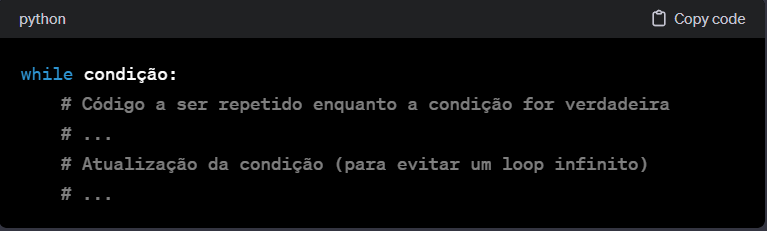
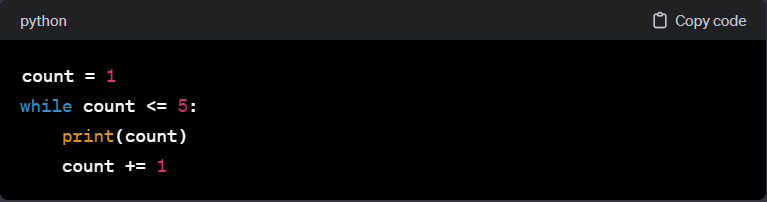
PYTHON while loops

[While Loops](https://www.w3schools.com/python/python_while_loops.asp)

Em python, um ‘While loop’ é uma estrutura de controle de fluxo que permite executar repetidamente um bloco de código enquanto uma condição especificada for verdadeira. A ideia é que o bloco de código dentro do loop seja executado repetidamente até a condição fornecida não seja mais atendida.



Exemplo simples que imprime os numeros de 1 a 5:



Neste exemplo, o loop continuará a ser executado enquanto a condição count <= 5 for verdadeira.O bloco de código dentro do loop imprime o valor de count e, em seguida, incrementa count em 1. O loop terminará quando count atingir o valor 6, por a condição não será mais verdadeira.

É importante tomar cuidado ao usar o WHILE LOOPS para evitar loops infinitos, onde a condição nunca se torna falsa, resultando em um programa que nunca termina de executar.

Uma alternativa é o FOR LOOPS que é usado para iterar sobre uma sequencia(tuplas, listas, conjuntos) em vez de depender de uma condição booleana. Escolher entre esses dois depende da situação específica e da lógica do programa que está sendo criado.

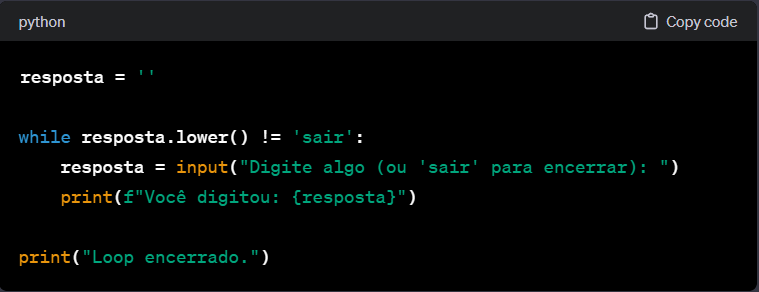
**ESTRUTURA GERAL**

O bloco de código dentro do **While** será repetido enquanto a condição especificada for verdadeira. Se a condição for falsa desde o inicio, o bloco de código não será executado.

**ATUALIZAÇÃO DA CONDIÇÃO**

É crucial garantir que a condição eventualmente se torne falsa, caso contrário, você terá um loop infinito. Dentro do bloco de código, normalmente você atualiza as variáveis que fazem parte da condição.

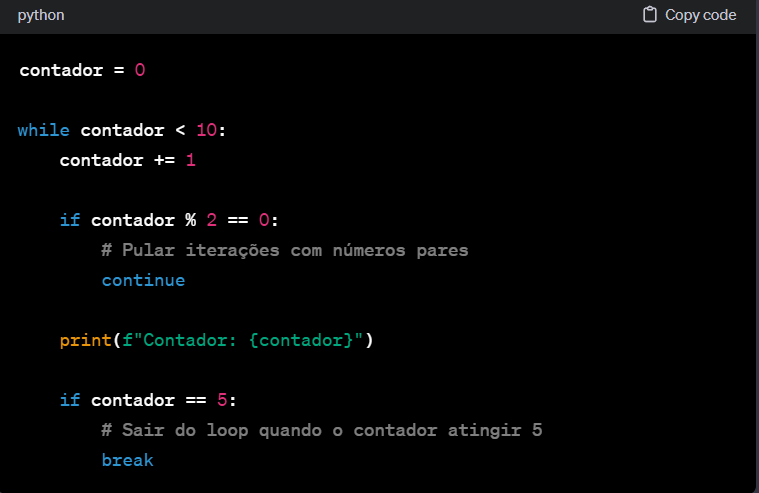
**EXEMPLO COM ENTRADA DO USUÁRIO**

 Um exemplo prático pode ser criar um **while** que solicita entrada do usuário até que uma entrada especifica seja fornecida:

Neste exemplo, o loop continua até que o usuário digite “sair”. A condição “resposta.lower()” != “sair” é verificada a cada iteração.

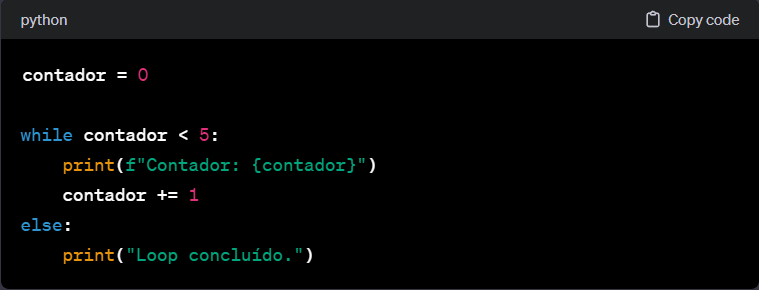
**CONTINUE E BREAK**

* A instrução ‘**continue**’ pode ser usada para pular o restante do código no loop e ir para a próxima iteração.
* A instrução ‘**break**’ pode ser usada para sair completamente do while loop antes de a condição se tornar falsa.



**USOS AVANÇADOS**

* **ELSE** em **WHILE** **LOOPS**: Similar ao **IF**, você pode usar else em um while loop, que será executado quando a condição do loop se tornar falsa.



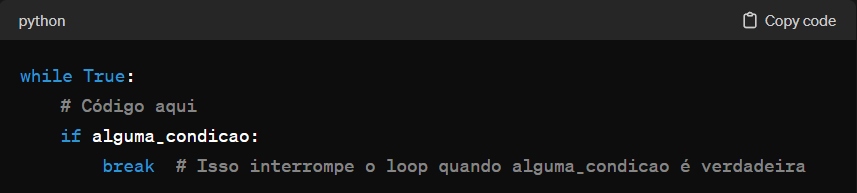
**INTERROMPENDO REPETIÇÕES**

* Para interromper uma repetição While, pode-se usar a instrução break, ela irá interromper instântaneamente a repetição;



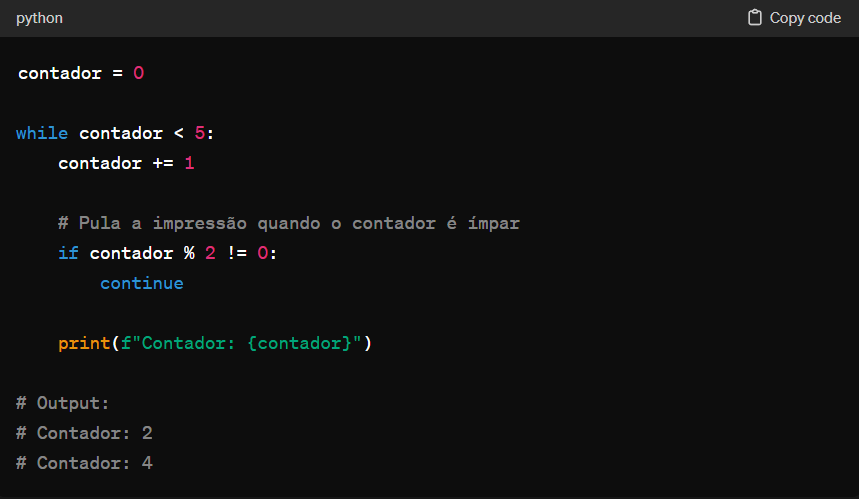
**WHILE TRUE**

* O while True cria um loop que contiuará a ser executado enquanto a expressão booleana após o while for avaliada como verdadeira. Neste caso, TRUE é um constante que sempre é verdadeira, então o loop será executado indefinidamente, a menos que seja interrompida por uma insturção break ou uma condição dentro do loop;



**CONTINUE**

* A instrução continue é usada em loops para pular o restante do código dentro do loop e continuar a próxxima iteração. Quando a instrução continue é encontrada, o controle é transferido de volta para o ínicio do loop, ignorando qualquer código restante abaixo dela dentro do mesmo loop;



Neste exemplo o loop while é executado até o contador seja menor que 5. Dentro do loop, quando contador é impar ‘continue’ é acionada pulando a impressão e indo para a proxima iteração.