

Instrução Continuar do Python

Declaração de continuação do Python

A **instrução continue do Python** é usada para pular a execução do bloco do programa e retornar o controle ao início do **loop** atual para iniciar a próxima iteração. Quando encontrado, o loop inicia a próxima iteração sem executar as instruções restantes na iteração atual.

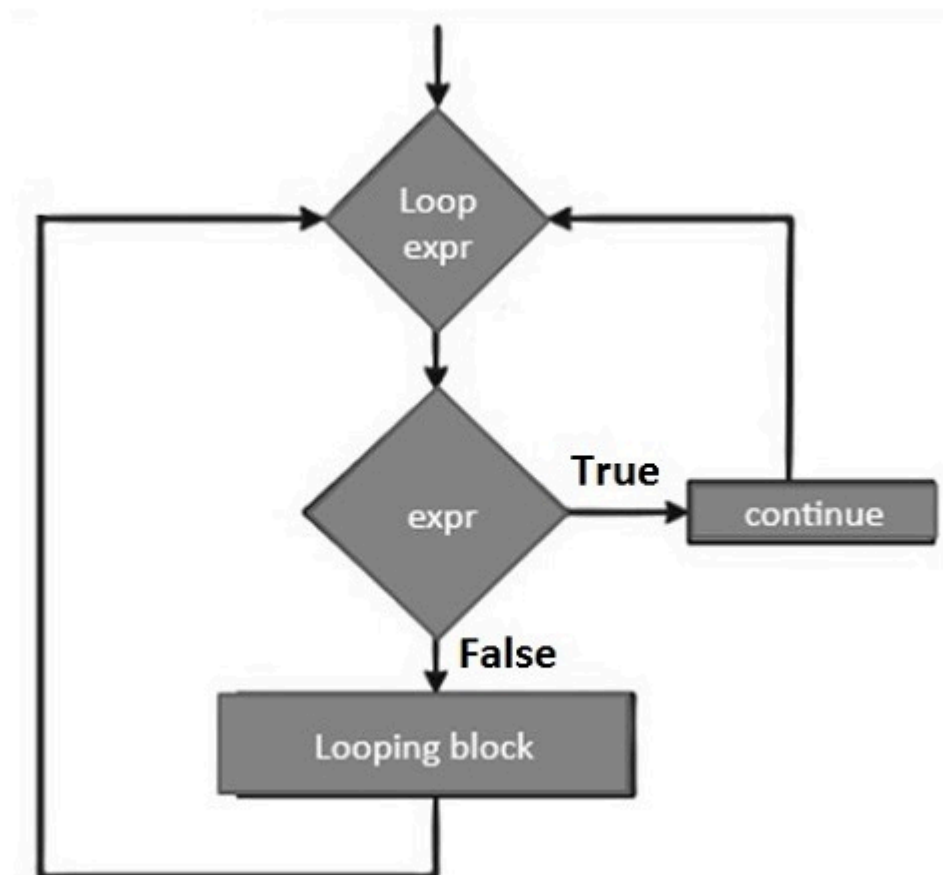
A instrução **continue** pode ser usada em loops **while** e **for**.

Sintaxe da instrução continue

```
continue
```

Diagrama de fluxo da instrução continue

O diagrama de fluxo da instrução **continue** é assim -



A instrução **continue** é exatamente o oposto daquela de **break**. Ele ignora as instruções restantes no loop atual e inicia a próxima iteração.



Exemplo de declaração continue

Agora vamos dar um exemplo para entender como funciona a instrução **continue** em Python

-

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

var = 10 # Second Example
while var > 0:
    var = var -1
    if var == 5:
        continue
    print ('Current variable value :', var)
print ("Good bye!")
```

Quando o código acima é executado, ele produz a seguinte **saída** -

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : o
Current Letter : n
Current variable value : 9
Current variable value : 8
Current variable value : 7
Current variable value : 6
Current variable value : 4
Current variable value : 3
Current variable value : 2
Current variable value : 1
Current variable value : 0
Good bye!
```

Instrução Python continue com loop While

A instrução continue do Python é usada com loops 'for' e também com loops 'while' para pular a execução da iteração atual e transferir o controle do programa para a próxima iteração.

Exemplo: Verificando fatores primos

Os usos do código a seguir continuam a encontrar os fatores primos de um determinado número. Para encontrar os fatores primos, precisamos dividir sucessivamente o número fornecido começando com 2, incrementar o divisor e continuar o mesmo processo até que a entrada se reduza a 1.

O algoritmo para encontrar fatores primos é o seguinte -

- Aceitar entrada do usuário (n)
- Defina o divisor (d) como 2
- Execute o seguinte até $n > 1$
- Verifique se determinado número (n) é divisível pelo divisor (d).
- Se $n \% d == 0$
 - a. Imprima d como um fator
 - Defina o novo valor de n como n/d
 - Repetir a partir de 4
- Se não
- Incrementar d em 1
- Repetir a partir de 3

Abaixo está o código Python para o propósito -

```
num = 60
print ("Prime factors for: ", num)
d=2
while num > 1:
    if num%d==0:
        print (d)
        num=num/d
        continue
    d=d+1
```

Ao ser executado, este código produzirá a seguinte **saída** -

```
Prime factors for: 60
2
2
3
5
```



Atribua um valor diferente (digamos 75) a num no programa acima e teste o resultado quanto aos seus fatores primos.

Prime factors for: 75

3

5

5