

01.02-Shell-Keyboard-Shortcuts

February 27, 2017

This notebook contains an excerpt from the [Python Data Science Handbook](#) by Jake VanderPlas; the content is available [on GitHub](#).

The text is released under the [CC-BY-NC-ND license](#), and code is released under the [MIT license](#). If you find this content useful, please consider supporting the work by [buying the book!](#)

< [Help and Documentation in IPython](#) | [Contents](#) | [IPython Magic Commands](#) >

1 Keyboard Shortcuts in the IPython Shell

If you spend any amount of time on the computer, you’ve probably found a use for keyboard shortcuts in your workflow. Most familiar perhaps are the Cmd-C and Cmd-V (or Ctrl-C and Ctrl-V) for copying and pasting in a wide variety of programs and systems. Power-users tend to go even further: popular text editors like Emacs, Vim, and others provide users an incredible range of operations through intricate combinations of keystrokes.

The IPython shell doesn’t go this far, but does provide a number of keyboard shortcuts for fast navigation while typing commands. These shortcuts are not in fact provided by IPython itself, but through its dependency on the GNU Readline library: as such, some of the following shortcuts may differ depending on your system configuration. Also, while some of these shortcuts do work in the browser-based notebook, this section is primarily about shortcuts in the IPython shell.

Once you get accustomed to these, they can be very useful for quickly performing certain commands without moving your hands from the “home” keyboard position. If you’re an Emacs user or if you have experience with Linux-style shells, the following will be very familiar. We’ll group these shortcuts into a few categories: *navigation shortcuts*, *text entry shortcuts*, *command history shortcuts*, and *miscellaneous shortcuts*.

1.1 Navigation shortcuts

While the use of the left and right arrow keys to move backward and forward in the line is quite obvious, there are other options that don’t require moving your hands from the “home” keyboard position:

Keystroke	Action
Ctrl-a	Move cursor to the beginning of the line
Ctrl-e	Move cursor to the end of the line
Ctrl-b or the left arrow key	Move cursor back one character
Ctrl-f or the right arrow key	Move cursor forward one character

1.2 Text Entry Shortcuts

While everyone is familiar with using the Backspace key to delete the previous character, reaching for the key often requires some minor finger gymnastics, and it only deletes a single character at a time. In IPython there are several shortcuts for removing some portion of the text you're typing. The most immediately useful of these are the commands to delete entire lines of text. You'll know these have become second-nature if you find yourself using a combination of Ctrl-b and Ctrl-d instead of reaching for Backspace to delete the previous character!

Keystroke	Action
Backspace key	Delete previous character in line
Ctrl-d	Delete next character in line
Ctrl-k	Cut text from cursor to end of line
Ctrl-u	Cut text from beginning of line to cursor
Ctrl-y	Yank (i.e. paste) text that was previously cut
Ctrl-t	Transpose (i.e., switch) previous two characters

1.3 Command History Shortcuts

Perhaps the most impactful shortcuts discussed here are the ones IPython provides for navigating the command history. This command history goes beyond your current IPython session: your entire command history is stored in a SQLite database in your IPython profile directory. The most straightforward way to access these is with the up and down arrow keys to step through the history, but other options exist as well:

Keystroke	Action
Ctrl-p (or the up arrow key)	Access previous command in history
Ctrl-n (or the down arrow key)	Access next command in history
Ctrl-r	Reverse-search through command history

The reverse-search can be particularly useful. Recall that in the previous section we defined a function called `square`. Let's reverse-search our Python history from a new IPython shell and find this definition again. When you press Ctrl-r in the IPython terminal, you'll see the following prompt:

```
In [1]:  
(reverse-i-search) `':
```

If you start typing characters at this prompt, IPython will auto-fill the most recent command, if any, that matches those characters:

```
In [1]:  
(reverse-i-search) `sqa': square??
```

At any point, you can add more characters to refine the search, or press Ctrl-r again to search further for another command that matches the query. If you followed along in the previous section, pressing Ctrl-r twice more gives:

```
In [1]:
(reverse-i-search)`sqa': def square(a):
    """Return the square of a"""
    return a ** 2
```

Once you have found the command you're looking for, press Return and the search will end. We can then use the retrieved command, and carry-on with our session:

```
In [1]: def square(a):
    """Return the square of a"""
    return a ** 2
```

```
In [2]: square(2)
Out[2]: 4
```

Note that Ctrl-p/Ctrl-n or the up/down arrow keys can also be used to search through history, but only by matching characters at the beginning of the line. That is, if you type **def** and then press Ctrl-p, it would find the most recent command (if any) in your history that begins with the characters `def`.

1.4 Miscellaneous Shortcuts

Finally, there are a few miscellaneous shortcuts that don't fit into any of the preceding categories, but are nevertheless useful to know:

Keystroke	Action
Ctrl-l	Clear terminal screen
Ctrl-c	Interrupt current Python command
Ctrl-d	Exit IPython session

The Ctrl-c in particular can be useful when you inadvertently start a very long-running job.

While some of the shortcuts discussed here may seem a bit tedious at first, they quickly become automatic with practice. Once you develop that muscle memory, I suspect you will even find yourself wishing they were available in other contexts.

< [Help and Documentation in IPython](#) | [Contents](#) | [IPython Magic Commands](#) >