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

## Loop-Related Jumps

### Concept

Loop-related jumps are jumps used specifically inside a loop. Jumps are a form of *execution control*. Loops are explained in the article *Loops*. Execution control is explained in the article *Execution Control*.

The following two forms of jump relate specifically to a loop:

- Exit Loop

- Continue

They each will be explained in their own article.

### Diagram

Loops-related jumps are a form of execution control explained in the article *Loop-Related jumps*. The articles that follow only explain their expression in a diagram.

There are two forms of loop-related jump:

- Exit Loop

- Continue

Each form is explained in a separate article. See the articles *Exit Loop in a Diagram*, *Continue in a Diagram.*

### Continue

#### Concept

The Continue statement is a loop-related jump. It is a jump that is only performed inside a loop.

Continue skips the remainder of the loop procedure and immediately moves on to the next repetition of a loop.

A loop procedure is implemented as a reference to a command, passed as an argument to an execution control command. So Continue means no more, than to exit the command that is looped, so it is a synonym for Exit Command, but then used inside a loop.

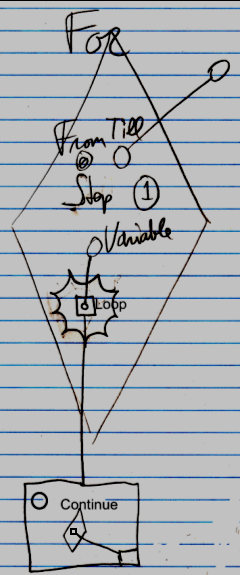
It will remove the command that is exited from the call stack, and immediately jumps to the command referred to by the Command End, that was passed to it.

#### Diagram

The concept of Continue was already covered by the article *Continue*. This article only explains its expression in a diagram.

As explained in the article *Continue*, the Continue command means no more than to exit the procedure that is looped.

So in the expression in a diagram, you see a loop, that has the command to loop associated to it. A continue statement looks the same as to exit a command:



### Exit Loop

#### Concept

The Exit Loop statement is a loop-related jump. It is a jump that is only performed inside a loop.

Exit Loop ommits all repetitions, that would have followed and immediately ends the complete loop.

Exit Loop is a jump to the Command End of the loop command. The Command End of a command is a reference to the next command to run, so the normal order that follows after the loop continues.

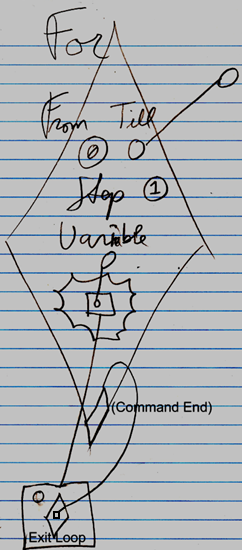
Do not confuse this with exiting the command that is *being* looped, because that causes the next repetition of the loop to run. That action is performed with the Continue command. The Exit Loop command ends the *whole loop*.

The implementation of the Exit Loop command is a lot like the implementation of Exit Command, only the Exit Loop command will delete the last *two* calls from the call stack, and immediately jumps to the command referred to by the Command End, that was passed to it.

#### Diagram

The concept of Exit Loop was explained the article *Exit Loop*. The current article only explains its expression in a diagram.

The Exit Loop command is displayed inside the clause that is looped as a jump to the Command End of the whole loop.



The text Command End is not required in the diagram:

