

Python break and continue

What is the use of break and continue in Python?

In Python, break and continue statements can alter the flow of a normal loop.

Loops iterate over a block of code until test expression is false, but sometimes we wish to terminate the current iteration or even the whole loop without checking test expression.

The break and continue statements are used in these cases.

Python break statement

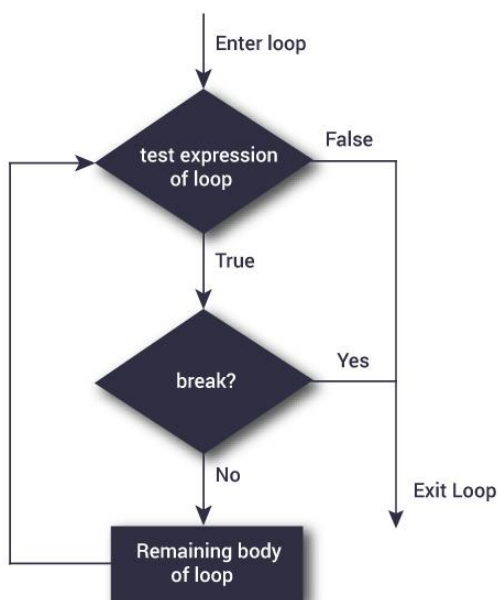
The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.

If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop.

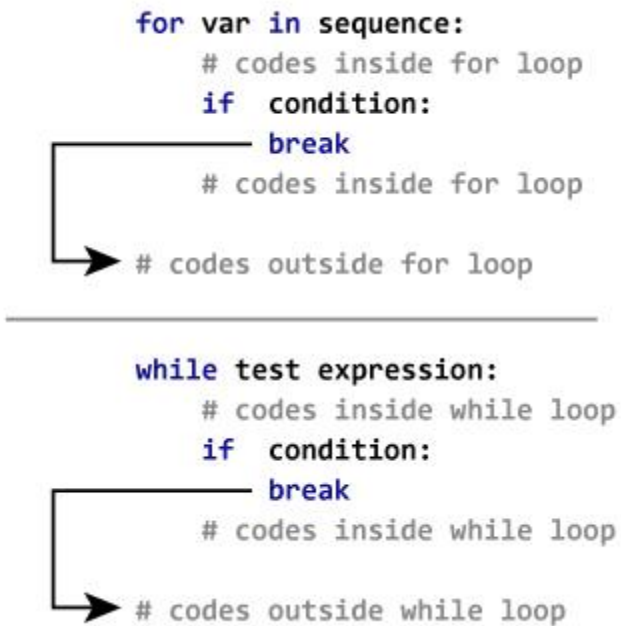
Syntax of break

```
break
```

Flowchart of break



The working of break statement in [for loop](#) and [while loop](#) is shown below.



Example: Python break

```
# Use of break statement inside loop
for val in "string":
    if val == "i":
        break
    print(val)
print("The end")
```

Output

```
s
t
r
The end
```

In this program, we iterate through the "string" sequence. We check if the letter is "i", upon which we break from the loop. Hence, we see in our output that all the letters up till "i" gets printed. After that, the loop terminates.

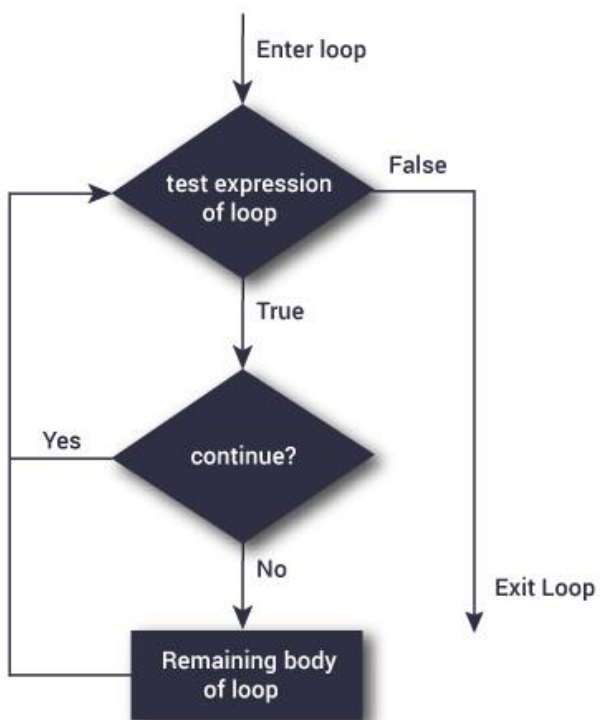
Python continue statement

The continue statement is used to skip the rest of the code inside a loop for the current iteration only. Loop does not terminate but continues on with the next iteration.

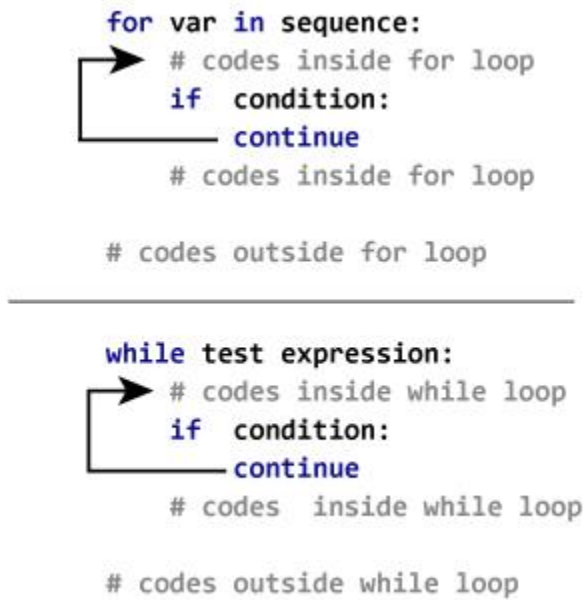
Syntax of Continue

```
continue
```

Flowchart of continue



The working of continue statement in for and while loop is shown below.



Example: Python continue

```
# Program to show the use of continue statement inside loops  
for val in "string":  
    if val == "i":  
        continue  
    print(val)  
print("The end")
```

Output

```
s  
t  
r  
n  
g  
The end
```

This program is same as the above example except the break statement has been replaced with continue.

We continue with the loop, if the string is "i", not executing the rest of the block. Hence, we see in our output that all the letters except "i" gets printed.

Python pass statement

What is pass statement in Python?

In Python programming, `pass` is a null statement. The difference between a comment and `pass` statement in Python is that, while the interpreter ignores a comment entirely, `pass` is not ignored.

However, nothing happens when `pass` is executed. It results into no operation (NOP).

Syntax of pass

```
pass
```

We generally use it as a placeholder.

Suppose we have a [loop](#) or a [function](#) that is not implemented yet, but we want to implement it in the future. They cannot have an empty body. The interpreter would complain. So, we use the `pass` statement to construct a body that does nothing.

Example: pass Statement

```
# pass is just a placeholder for
# functionality to be added later.
sequence = {'p', 'a', 's', 's'}
for val in sequence:
    pass
```

We can do the same thing in an empty function or [class](#) as well.

```
def function(args):
    pass

class example:
    pass
```