

IPython – An enhanced Interactive Python – Quick Reference Card

<code>obj?, obj??</code>	Get help, or more help for object (also works as <code>?obj, ??obj</code>).
<code>?foo.*abc*</code>	List names in 'foo' containing 'abc' in them.
<code>%magic</code>	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:

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

System commands:

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

History:

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

Autocall:

<code>f 1,2</code>	<code>f(1,2)</code>
<code>/f 1,2</code>	<code>f(1,2)</code> (forced autoparen)
<code>,f 1 2</code>	<code>f("1","2")</code>
<code>;f 1 2</code>	<code>f("1 2")</code>

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

The following magic functions are currently available:

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

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