|  |
| --- |
| Circle Language Spec: Commands |

## Sub-Commands

Executions of commands can run inside a parent command. Sub-commands are *active* commands contained inside a *parent* command.

A parent command automatically executes its sub-commands. After a sub-command completes, the process returns to the parent command, which will then continue, executing the next sub-command.

Inside a command, usually just more commands are invoked.

There are only a few commands that do something other than execute other commands. Those are special commands, that perform a machine instruction: an operation that is executed by the CPU, the central processing unit of the computer. On top of those special commands, a few basic commands exist, like If’s and For loops, that control the flow of a program, making the next command to call dependent on a condition.

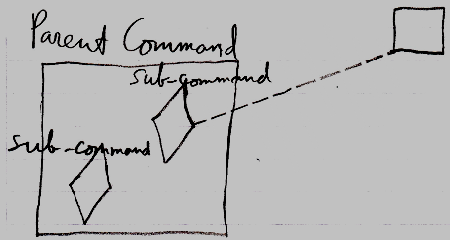
But basically, a command just calls more commands. Machine instructions, arithmetic operators, comparative and boolean algebra, assignments, and execution control statements such as If and For, are *all just commands*.

Apart from sub-commands, a command can also contain data.  
(Inactive clauses and inactive command references for instance are also considered data, and are not sub-commands, because they do not execute.)

### Sub-Commands in a Diagram

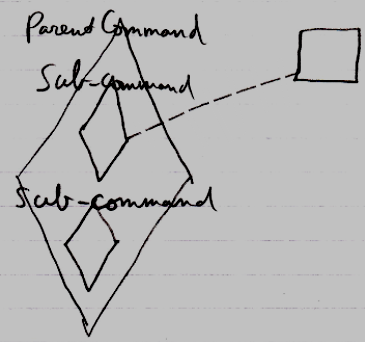
Sub-commands are active commands contained inside a parent command. You can encounter them inside any command symbol: both active and inactive commands symbols can contain sub-commands. The sub-commands are *active* commands: command calls, active clauses or active command references.

Below is a picture of two sub-commands inside an inactive command:



The Parent Command is an inactive command, because it is a square. In the diagram above, the Parent Command contains two sub-commands. The sub-commands are displayed as diamond shapes. One of the sub-commands is a call, because it has a dashed line going outside the Parent Command, tying the sub-command to its command definition. The other sub-command is an *active clause*, because it does not redirect its command definition.

Below is a picture of two sub-commands inside an active command:



It is the same picture as the other diagram, only now the Parent Command is an executable command symbol, not an inactive command.