

Objects Covered Cheatsheet

Part 12

Object []	What it does	Example Arguments (where applicable)
[hi]	Human interface device. Let's you get information from a gamepad, Wacom tablet, etc. and use it in MaxMSP. Connect the Right outlet to a [umenu], then send [hi] the message (menu) to populate it. Connect the [umenu] middle outlet to the [hi] and select your device. Send [hi] a (poll) with a time - like (poll 10) to poll every X number of MS. (poll 0) turns it off. Output will be a list of the controller and the value. VERY useful for creating custom physical controllers.	
[udpreceive]	Receives messages over a network (yes, including the internet). Arguments are the host IP address (IPv4 format) and the port. Essential for working with OSC.	[udpreceive 127.0.0.1 8000]
[mira.status]	Requires the Miraweb externals. Gets a list of all clients connected to Mira.	
[mira.frame]	Broadcasts Mira compatible UI objects to the Mira iPad app, or to a web browser.	
[textedit]	Creates a UI space that can be used to enter text. If you send it a (bang), it outputs the text. Using the (set) message sets the text without outputting it. Useful if not immediately obvious.	
[OSC-route]	ESSENTIAL OBJECT FOR OSC! Requires the CNMAT externals. It's very similar to the [route] object, but specifically for OSC addresses, including full URL-style slash delimiting, letting you dramatically simplify how Max handles OSC messages.	[OSC-route /iPhone/TouchOSC/XY1]
[patcherargs]	Generates initial arguments in a subpatch or abstraction. Set the arguments to what you want the values to be. It will output them as a list.	[patcherargs 10. 100.]
[mstoamps~]	Converts MS to samples at the current sample rate. VERY useful for changing time formats for things like [delay~] or [slide~].	
[slide~]	Filters a signal logarithmically between a change in signal values. This lets you use the output to follow the envelope of the incoming signal.	