conda vs. pip vs. virtualenv

If you've used pip and virtualenv in the past, you can use conda to perform all of the same operations. Pip is a package manager, and Virtualenv is an environment manager. Conda is both.

Task	Conda package and environment manager command	Pip package manager command	Virtualenv environment manager command
Install a package	`conda install $PACKAGE_NAME`	`pip install $PACKAGE_NAME`	X
Update a package	`conda update --name $ENVIRONMENT_NAME $PACKAGE_NAME`	`pip install --upgrade $PACKAGE_NAME`	X
Update package manager	`conda update conda`	Linux/OSX: `pip install -U pip` Win: `python -m pip install -U pip`	X
Uninstall a package	`conda remove --name $ENVIRONMENT_NAME $PACKAGE_NAME`	`pip uninstall $PACKAGE_NAME`	X
Create an environment	`conda create --name $ENVIRONMENT_NAME python`	X	`cd $ENV_BASE_DIR; virtualenv $ENVIRONMENT_NAME`
Activate an environment	`source activate $ENVIRONMENT_NAME`	X	`source $ENV_BASE_DIR/$ENVIRONMENT_NAME/bin/activate`
Deactivate an environment	`source deactivate`	X	`deactivate`
Search available packages	`conda search $SEARCH_TERM`	`pip search $SEARCH_TERM`	X
Install package from specific source	`conda install --channel $URL $PACKAGE_NAME`	`pip install --index-url $URL $PACKAGE_NAME`	X
List installed packages	`conda list --name $ENVIRONMENT_NAME`	`pip list`	X
Create requirements file	`conda list --export`	`pip freeze`	X
List all environments	`conda info --envs`	X	Install virtualenv wrapper, then `lsvirtualenv`
Install other package manager	`conda install pip`	`pip install conda`	X
Install Python	`conda install python=x.x`	X	X
Update Python	`conda update python` *	X	X

- `conda update python` updates to the most recent in the series, so Python 2 to latest 2.x, Python 3 to latest 3.x, and so on.
