IPython – An enhanced Interactive Python – Quick Reference Card

`obj?, obj??`	Get help, or more help for object (also works as ?obj, ??obj).
`?foo.*abc*`	List names in 'foo' containing 'abc' in them.
`%magic`	Information about IPython's 'magic' % functions.

Magic functions are prefixed by %, and typically take their arguments without parentheses, quotes or even commas for convenience.

Example magic function calls:

`%alias d ls -F`	'd' is now an alias for 'ls -F'
`alias d ls -F`	Works if 'alias' not a python name
`alist = %alias`	Get list of aliases to 'alist'
`cd /usr/share`	Obvious. cd -<tab> to choose from visited dirs.
`%cd??`	See help AND source for magic %cd

System commands:

`!cp a.txt b/`	System command escape, calls os.system()
`cp a.txt b/`	after %rehashx, most system commands work without !
`cp ${f}.txt $bar`	Variable expansion in magics and system commands
`files = !ls /usr`	Capture sytem command output
`files.s, files.l, files.n`	"a b c", ['a','b','c'], 'a\nb\nc'

History:

`_i, _ii, _iii`	Previous, next previous, next next previous input
`_i4, _ih[2:5]`	Input history line 4, lines 2-4
`exec _i81`	Execute input history line #81 again
`%rep 81`	Edit input history line #81
`_, __, ___`	previous, next previous, next next previous output
`_dh`	Directory history
`_oh`	Output history
`%hist`	Command history. '%hist -g foo' search history for 'foo'

Autocall:

`f 1,2`	f(1,2)
`/f 1,2`	f(1,2) (forced autoparen)
`,f 1 2`	f("1","2")
`;f 1 2`	f("1 2")

Remember: TAB completion works in many contexts, not just file names or python names.

The following magic functions are currently available:

%Exit	Exit IPython without confirmation.
%Pprint	Toggle pretty printing on/off.
%Quit	Exit IPython without confirmation (like %Exit).
%alias	Define an alias for a system command.
%autocall	Make functions callable without having to type parentheses.
%autoindent	Toggle autoindent on/off (if available).
%automagic	Make magic functions callable without having to type the initial %.
%bg	Run a job in the background, in a separate thread.
%bookmark	Manage IPython's bookmark system.
%cd	Change the current working directory.
%clear	Clear various data (e.g. stored history data)

%color_info	Toggle color_info.
%colors	Switch color scheme for prompts, info system and exception handlers.
%cpaste	Allows you to paste & execute a pre-formatted code block from clipboard.
%debug	Activate the interactive debugger in post-mortem mode.
%dhist	Print your history of visited directories.
%dirs	Return the current directory stack.
%doctest_mode	Toggle doctest mode on and off.
%ed	Alias to %edit.
%edit	Bring up an editor and execute the resulting code.
%env	List environment variables.
%exit	Exit IPython, confirming if configured to do so.
%hist	Alternate name for %history.
%history	Print input history (_i<n> variables), with most recent last.
%logoff	Temporarily stop logging.
%logon	Restart logging.
%logstart	Start logging anywhere in a session.
%logstate	Print the status of the logging system.
%logstop	Fully stop logging and close log file.
%lsmagic	List currently available magic functions.
%macro	Define a set of input lines as a macro for future re-execution.
%magic	Print information about the magic function system.
%p	Just a short alias for Python's 'print'.
%page	Pretty print the object and display it through a pager.
%pdb	Control the automatic calling of the pdb interactive debugger.
%pdef	Print the definition header for any callable object.
%pdoc	Print the docstring for an object.
%pfile	Print (or run through pager) the file where an object is defined.
%pinfo	Provide detailed information about an object.
%popd	Change to directory popped off the top of the stack.
%profile	Print your currently active IPyhton profile.
%prun	Run a statement through the python code profiler.
%psearch	Search for object in namespaces by wildcard.
%psource	Print (or run through pager) the source code for an object.
%pushd	Place the current dir on stack and change directory.
%pwd	Return the current working directory path.
%pycat	Show a syntax-highlighted file through a pager.
%quickref	Show a quick reference sheet
%quit	Exit IPython, confirming if configured to do so (like %exit)
%r	Repeat previous input.
%rehash	Update the alias table with all entries in $PATH.
%rehashx	Update the alias table with all executable files in $PATH.
%rep	Repeat a command, or get command to input line for editing
%reset	Resets the namespace by removing all names defined by the user.
%run	Run the named file inside IPython as a program.
%runlog	Run files as logs.
%save	Save a set of lines to a given filename.
%sc	Shell capture - execute a shell command and capture its output.
%store	Lightweight persistence for python variables.
%sx	Shell execute - run a shell command and capture its output.
%system_verbose	Set verbose printing of system calls.
%time	Time execution of a Python statement or expression.
%timeit	Time execution of a Python statement or expression
%unalias	Remove an alias
%upgrade	Upgrade your IPython installation
%who	Print all interactive variables, with some minimal formatting.
%who_ls	Shell capture - execute a shell command and capture its output.
%store	Lightweight persistence for python variables.
%sx	Shell execute - run a shell command and capture its output.
%system_verbose	Set verbose printing of system calls.
%time	Time execution of a Python statement or expression.
%timeit	Time execution of a Python statement or expression
%unalias	Remove an alias
%upgrade	Upgrade your IPython installation
%who	Print all interactive variables, with some minimal formatting.
%who_ls	Return a sorted list of all interactive variables.
%whos	Like %who, but gives some extra information about each variable.
%xmode	Switch modes for the exception handlers.