

# Instrução de interrupção do Python

## Instrução de interrupção do Python

A instrução `break` do Python é usada para encerrar o `loop` atual e retoma a execução na próxima instrução, assim como a `instrução break tradicional em C`.

O uso mais comum da instrução `break` do Python é quando alguma condição externa é acionada, exigindo uma saída apressada de um `loop`. A instrução **`break`** pode ser usada em `loops while` e `for` do Python .

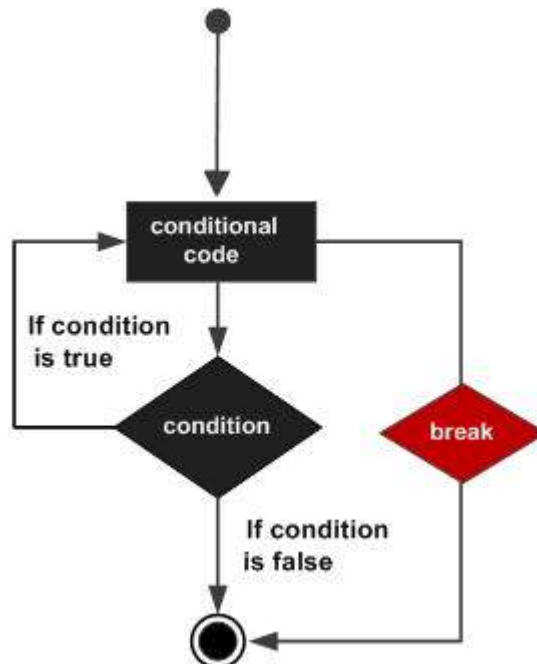
Se você estiver usando `loops aninhados em Python` , a instrução `break` interrompe a execução do `loop` mais interno e inicia a execução da próxima linha de código após o bloco.

## Sintaxe da instrução break

A sintaxe para uma instrução **`break`** em Python é a seguinte -

```
break
```

## Diagrama de fluxo da instrução break



## Exemplos de instrução break

Exemplo 1: Demonstrando o uso da instrução `break` do Python



```

for letter in 'Python':      # First Example
    if letter == 'h':
        break
    print ('Current Letter :', letter)

var = 10                      # Second Example
while var > 0:
    print ('Current variable value :', var)
    var = var -1
    if var == 5:
        break

print ("Good bye!")

```

Quando o código acima é executado, produz o seguinte resultado -

```

Current Letter : P
Current Letter : y
Current Letter : t
Current variable value : 10
Current variable value : 9
Current variable value : 8
Current variable value : 7
Current variable value : 6
Good bye!

```

## Exemplo 2: Verificando um número na lista

O programa a seguir demonstra o uso de break em um loop **for** iterando sobre uma **lista** . O **usuário insere** um número, que é pesquisado na lista. Se for encontrado, o loop termina com a mensagem 'encontrado'.

```

no=int(input('any number: '))
numbers=[11,33,55,39,55,75,37,21,23,41,13]
for num in numbers:
    if num==no:
        print ('number found in list')
        break
else:
    print ('number not found in list')

```

O programa acima produzirá a seguinte **saída** -

any number: 33  
number found in list  
any number: 5  
number not found in list

### Exemplo 3: verificação de número primo

Observe que quando a instrução **break** é encontrada, o Python abandona as instruções restantes no loop, incluindo o bloco **else** .

O exemplo a seguir aproveita esse comportamento para descobrir se um número é primo ou não. Por definição, um número é primo se não for divisível por nenhum outro número, exceto por 1 e por ele mesmo.

O código a seguir executa um loop **for** sobre números de 2 até o número 1 desejado. Se for divisível por qualquer valor da variável de loop, o número não é primo, portanto o programa sai do loop. Se o número não for divisível por nenhum número entre 2 e x-1, o bloco **else** imprime a mensagem de que o número fornecido é primo.

```
num = 37
print ("Number: ", num)
for x in range(2,num):
    if num%x==0:
        print ("{} is not prime".format(num))
        break
else:
    print ("{} is prime".format(num))
```

Atribua valores diferentes a num para verificar se é um número primo ou não.

Number: 37  
37 is prime  
Number: 49  
49 is not prime