



Day 5 RPA Master Bootcamp

What are Loops

For loop

While Loop

For Loops in Python

Python For loop is used for sequential traversal i.e. it is used for iterating over an iterable like String, Tuple, List, Set, or Dictionary.

For Loops Syntax

for var in iterable:

 # statements

Looping through a string:

```
message = "Hello, World!"  
  
for char in message:  
    print(char)
```

Looping through a range of numbers:

```
for num in range(1, 6):  
    print(num)
```

A nested for loop in Python is a loop within another loop. It allows you to iterate over elements in multiple sequences or perform operations on multi-dimensional data structures. The inner loop runs completely for each iteration of the outer loop.

The basic syntax of a nested for loop is as follows:

for outer_variable in outer_sequence:

 # Outer loop code



for inner_variable in inner_sequence:

Inner loop code

```
for i in range(1, 6): # Outer loop runs from 1 to 5
    for j in range(1, 11): # Inner loop runs from 1 to 10
        print(f"{i} x {j} = {i * j}")
```

Print the table of 5

```
for i in range(1, 11):
    result = 5 * i
    print(f"5 x {i} = {result}")
```

While loop

In Python, a while loop is a control flow statement that allows you to repeatedly execute a block of code as long as a given condition is true. It's used when you want to perform a task repeatedly until a certain condition is no longer met.

The basic syntax of a while loop in Python is as follows:

while condition:

Code block to be executed while the condition is true

Here's an example of a simple while loop that prints numbers from 1 to 5:

```
count = 1
while count <= 5:
    print(count)
    count += 1
```

```
count = 1
while True:
    print(count)
    count += 1
    if count > 5:
        break
```



The main difference between for and while loops in Python is the way they control the flow of execution and the situations where they are most commonly used.

For Loop:

A for loop is used to iterate over a sequence (e.g., list, tuple, string, etc.) or other iterable objects.

It has a fixed number of iterations and is ideal when you know the number of times you want to repeat a block of code.

The loop variable takes on the value of each element in the sequence during each iteration.

The for loop syntax in Python is more concise and straightforward compared to other languages.

Example:

```
for i in range(1, 6): # Outer loop runs from 1 to 5
    for j in range(1, 11): # Inner loop runs from 1 to 10
        print(f"{i} x {j} = {i * j}")
```

While Loop:

A while loop is used to repeatedly execute a block of code as long as a given condition is true.

It has a flexible number of iterations and is suitable when you want to repeat a task until a certain condition is no longer met.

The loop continues until the condition becomes false, and it's essential to ensure that the condition eventually becomes false to avoid infinite loops.

The while loop syntax involves explicitly defining the condition and managing the iteration variable within the loop.

Example:



```
count = 1
while count <= 5:
    print(count)
    count += 1
```

In Python, continue and break are control flow statements used within loops to modify the flow of execution.

continue:

The continue statement is used inside a loop to skip the rest of the current iteration and move on to the next iteration immediately.

When continue is encountered, the remaining code within the loop for the current iteration is skipped, and the loop proceeds with the next iteration.

It is useful when you want to avoid executing certain parts of the loop's body based on specific conditions.

Example:

```
for i in range(1, 6):
    if i == 3:
        continue
    print(i)
```

break:

The break statement is used inside a loop to terminate the loop prematurely when a certain condition is met.

When break is encountered, the loop immediately exits, and the program continues with the next statements outside the loop.

It is helpful when you want to stop the loop early, based on some specific condition.

Example:



```
for i in range(1, 6):  
    if i == 4:  
        break  
    print(i)
```

Practice ;

```
# Control Flow  
# Loops  
# For  
# While  
  
num1 = 1  
num2 = 2  
num3 = 3  
num4 = 4  
num5 = 5  
num6 = 6  
num7 = 7  
  
for i in range(8):  
    for j in range(8):  
        if i == 2 or j == 2:  
            print("value is i and j are ",  
i, j)  
        elif i == 5 or j == 2:  
            print("value is i and j are ",  
i, j)  
        else:  
            print("in else case i and j are  
", i, j)
```

```
print("For loop on String")
rpa_boot_camp = "RPA Boot camp"
for i in rpa_boot_camp:
    print(i)

# continue
# break
sum_value = 0
for i in range(1, 5):

    sum_value = sum_value + i
    print("sum of values ", sum_value)

print("sum of all values ", sum_value)

print("While Loop")

# while loop
counter = 1
while counter <= 6:
    if counter == 4:
        continue
    if counter == 5:
        break
    print("counter ", counter)
    counter = counter + 1
```