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

## Command Definition

### Concept

A command definition is a blue-print for another command.

There is a complete analogy between a command’s definition and the class of an object.

Anything that applies to classes, also applies to command definitions.

A command can select another command to function as its *prototype*. This prototype is called the *command definition*. The command definition describes the procedure of the command and how to link objects to the command. Commands with the same definition contain the same list of attributes, related items and related lists.

A command object will have the same structure as its definition, but not the same data. The values of the attributes can freely change for each individual command object. *Which* objects are referenced is also different for each individual command object. But initially the command object will be an exact replica of the definition. The definition’s attribute values and object references only function as a default.

A command, that is used as a definition, is usually not executable. Only individual calls to the command, that use the command as a definition, are executable. If a command definition is executable after all, then it is clearly mentioned, that it is an *active* command definition, because it is a special situation.

### Diagram Notation

Any command object can serve as another command’s protype. So any command symbol can be the prototype for another object:

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

But usually, it is a square, that is another command’s definition:



When an *object* is drawn with a dashed line, then it is only used as a class:



This notation will *not* be copied to the concept of commands. This is because a command’s definition will usually be a square and a square will usually be a command’s definition, so using dashed lines for command definitions, would create an overload of dashed squares. So command symbols, that are only used as a definition, do not get a dashed notation.