



## Learn, Share, Build

Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers.

Join the world's largest developer community.

[Sign Up](#)

## How can I use both Anaconda versions (2.7 & 3.5)?



I was using the Anaconda 3.5 [distro](#) in a Windows 10 machine. Due to dependencies in libraries that I want to work with, I had to have the 2.7 version installed as well.

The good news is that the libraries I needed can now work with the 2.7 version smoothly and Visual Studio 2015 automagically detected my new Python environment.

The problem comes when using the command line. Upon issuing the command

```
conda info --envs
```

I get

```
root * C:\Users\norah\Anaconda2
```

i.e. a single environment (to my understanding and search so far, according to [this](#) I should see two envs listed). This means I can't use `conda` to acquire new packages for my `Python3.5` installation, at least not at the command line as I used to since `conda` only refers to `Python2.7`. The same goes for the GUI version, Anaconda navigator (I'm not very fond of the GUI version but I tried it out).

There's also no way of launching `python3` from the command line since

```
$python
```

always fires up `python2.7` and issuing `$python3` or `$python3.5` in the command line doesn't seem to work (nor would adding the path of `python3` to the system since the actual executable has the same name as `python2` i.e. `python.exe`)

Is my system taken over by Python2.7? **Is anyone here using them both smoothly and if so could you please elaborate on how to do that?** Was it a "no no" move to install both versions of Anaconda?

[python](#) [python-2.7](#) [python-3.x](#) [anaconda](#)

edited Jan 27 at 10:32

asked Jan 25 at 22:17



[Lorah Atkins](#)  
1,518 6 32

In Anaconda with 3.5 duplicate `python.exe` and change name into `pyhton3.exe` and the same with `conda` - and then you can use both versions. – [furas](#) Jan 25 at 22:23

BTW: on Linux I have files `python`, `python3`, `python3.4`, `python3.5`, `python3.6` and `python2`, `python2.7`, `python2.6` and I can run all versions at the same time. You can do something similar on Windows. – [furas](#) Jan 25 at 22:27

@furas Yes, on Ubuntu specifically I do the same thing. Windows seems to be a different case – [Lorah Atkins](#) Jan 25 at 22:30

### 3 Answers

You can make Python 3.5 environment with your Anaconda 2.7:

```
conda create -n py35 python=3.5
```

Now, activate it:

```
activate py35
```

Finally you can install the desired packages:

```
conda install numpy
```

or, the whole anaconda:

```
conda install anaconda
```

The advantage of this approach is that you can also create Python 3.4 or 3.6 environments. Furthermore, you can create environments with different combinations and versions of libraries.

Actually, it makes sense to create a new environment for each larger project.

answered Jan 25 at 22:27



Mike Müller

35.6k 7 44 63

This was my initial approach but issuing the first command you mention lead to "recursion error" :( hence I went with a fresh install – [Lorah Atkins](#) Jan 25 at 22:34

@LorahAtkins This is the best way to manage different versions of python with Anaconda on Windows. If you get an error running the first command, there is an issue with the python installation. Make sure by "clean installation", you uninstalled all other anaconda distributions. Avoid having Anaconda 2.x and Anaconda 3.x installed together; pick one and use envs as shown here. – [pylang](#) Jan 28 at 2:54

@pylang then it's not OK to run both installers so I should delete one distribution and start all over again. I have to finish a project and I'm afraid messing with the installation right now but will try again in a week or so. +1 here and I'll wait to see if there are any more answers before accepting – [Lorah Atkins](#) Jan 28 at 8:32

@LorahAtkins Yes. I recommend, when and if you are able, truly start from scratch and delete all Anaconda distributions. Then install only one. One distribution is sufficient as you can install separate environments. This suggestion is the appropriate method, and I also recommend looking at the SO docs on this topic for assurance [stackoverflow.com/documentation/python/868/virtual-environments/....](https://stackoverflow.com/documentation/python/868/virtual-environments/....) Consider using a Python 3 distribution with Py 2 env since support will end eventually [pythonclock.org](https://pythonclock.org). – [pylang](#) Jan 28 at 18:01



I also despise the virtual environment switch that Anaconda tries to force on us. I prefer to have both executables always instantly available from the command line. I'm pretty sure I had this working on a Windows machine once:

1. Install Anaconda2 and Anaconda3 to the C:\ drive as "C:\Anaconda2\" and "C:\Anaconda3\" respectively.
2. Edit your "Path" environment variable (Control Panel -> System and Security -> System -> Advanced system settings -> Environment Variables) and make sure that "C:\Anaconda2\Scripts;C:\Anaconda2\Library\bin" is in front of "C:\Anaconda3\Scripts;C:\Anaconda3\Library\bin".
3. Copy and rename the file "C:\Anaconda3\python.exe" to "C:\Anaconda3\python3.exe".
4. Copy and rename the file "C:\Anaconda3\Scripts\conda.exe" to "C:\Anaconda3\Scripts\conda3.exe"
5. Copy and rename any other scripts you might use in "C:\Anaconda3\Scripts\", such as "pip.exe" to "pip3.exe", etc.

Now, when you type "python" or "conda" at the command line you will get the python2 version, and when you type "python3" or "conda3", etc. at the command line you will get the python3 version.

answered Jan 28 at 13:17



u55

401 3 4

this looks like something the installers could have done for us right? – [Lorah Atkins](#) Jan 28 at 13:23

@LorahAtkins Yes, the Anaconda packagers could have included all the python versions as individually-named executables (or soft links to them), such as python, python2, python2.7, python3, python3.5, etc., as is commonly done by the packagers for most Linux distributions, and possibly even let the user decide which

version "python" points to, but it seems the Anaconda devs are too infatuated by Virtualenv. – [u55](#) Jan 28 at 13:47

@u55 I am curious why are you opposed to Anaconda's virtual env style of handling python versions? – [pylang](#) Jan 28 at 18:08

@pylang Virtualenv is great for testing code against every python version under the sun, but I don't want to have to use it for daily work because `activate py35 && python` and `deactivate` is too much typing for me. --Not joking. I much prefer to be able to use `python` or `python3` whenever needed. Also, having both executables simultaneously makes it much easier to set up build commands for text editors/IDE's, and makes it much easier to call one version of python from a different version of python using "subprocess". – [u55](#) Jan 28 at 23:16

I think I can commiserate with your second argument. Thanks. – [pylang](#) Jan 28 at 23:56

The best way to use both Python 2.7x and Python 3.5x together is Jupyter Notebook.

<http://jupyter.org/>

You will be able to work on your browser with IPython Notebook style interface which is great for working with scripting languages.

I found some of these videos on YouTube very informative:

1) <https://www.youtube.com/watch?v=HW29067qVWk>

2) <https://www.youtube.com/watch?v=e9cSF3eVQv0>

Besides Python 2.7 and 3.5, there are a bunch of other languages that you will be able run with your Jupyter Notebook. The various Kernels are available in this link below:

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>

After installation, as you start your Jupyter Notebook, your browser will open up a new page showing your Home Directory. Among the 4 tabs (Files|Running|Clusters|Conda):

- 1) The Files tab shows all the files in your Home Directory.
- 2) Running tab shows all the Notebooks that are running.
- 3) Clusters tab is provided by IPython parallel.
- 4) Conda tab is where you need to add the Python version 3.5 (and other languages if needed) to your Jupyter Notebook (If Python 2.7 is default).

If you are interested to try C++ with your Jupyter Notebook, there are a couple of Kernels available.

1) JupyRoot - <https://github.com/root-mirror/root/tree/master/bindings/pyroot/JupyROOT>

2) Cling - <https://github.com/root-mirror/cling>

edited Jan 31 at 17:47

answered Jan 31 at 17:16



[prashanth manohar](#)

129 1 1 15