Cheat Sheet

Loop Control Statements

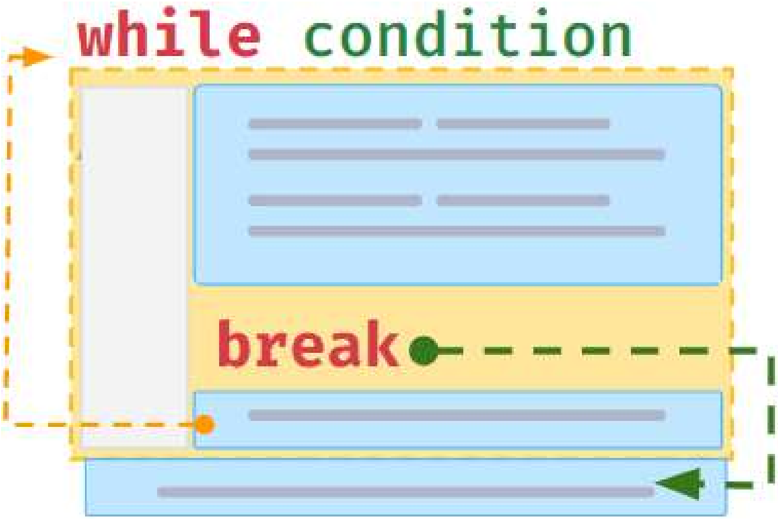
Control statements alter the sequential execution of a program.

Examples

- **if-elif-else**
- **while, for**
- **break, continue**

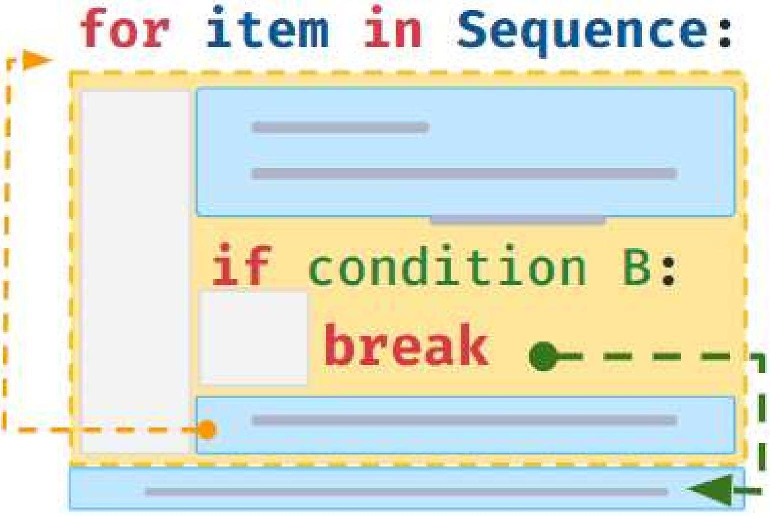
Break

Break statement makes the program exit a loop early.



Using Break

Generally, break is used to exit a loop when a condition is satisfied.



In the below example, when the variable

`i` value equals to `3` the `break` statement gets executed and stops the execution of the loop further.

Code

PYTHON

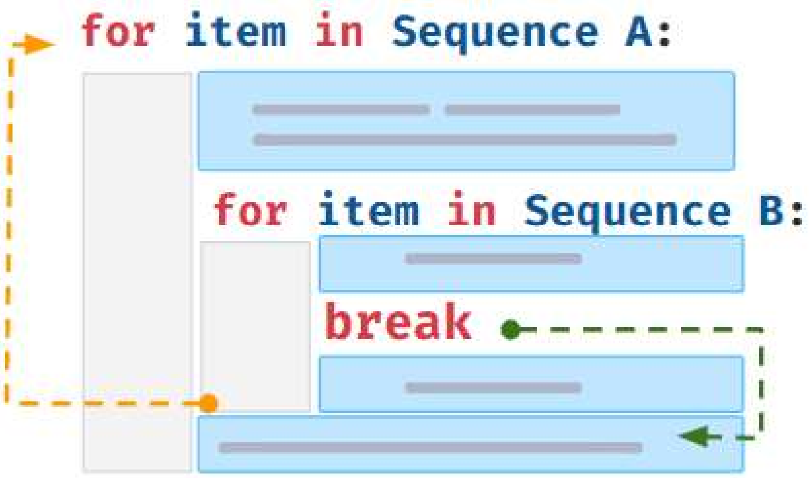
```python
for i in range(5):
  if i == 3:
    break
  print(i)
print("END")
```

Output

```
0
1
2
END
```

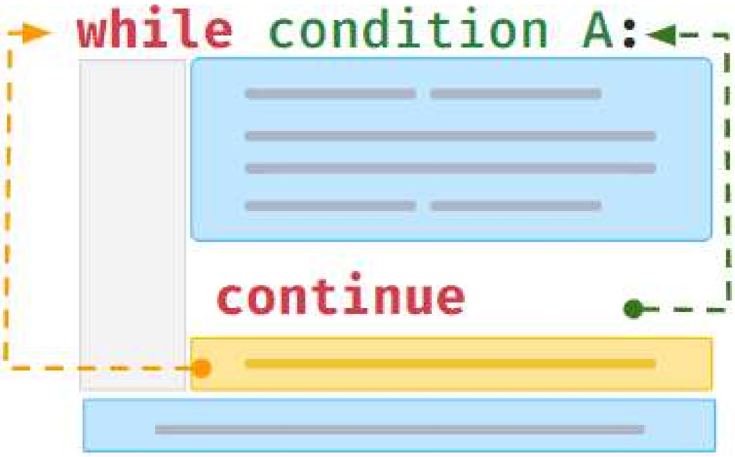
Break in Nested Loop

Break in inner loop stops the execution of the inner loop.



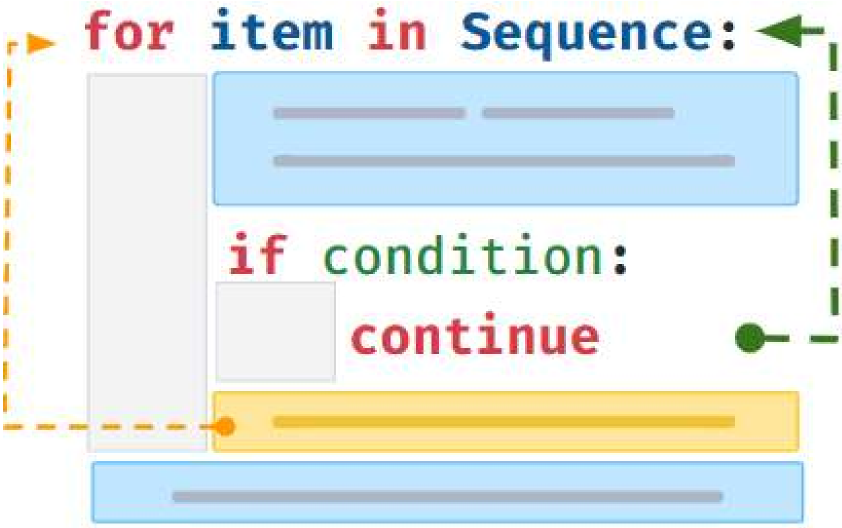
Continue

Continue makes the program skip the remaining statements in the current iteration and begin the next iteration.



Using Continue

Generally, continue is used to skip the remaining statements in the current iteration when a condition is satisfied.



In the below example, when the variable

`i` value equals to `3` the next statements in the loop body are skipped.

Code

PYTHON

```python
for i in range(5):
  if i == 3:
    continue
  print(i)
print("END")
```

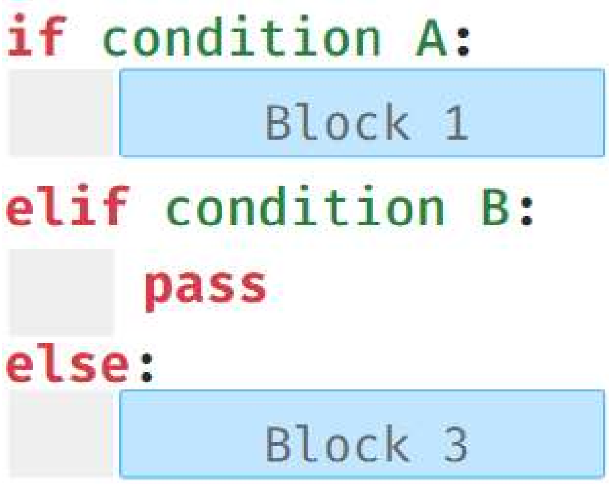
Output

```
0
1
2
4
END
```

Pass

Pass statement is used as a syntactic placeholder. When it is executed, nothing happens.

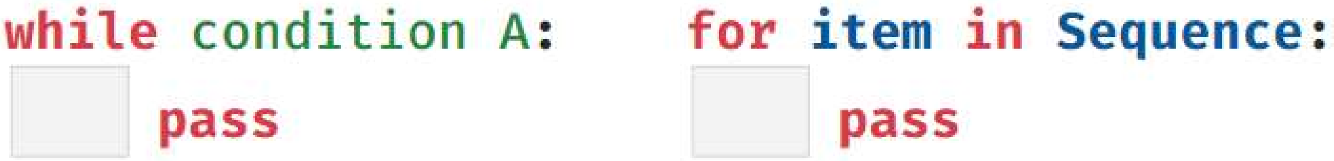
Generally used when we have to test the code before writing the complete code.



```
if condition A:
    Block 1
elif condition B:
    pass
else:
    Block 3
```

Empty Loops

We can use pass statements to test code written so far, before writing loop logic.



```
while condition A:
    pass
```

```
for item in Sequence:
    pass
```

Submit Feedback