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

## Execution Control in a Diagram

The concept of execution control is explained in the article *Execution Control*. The current article demonstrates its expression in a diagram.

The articles, that will follow, explain the diagram expression separately for each execution control statement. But to explain their expression in general, the If statement is used as an example here.

An If statement executes a command if a certain condition is met. If the condition is not met, then an alternate command can be run.

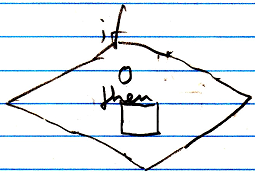
A textual expression of an If statement can look like this:

If Condition Then Command

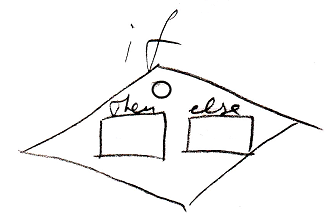
Or:

If Condition Then Command A Else Command B

In a diagram this looks as follows:

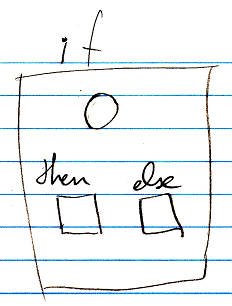


Or:



The circle is the condition. The word *condition* does not have to be shown in the diagram. The Then and the Else are two command references, that are passed to the If command.

The If statement is drawn out like a diamond shape. A diamond shape stands for a *call*. It is a *call* to an execution control command. The definition of the execution control commands is part of a system module of execution control commands. It looks like this:



The definitions of the execution control commands are squares. But usually you only see *calls* to execution control commands, with a diamond shape.

The Condition and the Then and Else commands are shown right inside the call. But you are likely to want to see the Then and Else commands defined *outside* the If call.

|  |  |
| --- | --- |
|  |  |

You can do this by defining references to the clauses on a higher level. This will automatically connect the clauses inside the If symbol to the clauses outside the If symbol. A clause inside the If symbol and a clause outside the If symbol are actually both no more than references the same clause. So this will create references to the same clause on multiple levels. Following the rules of automatic containment, references to the same thing are tied together with lines. The symbol on the highest containment level represents the command itself. Putting a reference an a higher level for the sole purpose of elevating the definition of something, up to a level it is not required to be defined in, is called an *esthetic reference*.

See the articles *Esthetic Reference* and *Automatic Containment* for more about those subjects.

It is a coincidence, that a diamond was already used, to express a *decision* in many types of control-flow diagram, that already existed before this new computer language. It turns out, that using the existing syntax for calling commands, passing along command references as parameters, works perfectly for a clear way to express execution control.