

## Loops Baseados em Condições

O `while` executa um bloco de código **enquanto** uma condição for verdadeira. Por exemplo:

```
contador = 0

while contador < 5:

    print("Contador:", contador)

    contador += 1
```

**Explicação:** Esse código imprime os números de 0 a 4. O loop continua enquanto `contador < 5`.

## Loop Baseado em sequência

O `for` percorre elementos de uma sequência (como listas, strings ou intervalos).

```
for i in range(5):

    print("Número:", i)
```

**Explicação:** Imprime os números de 0 a 4. O `range(5)` gera uma sequência de 5 números.

## Parando o Loop

**break** — Interrompe o loop

O break encerra o loop imediatamente, mesmo que a condição ainda seja verdadeira.

```
contador = 0
```

```
while True:
```

```
    print("Contador:", contador)
```

```
    contador += 1
```

```
    if contador == 3:
```

```
        Break
```

**Explicação:** O loop é infinito (while True), mas o break o interrompe quando contador == 3.

OBS! Sobre os métodos usados no python.

random — Módulo de aleatoriedade

O random é um **módulo** da biblioteca padrão do Python que permite gerar valores aleatórios. Ele é usado quando você quer simular sorteios, jogos, decisões imprevisíveis, etc. (random é um **módulo**, não uma função nem um método)

Antes de usar qualquer função dele, você precisa **importar** o módulo: **import random**

`randint(a, b)` — Número inteiro aleatório

Essa é uma **função** dentro do módulo `random`. Ela retorna um número inteiro **aleatório** entre `a` e `b`, **incluindo os dois**.