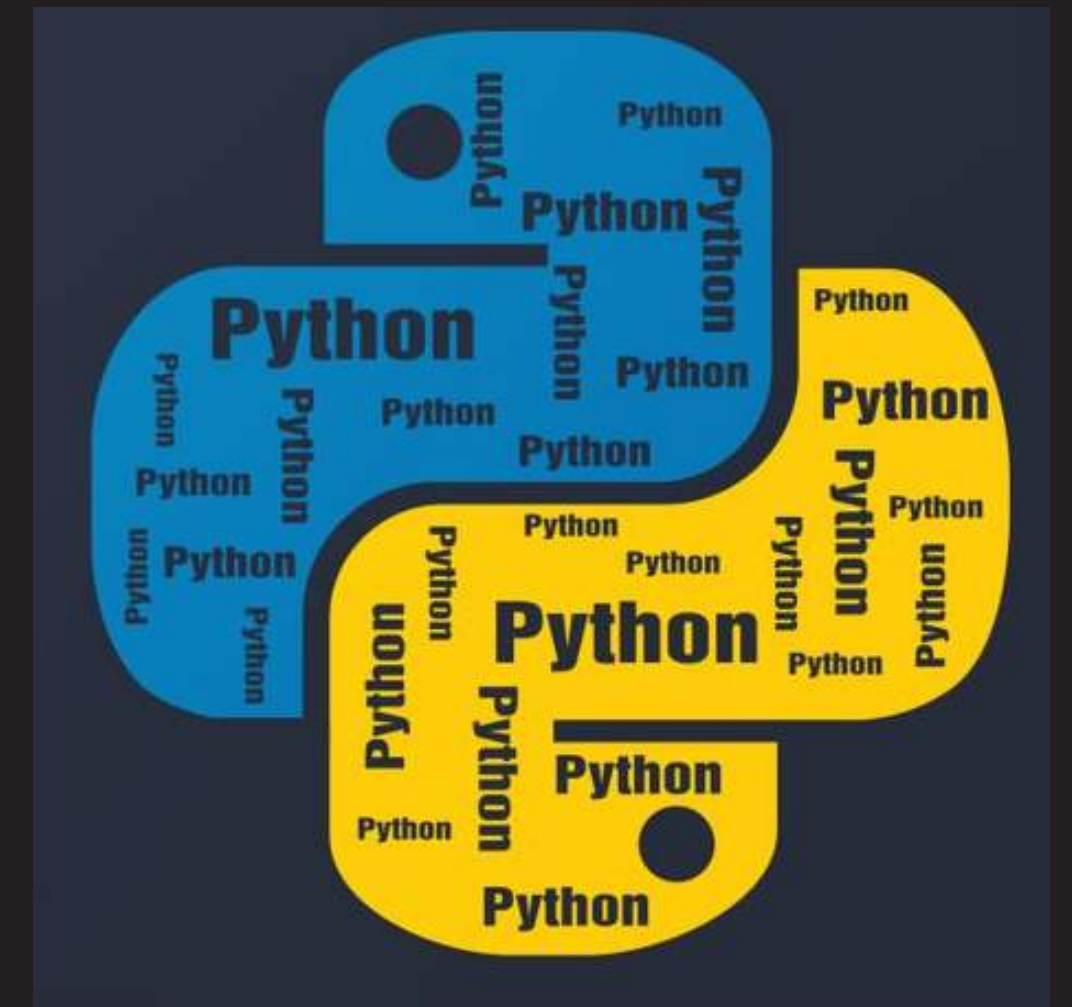


# While

A sintaxe básica do while é a seguinte:

while **condição**:

condição: Expressão lógica que determina se o loop deve continuar ou não.





# While

Exemplo 1:

Contagem regressiva

```
contador = 5

while contador > 0:
    print(contador)
    contador -= 1

print("Fim da contagem regressiva!")
```

# While

Exemplo 2:

Soma até o limite determinado.

```
soma = 0
limite = 10

while soma < limite:
    numero = int(input("Digite um número: "))
    soma += numero

print(f"A soma ultrapassou o limite de {limite}. Soma final: {soma}")
```

# While

Exemplo 3:

Adivinhação.

```
numero_secreto = 42
tentativa = 0

while tentativa != numero_secreto:
    tentativa = int(input("Tente adivinhar o número secreto: "))

print("Parabéns! Você acertou o número secreto.")
```

# While

Observações importantes.

- Certifique-se de fornecer uma maneira de quebrar o loop para evitar execução infinita.
- Atualize as variáveis dentro do loop para evitar loops infinitos acidentais.

# ATIVIDADE PRÁTICA

1. Crie um loop while que imprima os números pares de 2 a 10.
2. Implemente um jogo em que o usuário tenta adivinhar um número aleatório entre 1 e 20, dando dicas se a tentativa é muito alta ou muito baixa.

# While + else

A linguagem Python define a instrução else como uma estrutura dependente da instrução while cujo funcionamento novamente é análogo ao estudado na instrução if. Desta forma, em Python, há 4 estruturas em que a instrução else pode ser utilizado para definirmos o bloco de instrução a ser executando quando a condição verificada deixar de ser verdadeiro.

No caso, temos a instrução if, a instrução while a instrução for que será estudado na sequência e a estrutura de tratamento de exceção try except que estudaremos futuramente.



# While + else

```
contador = 5

while contador > 0:
    print(contador)
    contador -= 1
else:
    print("Fim da contagem regressiva!")
```

Neste exemplo, o loop while imprime os números de 5 a 1 e, quando a condição se torna falsa (contador <= 0), o bloco de código dentro do else é executado, indicando o fim da contagem regressiva.



# While + break

A instrução break finaliza abruptamente a execução das estruturas de repetição e assim, quando essa instrução for executada, o cursor de execução irá interromper a execução das instruções definidas pela Estrutura de Repetição e irá saltar para a linha seguinte após o Laço de Repetição.

Em outras palavras, temos que quando a instrução break for utilizada, o cursor de execução não irá passar por dentro do bloco definido pela instrução else, até porque, a instrução break encerra, imediatamente a execução da Estrutura de Repetição.

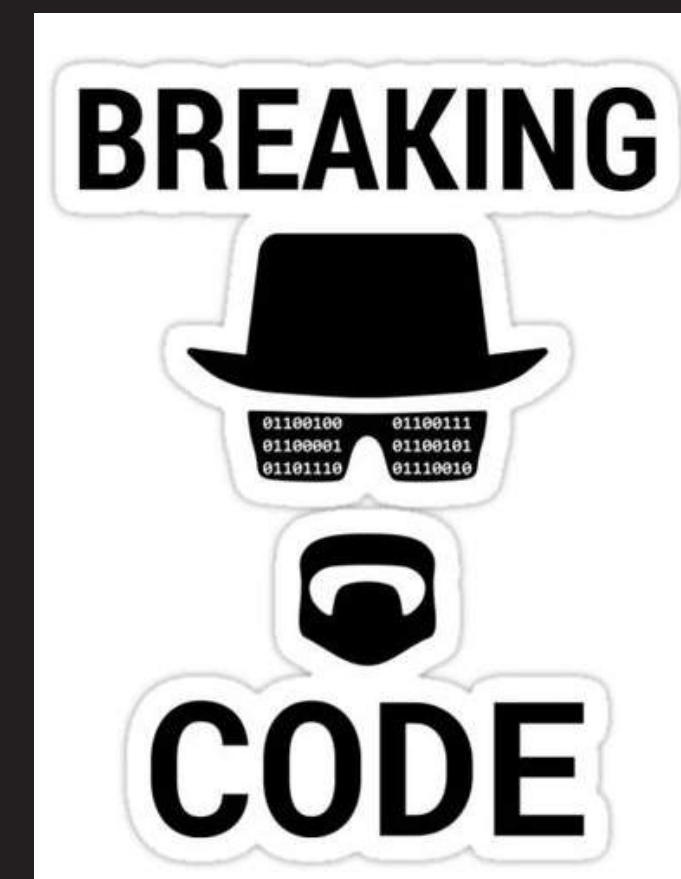
# While + break

```
soma = 0
limite = 10

while True:
    numero = int(input("Digite um número: "))
    soma += numero

    if soma >= limite:
        break # Sai do loop quando a soma atinge ou ultrapassa o limite

print(f"A soma ultrapassou o limite de {limite}. Soma final: {soma}")
```



Neste exemplo, o loop while continua indefinidamente até que a soma alcance ou ultrapasse o limite. A instrução break é usada para sair imediatamente do loop quando a condição é atendida.

# While

```
numero_secreto = 42
tentativas_maximas = 3
tentativa = 0

while tentativa != numero_secreto and tentativas_maximas > 0:
    tentativa = int(input("Tente adivinhar o número secreto: "))
    tentativas_maximas -= 1
    if tentativa != numero_secreto:
        print(f"Tentativas restantes: {tentativas_maximas}")

if tentativa == numero_secreto:
    print("Parabéns! Você acertou o número secreto.")
else:
    print(f"Suas tentativas acabaram. O número secreto era {numero_secreto}.")
```



Neste exemplo, o loop while continua enquanto o jogador não acertar o número secreto e ainda tiver tentativas restantes. O else é utilizado para imprimir uma mensagem adequada dependendo se o jogador acertou ou esgotou suas tentativas.



# While

Observações importantes:

- O bloco de código dentro do else é executado quando a condição do while se torna falsa.
- O break é usado para interromper um loop antes que a condição se torne falsa.



# ATIVIDADE PRÁTICA

Crie um programa que solicite ao usuário adivinhar um número entre 1 e 100, dando dicas se a tentativa é muito alta, muito baixa ou correta. Adicione um limite de tentativas.

# ATIVIDADE PRÁTICA

Crie um programa que solicite ao usuário que digite números inteiros. O programa deve continuar solicitando números até que a soma dos números pares digitados seja maior que 50. Ao atingir ou ultrapassar esse limite, o programa deve exibir a soma total e encerrar.

Dica:

- Use um loop while para continuar solicitando números até que a condição seja atendida.
- Mantenha uma variável para rastrear a soma dos números pares.
- Utilize a instrução break para sair do loop quando a condição for atendida.
- Exiba a soma total ao final.



# PASSO 2 PROJETO

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

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

A senha cadastrada anteriormente precisa de um looping, caso o usuário digite a senha errada.

A primeira senha deve ser igual a confirmação de senha. Enquanto o usuário não confirmar, o seu programa exibirá a mensagem:

“Senha incorreta, tente novamente.”



# SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 04 DE LÓGICA DA PROGRAMAÇÃO.  
ESTRUTURA DE REPETIÇÃO (FOR).

IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER



# Estrutura de repetição

O comando for em Python é utilizado para percorrer sequências de dados, como listas, tuplas, strings, e outros tipos iteráveis. Ele permite executar um bloco de código para cada elemento na sequência.

Mas, para entendermos de maneira prática e didática, usaremos o for como estrutura de repetição, pois, é a forma como ele se comporta ao ser executado.

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



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 02 - LÓGICA DE PROGRAMAÇÃO