LiveCode Overview

Basic LiveCode Vocabulary

The following LiveCode language vocabulary is only a small portion of the scripting language. However, even this small vocabulary can get you a long way. I suggest using this core vocabulary as a springboard toward learning to understand the basic structure and syntax of the LiveCode scripting language. Once you understand how these work you can easily find other language elements in the LiveCode Dictionary and experiment with them in your stacks.

Note: In the list below, text in *italics* are place holders to indicate the type of data or reference that should appear in that position in the statement. For example, use of the `move` command might look like this:

```
move button "mybtn" from 100,100 to 600,400 in 1 second
```

Object Types
- stack
- card
- button
- field
- image
- graphic
- group

Messages
- mouseDown
- mouseUp
- (pre)OpenCard

Commands
- hide/show *object*
- enable/disable *object*
- put *text string* into | before | after *container*
- grab *object*
- move *object* from *xy coordinate* to *xy coordinate* in *time duration*
- set the *property* of *object* to *value*
- wait *time duration*
- go *card | stack*

Functions (can be expressed in two forms)
- the date *-or-* date()
- the time *-or-* time()
- the random of *integer* *-or-* random(*integer*)

Control Structures

Message handle*r*
```
on message
   statements
end message
```

if-then-else structure
```
if condition then
   statements
else
   statements
end if
```

repeat loop
```
repeat with variable = lower limit to upper limit
   statements
end repeat
```

Keywords
- me
- the target

Properties

the location	the visible
the name	the enabled
the short name	

Variables - containers for holding information or data
- **Naming rules**:
- Any combination of letters, numbers and underscore (_).
- Must start with letter or _
- Must not be the same as a LiveCode language reserved word (i.e., any word used for other purposes.
- Create a variable by putting something into it:
  ```
  put "Hello World." into theVar
  ```
- **Scope**:
- local *variable name* - variable is recognized in all handlers in script
- global *variable name* - variable is recognized in all scripts where global is declared