Using the FOR Statement

Using the FOR Statement

Python has a special type of loop to allow us to visit elements in an array (or any other collection of variables), and it's called the FOR statement, so it works in a similar way to the WHILE statement, but is more compact. All we have to do is create a variable that will store each element in the collection one at a time, and that will run all the commands inside the loop for each value, until there are no more elements in the collection. So, the general form of the FOR statement is as follows:

The FOR Statement

```
for (variable) in (collection):
        do some stuff
```

So. if we have our array as follows:

```
StudentsAges = [9, 10, 9, 8, 10, 10, 9, 11]
```

We could print out all of the elements of array as follows:

```
for AnyNameAtAll in StudentsAges:
    print(AnyNameAtAll)
# EndFor;
```

And we will get the following:

```
9
10
9
8
10
10
9
11
```

If we wanted to search to check if a particular value is in an array, we could do the following:

```
for AnyNameAtAll in StudentsAges:
    if (AnyNameAtAll == 8):
        print("Number 8 has been found")
    # Endif;
# EndFor;
```

This will visit each element of the array, and for each occurrence of the number eight (8), the program will print out a message:

```
Number 8 has been found
```

So if our program needs to visit each value in an array, or some other collection, the FOR statement is a nice, clear way of doing it.