

# LOOPS (FOR AND WHILE)

## MOVING OBJECT

To get you started, we're going to begin with how to make an object moves. Let say in this case we want to make a clown moves.

**(code of the step)** This is to make the clown facing left, the number inside the bracket is the angle of Clown's movement to the left

```
♻️ clown → left turn(45)  
♻️ clown → forward(100)
```

Now try this one on your own! The clown will now moves in 45 degree to the left with the length of 100 moves. We are now wondering, if we want to make multiple moves, we need to write a lot of code to place the clown to move from one place to another, individually. But lucky for us, we have a shortcut called Loops.

There are different kind of loops, such as for and while. Let's start off with for loop.

## FOR LOOP

Here is the link to the tutorial on TouchDevelop. <http://tdev.ly/euxle>

For statement can be used to execute a line of code into a desired number of times. For example, rather than commanding the clown to moves 3 times to make an octagon, we can just use for loop and perform it 8 times:

This loop is to perform the action all over again

```
for 0 ≤ i < 8 do  
  The steps are split using a depth-first-search strategy.  
  ♻️ clown → left turn(45)  
  ♻️ clown → forward(100)  
  Run your program: This is how the clown moves  
end for
```

In this example, 'i' is acting as a random variable just to store how many time we want to perform the code. Now try it by yourself in your tablets!

## WHILE LOOP

Here is the link to the tutorial on TouchDevelop. <http://tdev.ly/acrfq>

We can implement the exact same thing as for loop by using a while loop. The only difference is while loop is using a stored variable on top of the actual loop. We'll start introducing the variable, then perform the while loop

```
var index := 7
while index ≥ 0 do
  ⌚ turtle2 → left turn(45)
  ⌚ turtle2 → forward(100)
  index := index - 1
end while
end action
```

Whoops, but wait, we want an 8-sided trail, why is it only 7? Well, take a look at the symbol there. We start counting from 0 instead of 1. Now try it by yourself and try out countless shapes that you can think of!