1. What are logical operators?

The logical operators are used to perform logical operations on Boolean values.
Gives

`True` or `False` as a result.

Following are the logical operators

- `and`
- `or`
- `not`

Logical AND Operator

Gives

`True` if both the booleans are true, else, it gives `False`

Code

PYTHON

```
print(True and True)
```

Output

```
True
```

Example

Code

PYTHON

```
print((2 < 3) and (1 < 2))
```

Step by Step Explanation

```
(2 < 3) and (1 < 2)
True and (1 < 2)
True and True
```

Output

```
True
```

Logical OR Operator

Gives

`True` if any one of the booleans is true, else, it gives `False`

Code

PYTHON

```python
print(False or False)
```

Output

```
False
```

Example

Code

PYTHON

```python
print((2 > 3) or (2 < 1))
```

Step by Step Explanation

```
(2 > 3) or (2 < 1)
False or (2 < 1)
False or False
```

Output

```
False
```

Logical NOT Operator

Gives the opposite value of the given boolean.

Code

PYTHON

```python
print(not(False))
```

Output

```
True
```

Example

Code

PYTHON

```python
print(not(2 < 3))
```

Step by Step Explanation

```
not(2 < 3)
not(True)
False
```

Output

```
False
```

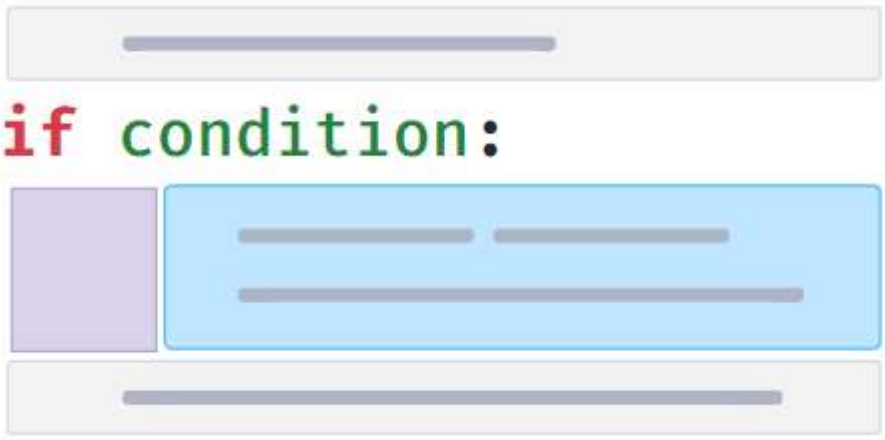
2. What are conditional statements?

The Conditional Statement allows you to execute a block of code based on a condition.

If statement

The Conditional Statement allows you to execute a block of code only when a specific condition is

`True` .



code

PYTHON

```python
if True:
    print("If Block")
    print("Inside If")
```

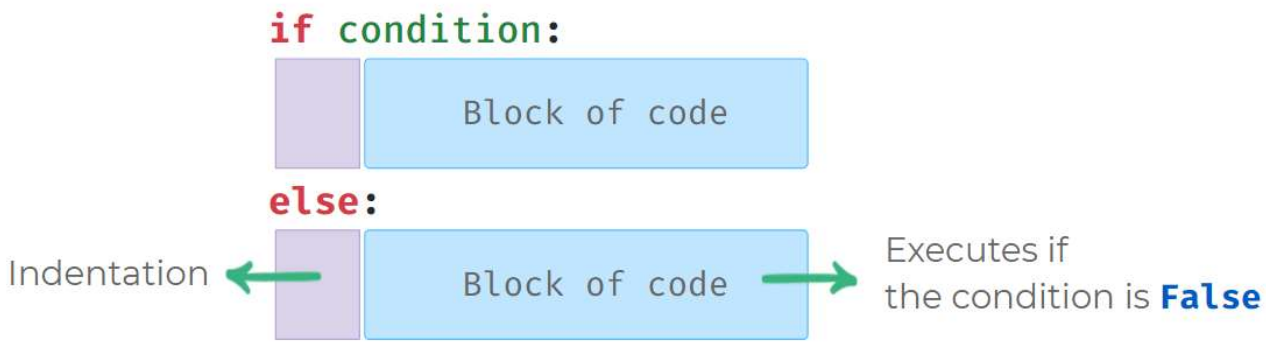
Output

```
If Block
Inside If
```

If-Else statement

When If-Else conditional statement is used, the Else block of code executes if the condition is

`False` .



Using If-Else

Code

PYTHON

```python
a = int(input())
if a > 0:
    print("Positive")
else:
    print("Not Positive")
print("End")
```

Input

```
2
```

Output

```
Positive
End
```

3. What are Loops?

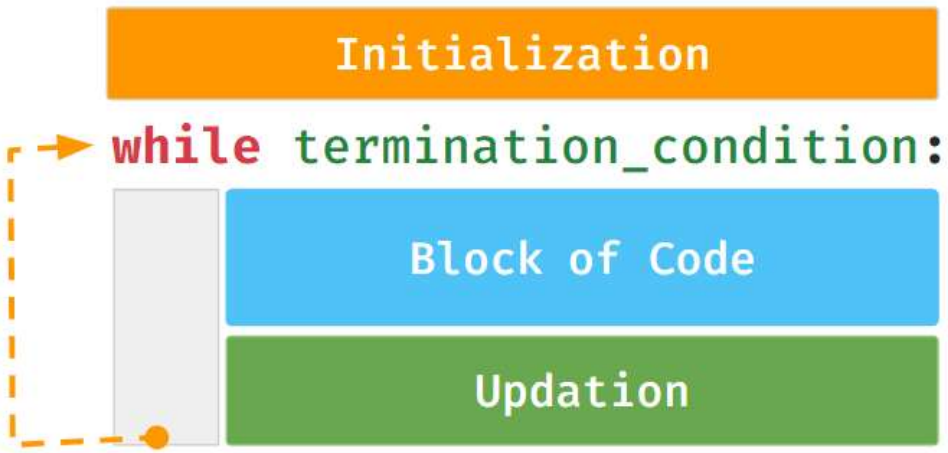
Loops allow us to execute a block of code several times.

The loops in Python are:

- While Loop
- For Loop

4. What is the use of `while` loop?

While loop allows us to execute a block of code several times as long as the condition is `True` .



Example

The following code snippet prints the next three consecutive numbers after a given number.

Code

PYTHON

```python
a = int(input())
counter = 0
while counter < 3:
    a = a + 1
    print(a)
    counter = counter + 1
```

Input

```
4
```

Output

```
5
6
7
```

5. How to create a `do-while` loop in python?

Generally, in other programming languages, we will have another loop called the

`do-while` loop.

In

`do-while` loops, the code in the loop runs at least once before checking if the condition is **True**.

The

`do-while` loop is not directly available in Python.

Example: do-while loop in c language

code

```c
#include <stdio.h>

int main () {
   int a = 1;

   do {
      printf("%d\n", a);
      a = a + 1;
   }while( a < 4 );

}
```

output

```
1
2
3
```

If the condition checked evaluates to **True**, the loop continues. For the cases where you would want your code to run at least one time then the

`do-while` loops will come in handy.

do-while loop in python:

In Python, we can create a

`do-while` loop by using the `while` loop to achieve similar behavior.

Code

PYTHON

```python
i = 1

while True:
    print(i)
    i = i + 1
    if(i > 3):
        break
```

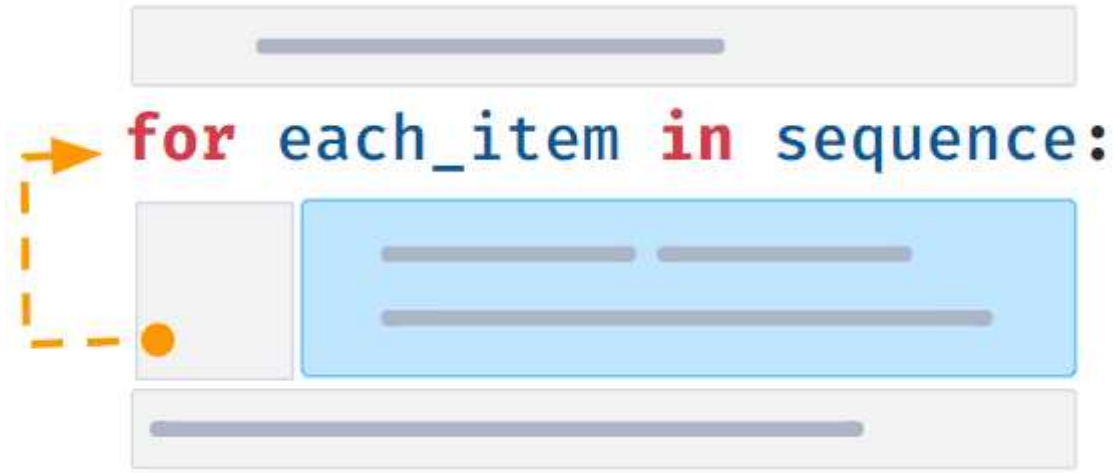
Output

```
1
2
3
```

6. What is the use of `for` loop?

The

`for` loop is used to execute a block of code a known number of times. The `for` statement iterates over each item of a sequence.



Examples of sequences:

- The sequence of Characters (string)
- The sequence of numbers, etc.

Code

PYTHON

```
word = "Python"
for each_char in word:
    print(each_char)
```

Output

```
P
y
t
h
o
n
```

7. What are `range()` and `xrange()` functions?

range():

The

`range()` function generates a sequence of integers starting from 0 to n(n is not included) and returns it.

Syntax:

`range(n)`

Code

PYTHON

```python
for number in range(3):
    print(number)
```

Output

```
0
1
2
```

Range with Start and End

Syntax:

`range(start, end)`

Generates a sequence of numbers starting from

`start` to `end` (`end` is not included).

Code

PYTHON

```python
for number in range(5, 8):
    print(number)
```

Output

```
5
6
7
```

xrange():

The

`xrange()` function generates a sequence of integers starting from 0 similar to `range()` function.

Code

PYTHON

```python
for number in xrange(4):
    print(number)
```

Output

```
0
1
2
3
```

Note: The `xrange()` function is deprecated from Python3. You cannot run `xrange()` function in our code playground.

8. When does an infinite loop occurs?

An infinite loop occurs when the condition always evaluates to `True` i.e. incorrect termination condition.

Example

Code

PYTHON

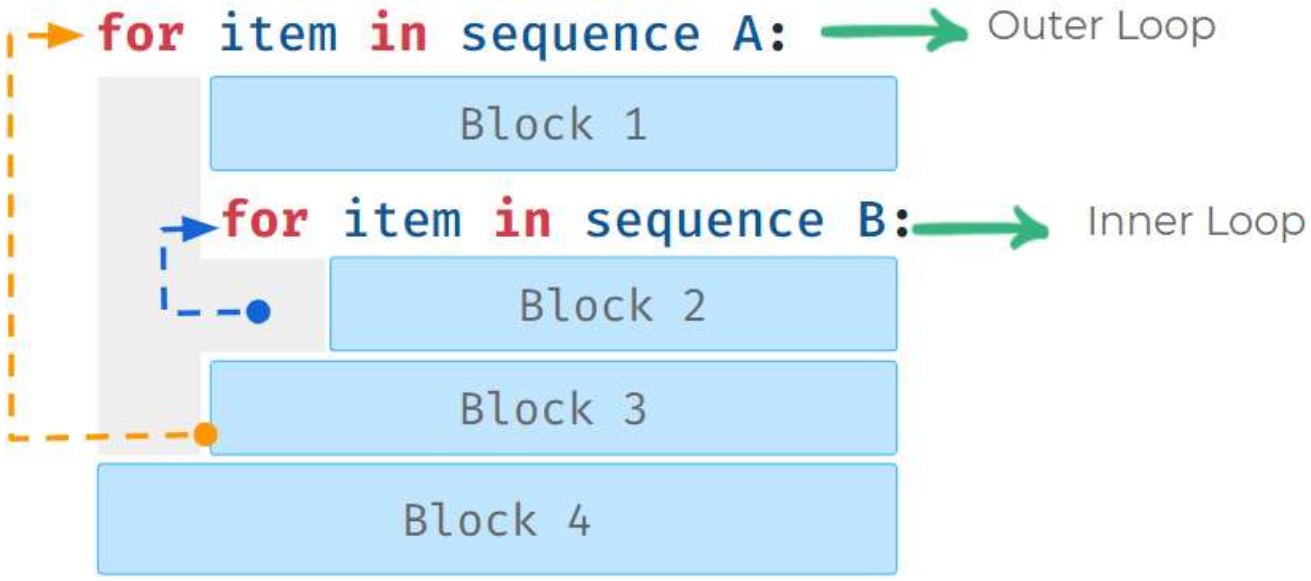
```python
a = 10
while a > 3:
    a = a + 1
    print(a)
```

In the above code the while loop will run infinite times as the condition always evaluates to `True` .

9. What are Nested Loops?

An inner loop within the repeating block of an outer loop is called Nested Loop.

The **Inner Loop** will be executed one time for each iteration of the **Outer Loop.**



Code

PYTHON

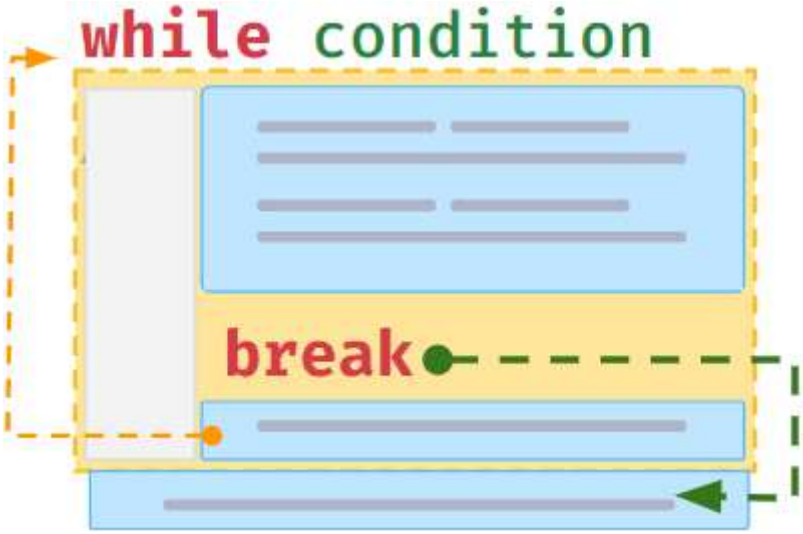
```python
for i in range(2):
    print("Outer: " + str(i))
    for j in range(2):
        print("  Inner: " + str(j))
```

Output

```
Outer: 0
  Inner: 0
  Inner: 1
Outer: 1
  Inner: 0
  Inner: 1
```

10. What is a break statement?/ How to exit from a loop?

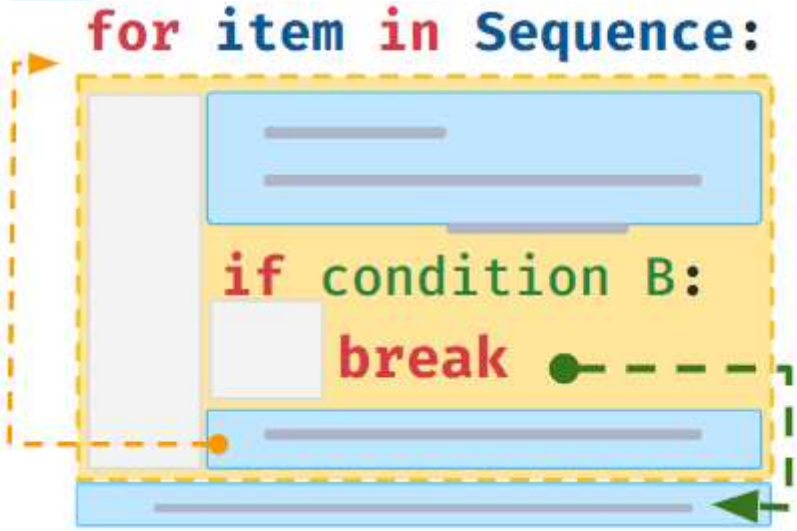
Break statement makes the program exit a loop early.



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` then 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")
```

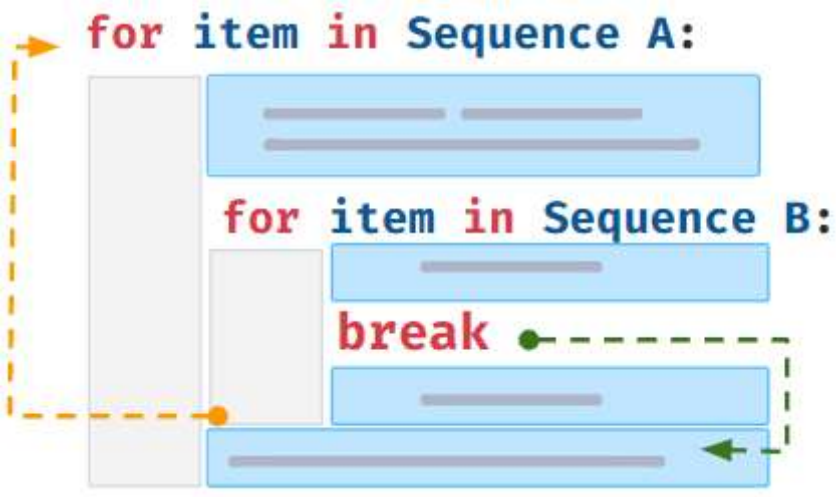
Output

```
0
1
2
END
```

Break in Nested Loop

The

`break` in the inner loop stops the execution of the inner loop.



Code

PYTHON

```python
for i in range(4):
    for j in range(4):
        if i * j > 10:
            break
    if (i > 0) and (j > 0):
        print(i * j)
```

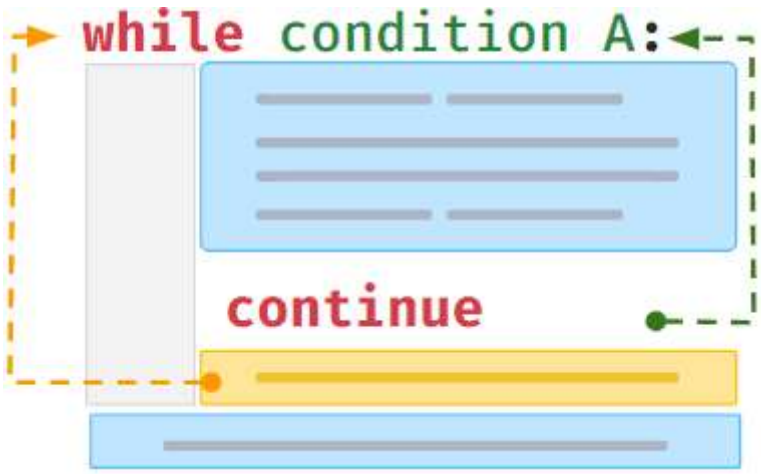
Output

```
3
6
9
```

11. What is `continue` in loops?

The

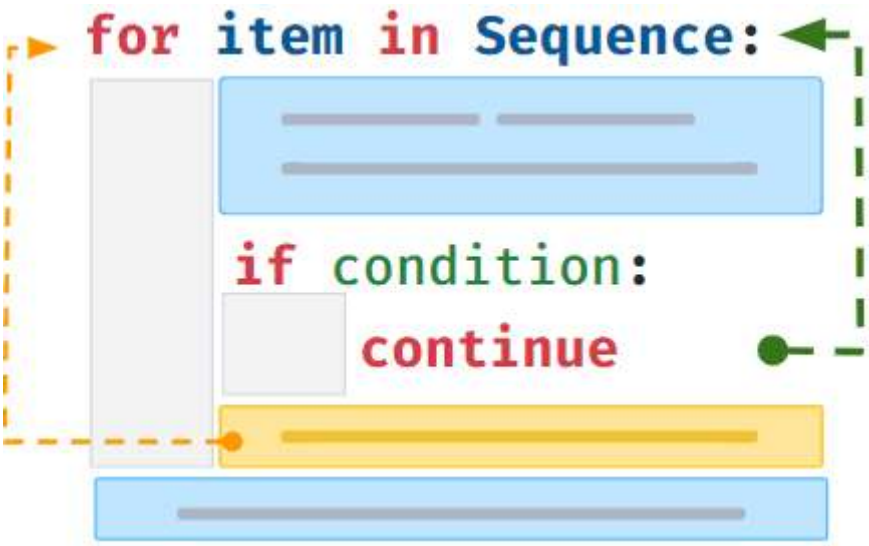
`continue` statement makes the program skip the remaining statements in the current iteration and begin the next iteration.



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` then the next statements in the loop body are skipped.

Code

PYTHON

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

Output

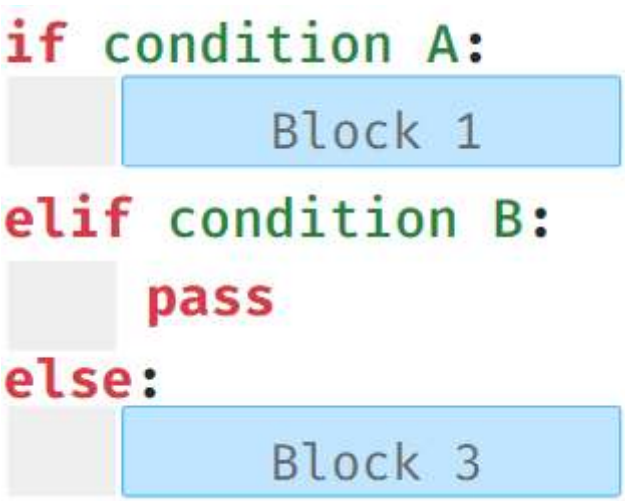
```
0
1
2
4
END
```

12. What is `pass` in Python?

The

`pass` statement is a syntactic placeholder. When it is executed, nothing happens.

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



Empty Loops

We can use

`pass` statements to test code written so far, before writing loop logic.

MARK AS COMPLETED

Submit Feedback

Notes Discussions