Lists

Director provides a special kind of variable called a list to allow storing sets of information in a group. There are two basic types of lists: linear lists and property lists. Linear lists are just a series of values of any kind contained within the list variable. Property lists are also a series of elements, but each element is actually a pairing of a property and a value associated with that property.

For example:

```lisp
set linearlist = [ 1, 99, "tree"]   --value types can be mixed
set anotherlinearlist = [ the member of sprite 15, the number of member "button" ]

set propertyList = [ #name: "Janice", #age: 25, #title: "designer" ]

Note that the property labels are preceded by a # sign. Technically, this makes them symbols.
```

Syntax for linear lists:

```lisp
set LIST = [ item1, item2, item3 ... ]
set LIST = list (item1, item2, item3 ... )
set LIST = [] --establishes an empty linear list
```

Lingo for manipulating linear lists:

```lisp
add (LIST, VALUE) - end of list if unsorted; in order if sorted (linear)
append (LIST, VALUE) - add to end of list (linear, script error w/prop. list)
addAt (LIST, POS, VALUE) - add to specified position. Following members shift. (linear)
deleteAt (LIST, POS) - remove value at position (linear & prop.)
getAt (LIST, POS) - data at position POS (linear & prop.)
getLast (LIST) - last item (linear & prop.)
getOne (LIST, VALUE) - position of VALUE in list (linear & prop.)
getPos (LIST, VALUE) - position of VALUE in list (linear)
setAt (LIST, POS, VALUE) - replace data at position POS (linear)
sort (LIST) - alpha/numeric sort (linear & prop.)
count (LIST) - number of items in list (linear & prop.)
max (LIST) - highest alpha/numeric value in list (linear & prop.)
min (LIST) - lowest alpha/numeric value in list (linear & prop.)
ilk (LIST) - #list or #propList (linear & prop.)
```

Duplicating lists:

```lisp
set listB = listA -- makes a "link" - changes to listA will occur to listB
set listB = duplicate (listA)-- makes a copy - lists are independent
```
Property lists:
  set PLIST = [ #label:value, #label:value, ... ]
  set PLIST = [:]

Property list Lingo:
  addProp (PLIST, PROP, VALUE) - add new entry; at end if unsorted, in order if sorted
  deleteProp (PLIST, PROP) - delete data associated with property PROP.
  CASE-SENSITIVE for property names.
  getProp (PLIST, PROP) - value associated with PROP.
  getPropAt (PLIST, POS) - property name at position POS.
  getAProp (PLIST, PO SorPROP) - value associated with PO S or PROP.
  setAProp (PLIST, PROP, VALUE) - replace data associated with PROP.
    If PROP not in PLIST, adds it.
  setProp (PLIST, PROP, VALUE) - replace data associated with PROP.
    If PROP not in PLIST, error.
  findPos (PLIST, PROP) - position in list where property appears.
  findPosNear (PLIST, PROP) - same as findPos, except when PROP doesn't exist - then
    returns closest position based on sort order.