Beginner's Guide to Loops in START Programming

Loops are essential constructs in programming that allow you to execute a block of code repeatedly. In START, you have two types of loops: while loops for conditional looping and for loops for iterating over arrays. Let's explore how to use each type of loop effectively.

While Loops

While loops are used for executing a block of code repeatedly as long as a condition is true. Lets look at the syntax of a while loop.

```
loop while condition {
    // Code block to execute
}
```

The code inside of the block will also execute as long as the condition is true. They are useful for running until a certain condition in the code has been met. Lets look at an example of a while loop:

```
i is 0

loop while i less than 10 {
    write(i) nl
    i is i + 1
}
write("done because i is now", 10) nl
```

This code produces the following output:

```
0
1
2
```

```
3
4
5
6
7
8
9
done because i is now 10
```

The block of code executed 10 times, as `i` started on 0, and once `i` was incremented from 9 up to 10, and the condition was checked again it failed, so the loop was then finished and could be skipped.

For Loops

For loops are used for iterating over arrays. They can be useful for checking elements of an array for a condition, or for accessing matrix (a list of lists) values. Lets look at the syntax of for loops.

```
loop for each item in arr {
    // Code block to execute for each item in the array
}
```

The code within the block will execute once for each item in the array, hence the code will execute for the length of the array. Lets see an example of this:

```
arr is [1,2,3]
loop for each element in arr{
    write(element) nl
}
```

For each run of the block, it outputs the elements of the array, the output of this code is:

```
1
2
```

```
3
```

Conclusion

Loops are powerful constructs in programming that allow you to automate repetitive tasks and iterate over collections of data. By understanding how to use while and for loops effectively, you'll be able to write more efficient and concise code. Practice using loops with different conditions and arrays to become proficient in controlling the flow of your programs.