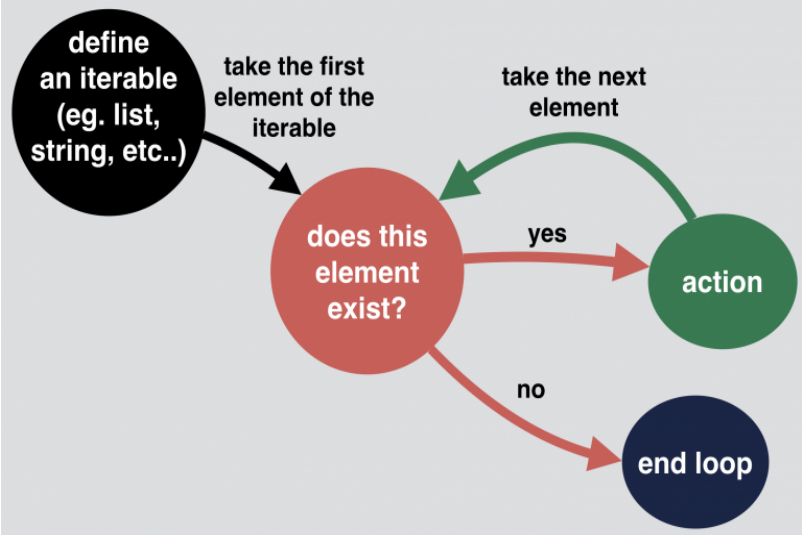
Loops

You go through your shopping list until you've collected every item from it. The dealer gives a card for each player until everyone has five.
The athlete does push-ups until reaching one-hundred... Loops everywhere! As for for loops in Python: they are perfect for processing repetitive programming tasks.

In Python, and all other modern programming languages, iteration statements (also called loops) allow a set of instructions to be repeatedly executed until a certain condition is reached. This condition may be predetermined (as in the for loop) or open-ended (as in the while and do-while loops). In another way we can say, A loop can b used to tell a program to execute statements repeatedly. Or we can say that a loop repeatedly executes the same set of instructions until a termination condition is met.



For Loop

Python implements an iterator-based 'for loop'. It is a type of 'for loop' that iterates over a list of items through ar explicit or implicit iterator.

The loop is introduced by the keyword 'for' which is followed by a random variable name

which will contain the values supplied by the object.

This is the syntax of Python's 'for loop':

```
for variable in list:
statements
else:
statements
```

Here is an example of a 'for loop' in Python:

```
pizza = ["New York Style Pizza", "Pan Pizza", "Thin n Crispy Pizza", "Stuffed Crust" ,"Pizza"]

for choice in pizza:
if choice == "Pan Pizza":
print("Please pay $16. Thank you!")
print("Delicious, cheesy " + choice)
else:
```

```
print("Cheesy pan pizza is my all-time favorite!")
print("Finally, I'm full!")
```

Run this and you'll get the following output on Python Shell:

```
Delicious, cheesy New York Style Pizza
Please pay $16. Thank you!
Delicious, cheesy Pan Pizza
Delicious, cheesy Thin n Crispy Pizza
Delicious, cheesy Stuffed Crust Pizza
Cheesy pan pizza is my all-time favorite!
Finally, I'm full!
```

Using the range() Function with the for Loop

The range() function can be combined with the 'for loop' to supply the numbers required by the loop. In the following example, the range(1, x+1) provided the numbers 1 to 50 needed by the 'for loop' to add the sum of 1 until 50:

```
x = 50
total = 0
for number in range(1, x+1):
total = total + number
print('Sum of 1 until %d: %d' % (x, total))
```

The Python Shell will display:

```
l.py
Sum of 1 until 50: 1275
```

The While Loop

A Python 'while loop' repeatedly carries out a target statement while the condition is true. The loop iterates as long as the defined condition is true. When it ceases to be true and becomes false, control passes to the first lin after the loop.

The 'while loop' has the following syntax:

```
while condition
statement
statement
```

Here is a simple 'while loop':

```
counter = 0
while (counter < 10):
print('The count is:' , counter)
counter = counter + 1
print('Done!')
```

If you run the code, you should see this output:

```
l.py
```

The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
The count is: 9
Done!