|  |
| --- |
| Circle Language Spec: Parameters |

## Required & Optional

Some programming languages have a thing called required and optional parameters. In other programming languages basically all parameters are required, unless they are typed optional. In the new computer language it is the other way around: any parameter is optional unless it is typed required. This is because in the new computer language commands are the same as objects, that just happen to be executable, and the parameters inside a command are analogus to sub-objects inside another object, whose filling-in is also totally arbitrary.

Some parameters can be made required, so you *have to* fill them in as input. In the new computer language, this means that reading and writing any accessible aspect of a parameter can be separately made required or optional. When output is required, it means you have to use the output. You will be unable to execute the command without picking up the return value or returned object.

Basically each system command can be made required or optional. Usually you make access from the *outside* required. When you make access from the *inside* required, this can give the user of the command guarantees about the parameter’s usage.

## Ideas

### Out of the original Symbol documentation

* Just as in procedures, you could make certain members of a type required, while others are optional. You have to fill in the required members on creation of the object. This should be there for the same reason as required parameters are there in a procedure: the function of the object just doesn’t make sense unless you write the required members. The programmer is made extra aware of that by making the members required.