

Managing Python

Contents

- Managing Python
 - Check Python versions
 - Install a different version of Python
 - Use a different version of Python
 - Create Python 2 or 3 environments
 - Create a Python 3.5 environment
 - Create a Python 2.6 environment
 - Update or Upgrade Python

Conda treats Python the same as any other package, so it's very easy to manage and update multiple installations.

Check Python versions

Check to see which versions of Python are available to install:

```
conda search python
```

Install a different version of Python

So now let's say you need Python 3 to learn programming, but you don't want to wipe out your Python 2.7 environment by updating Python. You can create and activate a new environment named snakes, and install the latest version of Python 3 as follows:

```
conda create --name snakes python=3
```

Linux, OS X: `source activate snakes`

Windows: `activate snakes`

TIP: It would be wise to name this environment a descriptive name like "python3" but that is not as much fun.

To verify that the snakes environment has now been added, type the command:

```
conda info --envs
```

Conda displays the list of all environments, with the current environment highlighted by a '*'.
Verify that the snakes environment uses Python version 3:

```
python --version
```

Use a different version of Python

To switch to the new environment with a different version of Python, you simply need to activate it. Let's switch back to 2.7:

Linux, OS X: `source activate snowflakes`

Windows: `activate snowflakes`

Verify that the snowflakes environment uses Python version 2:

```
python --version
```

After you are finished working in the snowflakes environment, to close it you can either deactivate it, or activate a new environment.

Create Python 2 or 3 environments

Anaconda supports Python 2.6, 2.7, 3.3, 3.4, and 3.5. The default is Python 2.7 or 3.5, depending on which installer you used. If the installer you used is Anaconda or Miniconda, the default is 2.7. If the installer you used is Anaconda3 or Miniconda3, the default is 3.5.

Create a Python 3.5 environment

To create a new environment with a different version of Python, use the `conda create` command. In this example, we'll make the new environment for Python 3.5:

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

Here, the 'py35' is the name of the environment you want to create, and 'anaconda' is the meta-package that includes all of the actual Python packages comprising the Anaconda distribution. When creating a new environment and installing Anaconda, you can specify the exact package and Python versions, for example, `numpy=1.7` or `python=3.5`.

Create a Python 2.6 environment

In this example, we'll make a new environment for Python 2.6:

```
$ conda create -n py26 python=2.6 anaconda
```

Update or Upgrade Python

If you are in an environment with Python version 3.4.2, this command will update Python to 3.4.3, which is the latest version in the 3.4 branch:

```
$ conda update python
```

And this command will upgrade Python to another branch such

as 3.5 by installing that version of Python:

```
$ conda install python=3.5
```