Python - break Keyword

The break keyword causes the abandonment of pending iterations of the current loop. The execution of the program jumps to the statement immediately after the body of the loop.

Typical use of break is found in a sequential search algorithm. For example, if you need to search for an object in a collection, you will have to execute a comparison expression in a loop. However, if the required object is found, an early exit from the loop is sought, without traversing the remaining collection.

The following example demonstrates the use of break inside a while loop:

Example: break within while loop

```
num=0
while num<5:
   num=num+1
   print ("Num =  {} ".format(num))
   if num==3:
      break
print ("Out of loop")
```

The above code will display the following result:

Output

```
Num = 1
Num = 2
Num = 3
Out of loop
```

Usage of break within the for loop is shown below:

Example: break within for loop

```
for num in range(1,6):
    print ("Num =  {} ".format(num))
    if num==3:
        break
print ("Out of loop")
```

The following program demonstrates the use of break in a loop. It accepts a number as input and determines whether it is a prime number or not. By definition, n is prime if it is not divisible by any number between the range 2 to n-1.

```
num=int(input("Enter a number: "))
for x in range(2,num):
    if num%x==0:
        print("{} is not prime".format(num))
        break
else:
    print ("{} is prime".format(num))
```

Output

```
Enter a number: 57
57 is not prime

Enter a number: 53
53 is prime
```