

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 % or %, and typically take their arguments without parentheses, quotes or even commas for convenience. Line magics take a single % and cell magics are prefixed with two %.

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
%timeit x=10      : time the 'x=10' statement with high precision.
%%timeit x=2**100 : time 'x**100' with a setup of 'x=2**100'; setup code is not
x**100           : counted. This is an example of a cell magic.
```

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 system 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 of current session.
%hist -g foo     : Search command history of (almost) all sessions for 'foo'.
%hist -g         : Command history of (almost) all sessions.
%hist 1/2-8      : Command history containing lines 2-8 of session 1.
%hist 1/ ~2/     : Command history of session 1 and 2 sessions before current.
%hist ~8/1-~6/5  : Command history from line 1 of 8 sessions ago to
                  line 5 of 6 sessions ago.
%edit 0/         : Open editor to execute code with history of current session.
```

Autocall:

```
f 1,2            : f(1,2) # Off by default, enable with %autocall magic.
/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:

```
%alias:
    Define an alias for a system command.
%alias_magic:
    ::
%autoawait:
```

```
%autocall:
    Make functions callable without having to type parentheses.
%automagic:
    Make magic functions callable without having to type the initial %.
%autosave:
    Set the autosave interval in the notebook (in seconds).
%bookmark:
    Manage IPython's bookmark system.
%cat:
    Alias for `!cat`
%cd:
    Change the current working directory.
%clear:
    Clear the terminal.
%colors:
    Switch color scheme for prompts, info system and exception handlers.
%conda:
    Run the conda package manager within the current kernel.
%config:
    configure IPython
%connect_info:
    Print information for connecting other clients to this kernel
%cp:
    Alias for `!cp`
%debug:
    ::
%dhist:
    Print your history of visited directories.
%dirs:
    Return the current directory stack.
%doctest_mode:
    Toggle doctest mode on and off.
%ed:
    Alias for `%edit`.
%edit:
    Bring up an editor and execute the resulting code.
%env:
    Get, set, or list environment variables.
%gui:
    Enable or disable IPython GUI event loop integration.
%hist:
    Alias for `%history`.
%history:
    ::
%killbgscripts:
    Kill all BG processes started by %script and its family.
%ldir:
    Alias for `!ls -F -o --color %l | grep /$`
%less:
    Show a file through the pager.
%lf:
    Alias for `!ls -F -o --color %l | grep ^-`
%lk:
    Alias for `!ls -F -o --color %l | grep ^l`
%ll:
    Alias for `!ls -F -o --color`
%load:
    Load code into the current frontend.
%load_ext:
    Load an IPython extension by its module name.
%loadpy:
    Alias of `%load`
%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.
%ls:
    Alias for `!ls -F --color`
%lsmagic:
    List currently available magic functions.
%lx:
    Alias for `!ls -F -o --color %l | grep ^-..x`
%macro:
    Define a macro for future re-execution. It accepts ranges of history,
%magic:
    Print information about the magic function system.
%man:
    Find the man page for the given command and display in pager.
%matplotlib:
    ::
%mkdir:
    Alias for `!mkdir`
%more:
    Show a file through the pager.
%mv:
    Alias for `!mv`
%notebook:
    ::
%page:
    Pretty print the object and display it through a pager.
%pastebin:
    Upload code to dpaste's paste bin, returning the URL.
%pdb:
    Control the automatic calling of the pdb interactive debugger.
%pdef:
    Print the call signature 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.
%pinfo2:
    Provide extra detailed information about an object.
%pip:
    Run the pip package manager within the current kernel.
%popd:
    Change to directory popped off the top of the stack.
%pprint:
    Toggle pretty printing on/off.
%precision:
    Set floating point precision for pretty printing.
%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.
%pypcat:
    Show a syntax-highlighted file through a pager.
%pylab:
    ::
%qtconsole:
    Open a qtconsole connected to this kernel.
%quickref:
    Show a quick reference sheet
%recall:
    Repeat a command, or get command to input line for editing.
```

```
%rehashx:
    Update the alias table with all executable files in $PATH.
%reload_ext:
    Reload an IPython extension by its module name.
%rep:
    Alias for `%recall`.
%rerun:
    Re-run previous input
%reset:
    Resets the namespace by removing all names defined by the user, if
%reset_selective:
    Resets the namespace by removing names defined by the user.
%rm:
    Alias for `!rm`
%rmdir:
    Alias for `!rmdir`
%run:
    Run the named file inside IPython as a program.
%save:
    Save a set of lines or a macro to a given filename.
%sc:
    Shell capture - run shell command and capture output (DEPRECATED use !).
%set_env:
    Set environment variables. Assumptions are that either "val" is a
%store:
    Lightweight persistence for python variables.
%sx:
    Shell execute - run shell command and capture output (!! is short-hand).
%system:
    Shell execute - run shell command and capture output (!! is short-hand).
%tb:
    Print the last traceback.
%time:
    Time execution of a Python statement or expression.
%timeit:
    Time execution of a Python statement or expression
%unalias:
    Remove an alias
%unload_ext:
    Unload an IPython extension by its module name.
%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.
%xdel:
    Delete a variable, trying to clear it from anywhere that
%xmode:
    Switch modes for the exception handlers.
%%!:
    Shell execute - run shell command and capture output (!! is short-hand).
%%HTML:
    Alias for `%%html`.
%%SVG:
    Alias for `%%svg`.
%%bash:
    %%bash script magic
%capture:
    ::
%debug:
    ::
%%file:
    Alias for `%%writefile`.
%html:
    ::
%javascript:
    Run the cell block of Javascript code
%js:
```

```
Run the cell block of Javascript code
%%latex:
    Render the cell as a block of latex
%%markdown:
    Render the cell as Markdown text block
%%perl:
    %%perl script magic
%%prun:
    Run a statement through the python code profiler.
%%pypy:
    %%pypy script magic
%%python:
    %%python script magic
%%python2:
    %%python2 script magic
%%python3:
    %%python3 script magic
%%ruby:
    %%ruby script magic
%%script:
    ::
%%sh:
    %%sh script magic
%%svg:
    Render the cell as an SVG literal
%%sx:
    Shell execute - run shell command and capture output (!! is short-hand).
%%system:
    Shell execute - run shell command and capture output (!! is short-hand).
%%time:
    Time execution of a Python statement or expression.
%%timeit:
    Time execution of a Python statement or expression
%%writefile:
    ::
```