Autoreload of modules in IPython [duplicate]

Ask Question

This question already has an answer here:
Reloading submodules in IPython *8 answers*

Is there a way to have IPython automatically reload all changed code? Either before each line is executed in the shell or failing that when it is specifically requested to. I'm doing a lot of exploratory programming using IPython and SciPy and it's quite a pain to have to manually reload each module whenever I change it.

python ipython

edited May 12 '12 at 14:29
minrk
24.1k 5 69 68

asked Dec 15 '09 at 14:55
Thomas Parslow
2,151 3 18 28

marked as duplicate by Trevor Boyd Smith, Paul Sweatte, Sherif, Ami Tavory python Sep 23 '16 at 21:05

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

1 Here it's implemented as an extension projects.scipy.org/ipython/ipython/ticket/154 – Facundo Casco Dec 15 '09 at 15:09

You might consider changing the accepted answer. – Peque Sep 25 '15 at 8:55

I think it's usually fair to accept the first answer that answers the question, it seems a bit mean to change it years later! – Thomas Parslow Sep 28 '15 at 9:58

It is fine, Thomas. From What does it mean when an answer is "accepted"?: *Accepting an answer is not meant to be a definitive and final statement indicating that the question has now been answered perfectly. It simply means that the author received an answer that worked for him or her personally. Not every user comes back to accept an answer, and of those who do, they might not change the accepted answer even if a newer, better answer comes along later.* – fedorqui Aug 11 '17 at 14:55

6 Answers

REVISED - please see Andrew_1510's answer below, as IPython has been updated.

...

It was a bit hard figure out how to get there from a dusty bug report, but:

It ships with IPython now!

```
import ipy_autoreload
%autoreload 2
%aimport your_mod

# %autoreload? for help
```

... then every time you call `your_mod.dwim()` , it'll pick up the latest version.

edited May 23 '17 at 12:26
Community ♦
1 1

answered Jan 22 '11 at 0:12
Mike McCabe
867 8 8

4 What if it is less direct? `%run sometest.py` contains `import themod` . After editing `themod.py` , I'd like to just `%run sometest.py` , but it doesn't pick up the changes. – Jed May 22 '11 at 8:20

2 I think ipython 0.11 did away with this feature. Or is it just renamed/hidden someplace? – SirVer Aug 1 '11 at 8:51

SirVer, you're right. Sigh. Evidently, it's in the 'quarantine' package: archlinux.org/packages/community/any/ipython/files – Mike McCabe Aug 19 '11 at 5:51

Explanation here - with an invitation to port to 0.11 :) 'from IPython.quarantine import ipy_autoreload' succeeds, and creates an %autoreload command... but in my initial tests, it doesn't seem to work. – Mike McCabe Aug 19 '11 at 5:58

Looks like it's back in as of 9/30/2011 - so maybe we'll see it in an upcoming release. github.com/ipython/ipython/pull/746 – Mike McCabe Nov 15 '11 at 4:35

For IPython version 3.1, 4.x, and 5.x

```
In [1]: %load_ext autoreload

In [2]: %autoreload 2
```

```
Docstring:
``autoreload`` is an IPython extension that reloads modules
automatically before executing the line of code typed.

This makes for example the following workflow possible:

.. sourcecode:: ipython

   In [1]: %load_ext autoreload

   In [2]: %autoreload 2

   In [3]: from foo import some_function

   In [4]: some_function()
   Out[4]: 42

   In [5]: # open foo.py in an editor and change some_function to return 43

   In [6]: some_function()
   Out[6]: 43

The module was reloaded without reloading it explicitly, and the
object imported with ``from foo import ...`` was also updated.
```

There is a trick: when you **forget all** of the above when using `ipython` , just try:

```
import autoreload
?autoreload
# Then you get all the above
```

edited Feb 21 '17 at 18:07
Cal Yun
7 5

answered May 6 '12 at 17:37
Andrew_1510
4,765 7 36 44

Is there a way to do this in `ipdb` ? Say, I am in ipd, and I notice a line didnt work. So I changed the line, and want to reload the file. Will this work? – alpha_989 Mar 30 at 16:49

As mentioned above, you need the `autoreload` extension. If you want it to automatically start every time you launch `ipython` , you need to add it to the `ipython_config.py` startup file:

It may be necessary to generate one first:

```
ipython profile create
```

Then include these lines in `~/.ipython/profile_default/ipython_config.py` :

```
c.InteractiveShellApp.exec_lines = []
c.InteractiveShellApp.exec_lines.append('%load_ext autoreload')
c.InteractiveShellApp.exec_lines.append('%autoreload 2')
```

As well as an optional warning in case you need to take advantage of compiled Python code in `.pyc` files:

```
c.InteractiveShellApp.exec_lines.append('print "Warning: disable autoreload in
ipython_config.py to improve performance." ')
```

edit: the above works with version 0.12.1 and 0.13

edited Apr 12 '13 at 16:49

answered Dec 19 '12 at 21:16
kara deniz
1,591 13 14

1 This is actually great. I was wondering why no one else was posting solutions to preserve it. Does this work with older versions of IPython as well? I've been using 0.12+. I recall that the way ipython stores customizations changed significantly. – Ehtesh Choudhury Dec 27 '12 at 23:55

I'm using 0.12.1, and haven't yet tried 0.13, so I don't know whether it will work with 0.13+ – kara deniz Jan 2 '13 at 18:12

6 This is a good approach, but I think all you need to do is fill in the extentions which should be around line 27: `c.InteractiveShellApp.extensions = ['autoreload']` – dvreed77 May 16 '13 at 16:15

6 use `c.InteractiveShellApp.extensions = ['autoreload']` , and `c.InteractiveShellApp.exec_lines = ['%autoreload 2']` . I am not sure but in the default profile of version 0.13 under Ubuntu 13.04 I found a 'startup' folder that contains a script '50_autoreload.ipy' to activate autoreload. Maybe nothing is required at all – spinxz May 28 '13 at 17:41

I have to find this answer on any new install, this is the only sane config for development in iPython. – dashesy Sep 14 '13 at 19:34

---

if you add ipython_config.py into ~/.ipython/profile_default dir with lines like below then autoreload functionality will be loaded on ipython startup (tested on 2.0.0):

```
print "--------->>>>>>>> ENABLE AUTORELOAD <<<<<<<<<------------"

c = get_config()
c.InteractiveShellApp.exec_lines = []
```

```
c.InteractiveShellApp.exec_lines.append('%load_ext autoreload')
c.InteractiveShellApp.exec_lines.append('%autoreload 2')
```

answered May 21 '14 at 19:55
lowtech
1,496 1 14 26

You can use:

```
import ipy_autoreload
%autoreload 2
%aimport your_mod
```

edited May 23 '13 at 15:22
Lance Roberts
16.7k 26 94 124

answered May 23 '13 at 15:03
dongweiming
783 1 7 6

There is an extension for that, but I have no usage experience yet:

http://ipython.scipy.org/ipython/ipython/attachment/ticket/154/ipy_autoreload.py

answered Dec 15 '09 at 15:06
miku
119k 29 235 263