

INTRO TO PROGRAMMING: WHILE LOOPS

WHAT IS A WHILE LOOP?

Imagine that you had a remote control humanoid robot. You could command the robot to take one step forward using a button on the remote. 🤖👣

Let's say that you wanted this robot to walk in a straight line along a beach 🏖️ until it reached the other end, roughly 1km away. You will need to press a button on your remote for every move you make, every step you take.

Doing this for an entire kilometre would get tiring, real fast. 🤯 What you really want to do is tell the robot to keep taking steps on its own until it reaches the end. This sounds like a job for some programming!

The problem is, assuming that each step is half a metre, you would have to add 2,000 step() functions/blocks to your code which is just as annoying! 😫

However, the bigger problem is that you don't know precisely how long the beach is and, therefore, the exact number of steps that the robot needs to take. The solution to this problem is to use a while loop. 🔄

A while loop is a piece of code that will continue to re-run itself while a specific condition is TRUE.

In the following example, you can compare code with and without a while loop. Without a while loop, you have to write the same function `Step()` 2,000 times, whereas with the while loop, you only have to write three lines of code (and add an additional function).

```

1  Step();
2  Step();
3  Step();
4  Step();
5  Step();
6  Step();
7  Step();
8  Step();
...
1989 Step();
1990 Step();
1991 Step();
1992 Step();
1993 Step();
1994 Step();
1995 Step();
1996 Step();
1997 Step();
1998 Step();
1999 Step();
2000 Step();

```

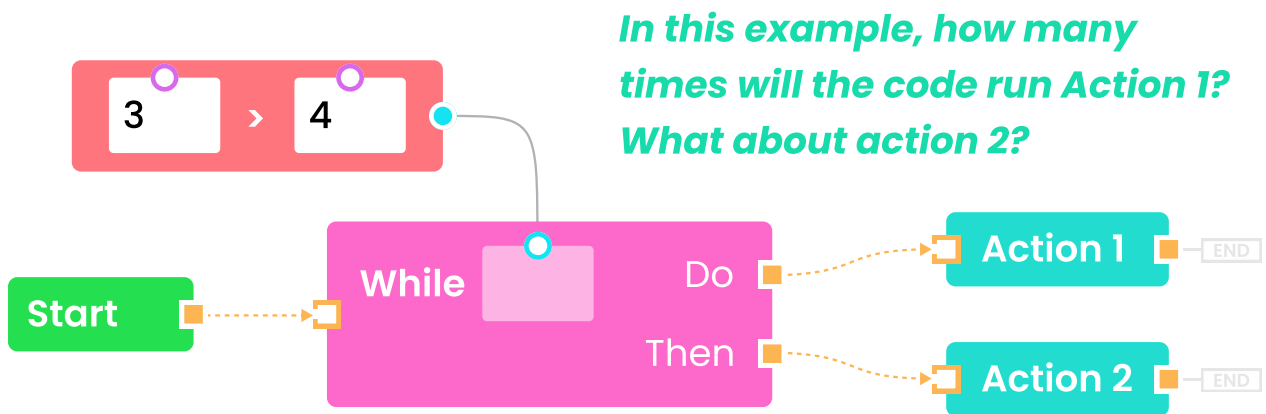
```

1  while (NotAtFinishLine()) {
2      step();
3  }

```

WHILE BLOCKS

We can create while loops using the WHILE flow block. This block continually re-runs the code connected to DO (Action 1) while the input condition is TRUE. It will continue repeating this code forever until the input condition becomes FALSE. When the condition does become FALSE, it will then run the code connected to THEN (Action 2).



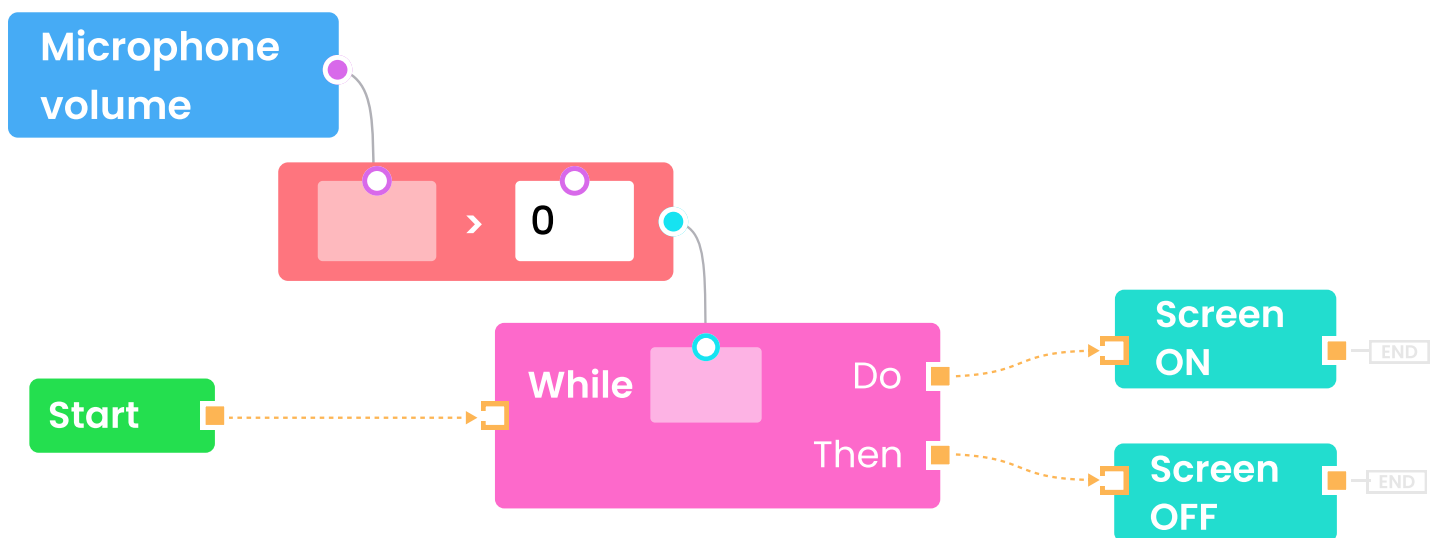
If the condition is FALSE to begin with, then Action 1 is never run. If the condition is TRUE to begin with, but never changes, then Action 1 will be repeated infinitely. This means that the condition must switch to FALSE at some point or the code will never stop running!

While loops are useful because they allow us to repeat a piece of code a dynamic number of times – the number of times that the code is repeated depends on the condition.

Let's say that you wanted to write a piece of code that would keep the screen of your mobile phone on while you were talking.

Because you don't know beforehand how long you will be talking (it will likely change every time), we can't just set the screen to stay on for X seconds.

Instead, we could continually check the output of the mobile phone's microphone.



While the volume of the sound detected by the microphone is greater than 0, keep the screen on. Then, when the output becomes 0, turn it off.