



Python Programming

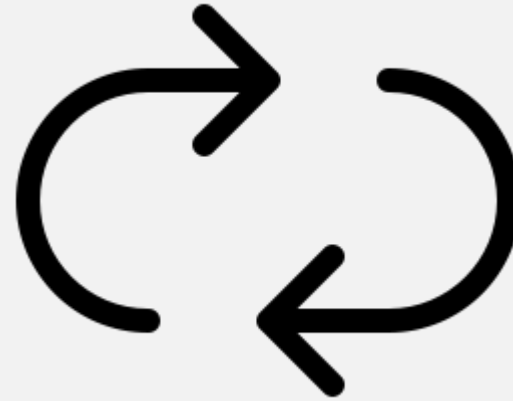
Lesson 6

Loopy Loop

Presented by Advaspire Team



Loops – Variation of Loops



Loops by definition just mean circular or oval shape that two ends will cross each other. In other word, we are repeating in a pattern.

Some example of loop that we can use in our day are listening to Spotify, watching Youtube videos and even when using the stove.



Variation of Loops in Python

There are 2 kind of loops, While and For which will have some functions to use.

While Loop



For Loop



While Loop

```
i = 1
while i<=10:
    print (i)
    i = i+1
```

```
C:\Users\haziq\PycharmProjects\
1
2
3
4
5
6
7
8
9
10
```

While Loop is a loop that will repeat while its in a True condition. Once it arrive to a False condition, it will stop repeating itself. The program will check condition first before executing.



Setting up the While Loop

In the example shown, the code after the indent will be considered inside the while loop. So anything outside the indent will not be considered while checking the condition.

indent

```
i = 1
while i<=10:
    print(i)
    i = i+1
```

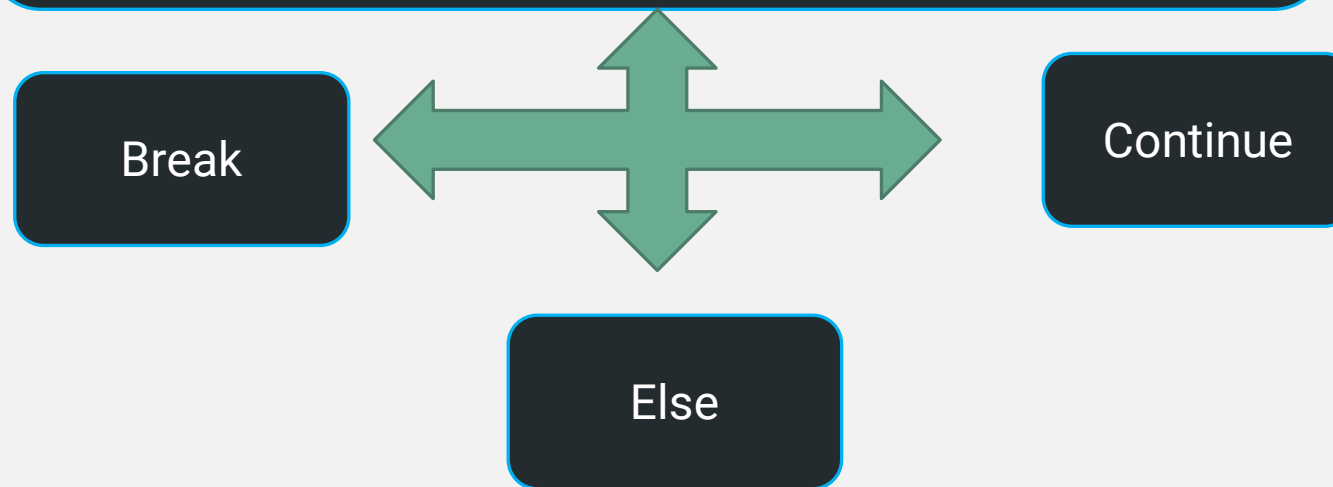
To start, create a variable with a value. This will act as the item we want to be repeated. Then on the second line, type like in the example. This will be the True condition. We can interpret it as it will only stop looping when the value reach 10 or less. So the false condition will be 11 and more.

The (i= i+1) is there to act as a stoppage of kind. The increment will be the trigger for false condition.



Statement for the While Loop

Statement in Loop is like the condition for the Loop to either continue looping or stop. There are several different statement that we can use .





Break and Continue

Break statement means you want the loop to stop looping at a specific position. This means you can stop it even if the condition is still True.

To use it, just add another line of code to specify if the sequence reach a number, it needs to break.

```
i = 1
while i<=10:
    print_(i)

    if (i== 5):
        break

    i+=1
```

```
C:\Users\haziq\
1
2
3
4
5
```

Continue statement means it will break the current iteration, then continue looping. So this is a bit different than the one for Break.

The variable used is also different, as it will only print the number after variable. For example, if you set 2 as the variable, then it will start printing at number 3. The number or value set as the continue trigger will also be skipped.

```
i = 0
while i < 10:
    i += 1
    if i== 4:
        continue
    print_(i)
```

```
1
2
3
5
6
7
8
9
10
```



Else Statement

Else statement means we can run a block of code once the condition is no longer True. Like in the example, once the i is more than 10, it will stop printing the (i) but continue with print the Else statement.

```
i = 0
while i < 10:
    i += 1
    print(i)
else:
    print('i is no longer more than 10')
```

```
1
2
3
4
5
6
7
8
9
10
i is no longer more than 10
```




For Loop

For loop is an iteration or repetition of a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

Unlike While loop, you don't need a variable with number to start. You can use almost any kind of data or storage structure.

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

```
C:\Users\haziq\i  
apple  
banana  
cherry
```



Setting Up the For Loop

Depending on what you want to iterate, you may need to create storage structure first like list or dictionary. Just make sure the value inside is in string.

Then write down the code like in the example. The x after for is like a placeholder so it can be anything. After that, you must include the in and the variable or list name. Only then it will print out the list in a sequence.

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```



String and Break Statement.

The way string is print in a For loop is a bit different that when you are using it in a storage structure.

When its only a string, the iteration will be for each letter of the string until the final letter.

```
for x in 'advaspire adcoins':  
    print_(x)
```

a
d
v
a
s
p
i
r
e

a
d
c
o
i
n
s



Break Statement

Break statement here is the same as in the While loop. So I'm just going to show you the example.

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)  
    if x == "banana":  
        break
```

```
C:\Users\haziq\Py  
apple  
banana
```



You can direct message your teacher and ask your question through [Slack Robotene Community](#) or arrange a [One-to-One Consultation](#) with your teacher.



Any Questions?



Thank you :)