|  |
| --- |
| Circle Language Spec: Execution Control |

## For (conditional)

There are three forms of For loop, as mentioned in the article *For*.

This article explains the form of For loop originally intended for going through a range of values, but more flexible than that. Unlike the *range* For this kind of For is a *conditional* loop.

The initial value of the range is replaced by an initialization procedure, which could do anything, but usually it assigns an initial value to a loop variable.

The last value of the range is replaced by a condition, which could be any condition, but usually it is the condition of the loop variable staying below or equal to the last value of the range.

The step to take going through the range of numbers is replaced by an action, which can be *any command*, but it usually is a procedure, that increments the loop variable with 1.

The For command takes a command reference to the Initialization procedure, which is called once, at the beginning of the loop. It also takes a command reference to the Action, which is a command called at the end of each repetition of the loop. The For command also takes a reference to another command, which is called the Loop command reference, which is the actual loop to repeat. The For command also takes a Boolean Condition, that has to stay True for the loop to keep repeating.

The Condition is usually passed to the For command as a reference to a reference to a Boolean. It needs to be a reference to a reference, because the condition needs to be recalculated every time it is consulted. Why making it a reference to a reference solves that problem is explained in the second last paragraph of the article *Execution Control*, part of which I repeat here:

In a conditional *loop*, the condition must be re-evaluated on every repetition of the loop. Recalculation of the condition, every time the condition is consulted, can be established by making the condition a reference to a reference to a Boolean, rather than a reference to a specific Boolean object.

So the condition argument refers to a *reference*.

The reference, that is referred to, can perform a calculation before it returns the Boolean. It performs the calculation every time the reference is consulted. How a reference can recalculate the value of an object whenever it is retrieved, is explained by the article *System Interface*.

The implementation of this kind of For command uses a machine instruction to jump back in code to call the Loop again. For the rest it just calls the Loop and the Action command references and evaluates the Condition to determine whether to run the loop again.