## Commonly-Used Lingo

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

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

**Sprites:**

- `puppetSprite` turns on puppeting for the given sprite.
- `sprite(x).puppet` the puppeted state for the given sprite.
- `sprite(x).member` member reference for sprite (member x of castLib y).
- `sprite(x).memberNum` member's number in its cast library.
- `sprite(x).castLibNum` number of the cast library itself.
- `sprite(x).height/width` height or width in pixels of the 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.
- `sprite(x).rect` the rectangle which describes the sprite (left, top, right, bottom).
- `sprite(x).stretch` stretched state of sprite (TRUE or FALSE).
- `sprite(x).visible` visible state of sprite (TRUE or FALSE).
- `sprite(x).moveableSprite` if TRUE, a sprite is draggable by the user; else, the sprite is not draggable.
- `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).
- `spriteNum` rollover (TRUE if cursor is over the given sprite, else FALSE).
- `sprite(x).rollover` rollover number of the uppermost sprite the cursor is over.
- `sprite(x).intersects` if sprite x intersects sprite y then ...

**Mouse Lingo:**

- `mouseH/the mouseV` current position of the mouse (cursor).
- `mouseCast` number of cast member currently under the mouse.
- `mouseMember` handlers triggered when the cursor enters, leaves, or is within the bounding box of the sprite to which the script is attached.
- `mouseDown/Up/UpOutside` handlers triggered when the mouse is pressed, released, or released outside the sprite to which the script is attached.
- `stillDown` TRUE as long as the mouse button is down.
- `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` determines whether Director interprets a mouse click with the Control key press on the Macintosh the same as a right mouse click in Windows (TRUE) or not (FALSE).
- `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  
\text{sprite(x).editable}  
TRUE if the sprite can be edited by the user at runtime. Applies only to fields.

the cursor of sprite  
\text{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  
\text{sprite(x).constraint}  
the sprite that defines the limit of movement of another, moveable, sprite. Example: set the constraint of sprite \text{x} to \text{y}. Has a value of 0 if no constraint is in effect.

\text{constrainH/constrainV}  
\text{constrainH/constrainV}  
nearest coordinate to contain sprite in horiz/vert direction. Syntax: \text{constrainH} (\text{spriteNum}, \text{horiz/vertcoordinate})

To keep sprite \text{x} within sprite \text{y}'s horiz (or vert) coordinate:
set the loc\text{H} of sprite \text{x} = \text{constrainH(y, the mouseH)}

the tweened of sprite  
\text{sprite(x).tweened}  
TRUE if 1st frame of sprite is a keyframe; FALSE if all frames are keyframes

the ink of sprite  
\text{sprite(x).ink}  
in effect code of the sprite (see list)

the type of sprite  
\text{sprite(x).type}  
always 16 for any sprite, or 0 if no sprite in channel

the type of member  
\text{member(x).type}  
symbol representing member type (see list)

the trails of sprite  
\text{sprite(x).trails}  
TRUE if trails is turned on, else FALSE.

the blend of sprite  
\text{sprite(x).blend}  
the blend (transparency) value of the sprite, 0 to 100

the forecolor of sprite  
\text{sprite(x).forecolor}  
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 backcolor of sprite  
\text{sprite(x).backcolor}  

the linesize of sprite  
\text{sprite(x).linesize}  
width of border (in pixels) for shape sprites

Useful in behaviors:

the spriteNum of me  
\text{the spriteNum of me}  
the number of the sprite that triggered the behavior

the currentSpriteNum  
\text{the currentSpriteNum}  
the number of the sprite that triggered the behavior

Digital video sprites:

the movieTime of sprite  
\text{sprite(x).movieTime}  
current position in the video, in ticks

the movieRate of sprite  
\text{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  
\text{sprite(x).startTime}  
sets default start and stop times for video

the stopTime of sprite  
\text{sprite(x).stopTime}  
duration of video (in ticks)

the duration of member  
\text{member(x).duration}  
TRUE if sound/video track of the video is enabled

the sound/video of member  
\text{member(x).sound/.video}  
number of tracks in specified video sprite or member


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

**Working with Variables**

showGlobals  
showLocals  
local variables  
global variables  
boolean items  
property  
intP/floatP/stringP/symbolP  
listP/voidP  
constants

Displays a list of all global variables and their values
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.)
Locals variables are known only in the handler in
which they are defined. Global variables are known
throughout the movie.
Boolean variables or expressions have a value of
TRUE or FALSE
A property is a kind of local variable used within
behaviors.
Tests the type of the specified item. example:

`intP(x)` returns TRUE if the variable “x” is an
integer.
special names containing fixed values in Director:
TRUE, FALSE, EMPTY, SPACE, BACKSPACE,
QUOTE, RETURN, ENTER