Commonly-Used Lingo

DIRECTOR 6 syntax	DIRECTOR 7 "dot" syntax	DESCRIPTION/EXAMPLE
Important Lingo Commands:		
updateStage		Forces Director to update the screen display.
put		Retrieves the value of specified item or expression.
set		Sets the specified variable or property to a value.
alert "alert text"		Displays an alert box with the specified text.
the frame		Current frame
go [to] <location>		Execution skips to specified frame or marker
go the frame		Loop on the current frame
go [to] <location> of movie <movieName>		Execution skips to specified location (frame or marker) of the specified Director movie
go loop		Execution skips to prior marker or start of movie
go next/previous		Execution skips to next/previous marker
play member <memberName/memberNum>		Plays the specified SWA (Shockwave) audio member.
Play [frame <frameName/frameNum>] [of movie <movieName>]		Execution skips to specified location (frame or marker), optionally of a specified Director movie. If "play done" is encountered, execution resumes from where the original "play" call was made.
play done		Resumes execution from where an original "play" call was made.
exit		Leaves the currently executing handler.
halt		Stops the execution of the movie
quit		Exits the movie back to the desktop
puppetTransition <# or transition cast member>		Specifies the transition for the next screen update (see list of codes)
label (labelName)		number of frame with the given labelName
the labelList		list of marker names
the frameLabel		the name of the marker on the current frame, or 0 if there is no marker.
the runMode		Returns a string representing the mode the program is running in: "Author", "Projector", "Plugin"
the machineType		Returns a code representing the type of computer the program is running on - unique number for Macs, 256 for all PCs (see table)
the platForm		Returns a string representing the type of computer the program is running on - "Macintosh,68k", "Macintosh,PowerPC", "Windows,16", "Windows,32"

Events:

on prepareMovie on startMovie on stopMovie		handlers executed just before a movie executes (prepareMovie), or when a movie starts or stops
on prepareFrame on enterFrame on exitFrame		handlers executed just before a frame is executed (prepareFrame), or when a frame is entered or exited

Sprites:

puppetSprite	puppetSprite	turns on puppeting for the given sprite
the puppet of sprite	sprite(x).puppet	the puppeted state for the given sprite
the member of sprite	sprite(x).member	member reference for sprite (member x of castLib y)
the memberNum of sprite	sprite(x).memberNum	member's number in its cast library
the castLibNum of sprite	sprite(x).castLibNum	number of the cast library itself
the height/width of sprite	sprite(x).height/width	height or width in pixels of the sprite
the left/top/right/bottom of sprite	sprite(x).left/top/right/bottom	location of left, top, right, bottom of sprite in pixels from top left corner (0,0) of Stage.
the rect of sprite	sprite(x).rect	the rectangle which describes the sprite (left, top, right, bottom)
the stretch of sprite	sprite(x).stretch	stretched state of sprite (TRUE or FALSE)
the visible of sprite	sprite(x).visible	visible state of sprite (TRUE or FALSE)
the moveableSprite of sprite	sprite(x).moveableSprite	if TRUE, a sprite is draggable by the user; else, the sprite is not draggable.
the loc/locH/locV of sprite	sprite(x).loc/locH/locV	stage dimensions and coordinates: upper left = (0,0), upper right = (640, 0), lower left = (0, 480), lower right= (640,480)
rollover (spriteNum)	rollover (spriteNum)	TRUE if cursor is over the given sprite, else FALSE
the rollover	the rollover	number of the uppermost sprite the cursor is over
intersects	intersects	if sprite x **intersects** sprite y then ...

Mouse Lingo:

the mouseH/the mouseV	the mouseH/the mouseV	current position of the mouse (cursor)
the mouseCast the mouseMember	the mouseCast the mouseMember	number of cast member currently under the mouse
on mouseEnter on mouseLeave on mouseWithin	on mouseEnter on mouseLeave on mouseWithin	handlers triggered when the cursor enters, leaves, or is within the bounding box of the sprite to which the script is attached.
on mouseDown/Up/UpOutside	on mouseDown/Up/UpOutside	handlers triggered when the mouse is pressed, released, or released outside the sprite to which the script is attached.
the stillDown	the stillDown	TRUE as long as the mouse button is down.
on rightMouseDown/Up	on rightMouseDown/Up	handlers executed when the right mouse button [Control-click on the Mac] is pressed or released. Only works on Mac if emulateMultiButtonMouse property is TRUE.
emulateMultiButtonMouse	emulateMultiButtonMouse	determines whether Director interprets a mouse click with the Control key pressed on the Macintosh the same as a right mouse click in Windows (TRUE) or not (FALSE).
the mouseChar/Word/Item/Line	the mouseChar/Word/Item/Line	returns the number (sequential) of the character, word, item or line currently under the cursor. Works only for fields; returns -1 if not over a field.

More Sprite Stuff:

the editable of sprite	sprite(x).editable	TRUE if the sprite can be edited by the user at runtime. Applies only to fields.
the cursor of sprite	sprite(x).cursor	the cursor setting of the given sprite: 0 (none - system default), -1 (pointer), 1 (I-beam), 2 (crosshair), 3 (crossbar), 4 (watch), 200 (invisible); custom: cast members and masks ([x, y], 1-bit, 16x16 pixels)
the constraint of sprite	sprite(x).constraint	the sprite that defines the limit of movement of another, moveable, sprite. Example: set the contraint of sprite x to y. Has a value of 0 if no constraint is in effect.
contrainH/constrainV	contrainH/constrainV	nearest coordinate to contain sprite in horiz/vert direction. syntax: constrainH (spriteNum, horiz/vertcoordinate) To keep sprite x within sprite y's horiz (or vert) coordinate: set the locH of sprite x = constrainH(y, the mouseH)
the tweened of sprite	sprite(x).tweened	TRUE if 1st frame of sprite is a keyframe; FALSE if all frames are keyframes
the ink of sprite	sprite(x).ink	ink effect code of the sprite (see list)
the type of sprite	sprite(x).type	always 16 for any sprite, or 0 if no sprite in channel
the type of member	member(x).type	symbol representing member type (see list)
the trails of sprite	sprite(x).trails	TRUE if trails is turned on, else FALSE.
the blend of sprite	sprite(x).blend	the blend (transparency) value of the sprite, 0 to 100
the forecolor of sprite the backcolor of sprite	sprite(x).forecolor sprite(x).backcolor	the foreground (text) or background color value for the sprite at the current color depth. Note that numerical color values are different for different color depths.
the linesize of sprite	sprite(x).linesize	width of border (in pixels) for shape sprites

Useful in behaviors:

the spriteNum of me	the spriteNum of me	the number of the sprite that triggered the behavior
the currentSpriteNum	the currentSpriteNum	the number of the sprite that triggered the behavior

Digital video sprites:

the movieTime of sprite	sprite(x).movieTime	current position in the video, in ticks
the movieRate of sprite	sprite(x).movieRate	playback rate: 1=forward normal speed, 0=stopped, -1=backward normal speed. Fractional and accelerated speeds are valid.
the startTime of sprite the stopTime of sprite	sprite(x).startTime sprite(x).stopTime	sets default start and stop times for video
the duration of member	member(x).duration	duration of video (in ticks)
the sound/video of member	member(x).sound/.video	TRUE if sound/video track of the video is enabled
the trackCount of sprite the trackCount of member	sprite(x).trackCount member(x).trackCount	number of tracks in specified video sprite or member

trackType (sprite x, trackNum) trackType (member x, trackNum)	trackType (sprite x, trackNum) trackType (member x, trackNum)	symbol representing the track type of the given sprite or member: #video, #sound, #text, #music (MIDI)
trackEnabled (sprite x, trackNum)	trackEnabled (sprite x, trackNum)	TRUE if the given track of the video is enabled
the volume of sprite	sprite(x).volume	volume of video sprite: 0 to 255 (also audio sprites)
the mostRecentCuePoint of sprite the mostRecentCuePoint of sound	sprite(x).mostRecentCuePoint sound(x).mostRecentCuePoint	cue point last passed (ordinal value); 0 if none

Working with Variables

showGlobals	showGlobals	Displays a list of all global variables and their values
showLocals	showLocals	Displays a list of all the local variables for the current handler, and their values. (Useful only when stopped at a breakpoint in the debugger.)
local variables global variables		Locals variables are known only in the handler in which they are defined. Global variables are known throughout the movie.
boolean items		Boolean variables or expressions have a value of TRUE or FALSE
property	property	A property is a kind of local variable used within behaviors.
intP/floatP/stringP/symbolP listP/voidP	intP/floatP/stringP/symbolP listP/voidP	Tests the type of the specified item. example: **intP(x)** returns TRUE if the variable "x" is an integer.
constants		special names containing fixed values in Director: TRUE, FALSE, EMPTY, SPACE, BACKSPACE, QUOTE, RETURN, ENTER