Package help

Most software has a man(ual) page associated with it. Type `man <package name>` at the shell prompt.

Environment Modules

Environment Modules is a software environment management system which allows multiple versions of packages to be available to users.

Common commands

- **module avail** - to list available modules. Modules with (default) will be loaded if a version is not specified.
- **module load <module name>** - to load a module.
- **module unload <module name>** - to unload a module.
- **module list** - to list loaded modules.

Examples - Using Python and Pip with Environment Modules

Python

Several versions of Python are available using Environment Modules, in addition to the default package versions. You can determine what version is available by appending the --version switch to your python command.

**NOTE:** By default the command "python" runs the Python 2.x interpreter and the command "python3" runs the Python 3.x interpreter.

Python example:

- List current modules.

  ```
  ~ $ module list
  No Modulefiles Currently Loaded.
  ```

- Show current Python and Python3 versions. Since no Python modules are loaded, they are the OS provided versions.

  ```
  ~ $ python --version
  Python 2.7.5
  ~ $
  ~ $ python3 --version
  Python 3.4.5
  ```

- List available Python modules. Default module is indicated with (default).
~ $ module avail python
------------------------- /opt/software/modules/ --------------
-------------------
python/2.7.9 python/3.4.2 python/3.5.1(default)

- Load default Python module.

~ $ module load python

- Show Python and Python3 versions. Since only the Python 3.5.1 module was
  loaded, it was the only version that changed.

~ $ python --version
Python 2.7.5
~ $
~ $ python3 --version
Python 3.5.1

- Unload Python module.

~ $ module unload python

**Pip and Virtualenv**

Pip can also be used with python to install python packages. Virtualenv is available

As with Python, pip has different commands for different versions:

- **pip, pip2, and pip2.7** - to install OS maintained Python 2.7 pip packages, or
  a Python 2.7.x module if loaded.
- **pip3** - to install OS maintained Python 3.4 pip packages.
- **pip3.4** - to install OS maintained Python 3.4 pip packages, or a Python 3.4
  version python module is loaded.
- **pip3.5** - to install Python 3.5 pip packages, if a Python 3.5.x module has
  been loaded.

**Pip examples:**

- List current modules.

~ $ module list
No Modulefiles Currently Loaded.
Code01 (replacing Code and Olin)

- Display pip3 and pip3.4 versions. With no Python modules loaded, they use the OS provided version.

  ~ $ pip3 --version
  pip 9.0.1 from /usr/lib/python3.4/site-packages (python 3.4)
  ~ $
  ~ $ pip3.4 --version
  pip 9.0.1 from /usr/lib/python3.4/site-packages (python 3.4)

- Load Python 3.4.2 specific module.

  ~ $ module load python/3.4.2

- Display pip3 and pip3.4 versions. Now they show the version from the loaded Python 3.4.2 module.

  ~ $ pip3 --version
  pip 9.0.1 from /opt/software/python/python-3.4.2/lib/python3.4/site-packages (python 3.4)
  ~ $
  ~ $ pip3.4 --version
  pip 9.0.1 from /opt/software/python/python-3.4.2/lib/python3.4/site-packages (python 3.4)

- Display the pip3.5 version. Since a Python 3.5 module is not loaded, the command fails.

  ~ $ pip3.5 --version
  -bash: pip3.5: command not found

- Unload Python 3.4.2

  ~ $ module unload python/3.4.2

- Load the default python module, which currently is Python 3.5.1.

  ~ $ module load python

- Display the pip3 and pip3.5 versions. Now they show the version from the loaded 3.5.1 Python module.

  ~ $ pip3 --version
Install a pip module into your userspace. Be sure to specify the version of Pip you want to use.

~ $ pip3.5 install --user <pip_module>
Collecting <pip_module>
  Downloading <pip_module>.whl (43.1MB)
    100% |~~~~~~~~~~~~~~~~~~~~~~~~~~~~~| 43.1MB 12kB/s

For more information, please see the official web page for Environment Modules or Wikipedia article.
Unique solution ID: #1378
Author: James Cicak
Last update: 2017-01-24 11:31