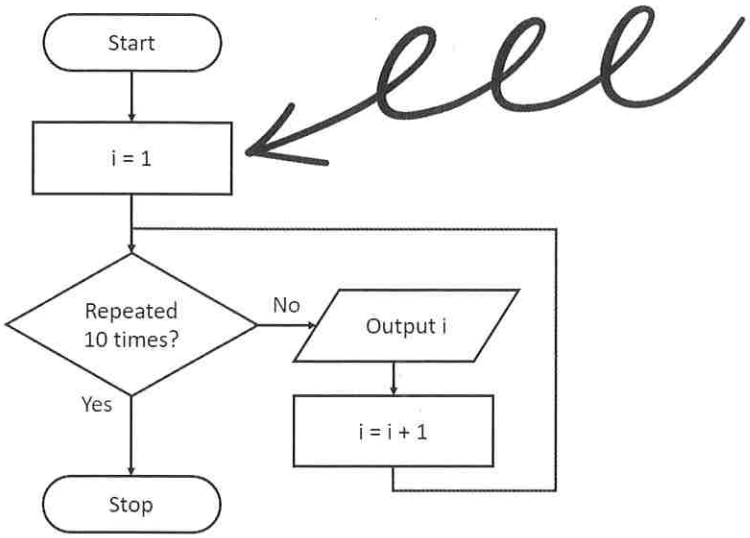
For Loop

Explanation

A **`for loop`** allows Python to keep repeating code a set number of times. It is sometimes known as a **`counting loop`** because you know the number of times the loop will run before it starts.



In this case, it starts at 1 and will keep repeating the loop (displaying i) until it reaches 10 and then stops. This is how this loop would look in Python

```
for i in range(1,10):
    print(i)
```

In this example, the outputs would be 1, 2, 3, 4, 5, 6, 7, 8 and 9. **When it gets to 10 the loop would stop so 10 would not be shown in the output.**

Remember to indent the lines of code within the for loop.

Example Code

The range function is often used in for loops and lists the starting number, the ending number and can also include the steps (e.g. counting in 1s, 5s or any other value you wish).

```
for i in range(1,10):
 print(i)
```

This loop uses a variable called "i" to keep track of the number of times the loop has been repeated. It will start i at 1 (as that is the starting value in the range function) and repeat the loop, adding 1 to i each time and displaying the value of i until it reaches 10 (as dictated by the range function), where it will stop. Therefore, it will not repeat the loop a tenth time and will only have the following output:

1, 2, 3, 4, 5, 6, 7, 8, 9

```
for i in range(1,10,2):
 print(i)
```

This range function includes a third value which shows how much is added to i in each loop (in this case 2). The output will therefore be: **1, 3, 5, 7, 9**

```
for i in range(10,1,-3):
 print(i)
```

This range will subtract 3 from i each time. The output will be: **10, 7, 4**

Using loops is a powerful programming tool that you will use a lot in the more challenging programs.

```
for i in word:
 print(i)
```

This would display each character in a string called "word" as a separate output (i.e. on a separate line).