Python 3.3 venv Module

Python 3.3 Virtual Environments

Monday, February 25, 2013

PYPTUG

Calloway School of Business

Wake Forest University

Chris Calloway

University of North Carolina

Department of Marine Sciences

Python 3.3 venv Module

venv

is a new standard

library module included

with Python 3.3

What happens when you import a module?

Python 3.3 venv Module

```
>>> import sys
>>> from pprint import pprint
>>> pprint(sys.path)
['',
 '/opt/python330/lib/python33.zip',
 '/opt/python330/lib/python3.3',
 '/opt/python330/lib/python3.3/plat-darwin',
 '/opt/python330/lib/python3.3/lib-dynload',
 '/opt/python330/lib/python3.3/site-packages']
>>>
```

Python 3.3 venv Module

How do modules get into site-packages?

Python 3.3 venv Module

- You put a module or package there
- You ran a package's setup.py
- You used a Python package manager:
    - easy_install
    - pip

Installing in site-packages has some problems

Python 3.3 venv Module

- Only pip has an uninstall option
- Packages may be interdependent
- Third party packages vary in quality
- Your packages may be under development

The answer:

Python "Virtual Environments"

Python 3.3 venv Module

- Throw-away Python instances
- Based on a previously installed Python
- Includes tool to "activate" the environment
- Install packages into the virtual environment

Python 3.3 venv Module

- You may have many virtual environments
- Allows you to test with different versions
- Formerly accomplished with virturalenv
- Python 3.3 ships with the venv module
- Also includes the pyvenv script

Python 3.3 venv Module

```
> /opt/python330/bin/pyvenv myenv
> cd myenv
> ls
bin   include   lib   pyvenv.cfg
> ls bin
activate   pydoc   python   python3   python3.3
> . bin/activate
(myenv) >
```

Python 3.3 venv Module

activate changes the operating environment to search the virtual environment first

Python 3.3 venv Module

```
(myenv) > which python
/Users/cbc/Desktop/myenv/bin/python
(myenv) > echo $PATH
/Users/cbc/Desktop/myenv/bin:.../usr/bin:/bin:,,,
(myenv) > python
Python 3.3.0 (default, Oct 19 2012, 15:13:38)
>>>
```

Python 3.3 venv Module

```
>>> import sys
>>> from pprint import pprint
>>> pprint(sys.path)
['',
 '/opt/python330/lib/python33.zip',
 '/opt/python330/lib/python3.3',
 '/opt/python330/lib/python3.3/plat-darwin',
 '/opt/python330/lib/python3.3/lib-dynload',
 '/Users/cbc/Desktop/myenv/lib/python3.3/site-packages']
>>>
```

Python 3.3 venv Module

```
>>> ^D
(myenv) > deactivate
> cd ..
> rm -rf myenv
>
```

Python 3.3 venv Module

```
> /opt/python330/bin/python3
Python 3.3.0 (default, Oct 19 2012, 15:13:38)
>>> import venv
>>> venv.create("myenv")
>>> ^D
> cd myenv
> . bin/activate
(myenv) > python
Python 3.3.0 (default, Oct 19 2012, 15:13:38)
>>>
```

Python 3.3 venv Module

```
>>> ^D
(myenv) > ls bin
activate   pydoc   python   python3
(myenv) >
```

Wait a minute...

virtualenv also

installed pip

Python 3.3 venv Module

```
(myenv) > python
Python 3.3.0 (default, Oct 19 2012, 15:13:38)
>>> import venv
>>> help(venv)
```

Python 3.3 venv Module

```
Help on package venv:

NAME
    venv - Virtual environment (venv) package for
    Python. Based on PEP 405.
...
    class EnvBuilder(builtins.object)
     |   This class exists to allow virtual
     |   environment creation to be
     |   customised.
```

Python 3.3 venv Module

```
:q
>>> help(venv.EnvBuilder)
```

Python 3.3 venv Module

```
Help on class EnvBuilder in module venv:

class EnvBuilder(builtins.object)
...
 |   create(self, env_dir)
 |       Create a virtual environment in a
 |       directory.
 |
 |       :param env_dir: The target directory to
 }       create an environment in.
```

Python 3.3 venv Module

```
def create(self, env_dir):
    """
    Create a virtualized Python environment in a
    directory.
    env_dir is the target directory to create an
    environment in.
    """
    env_dir = os.path.abspath(env_dir)
    context = self.create_directories(env_dir)
    self.create_configuration(context)
    self.setup_python(context)
    self.setup_scripts(context)
    self.post_setup(context)
```

Python 3.3 venv Module

```
>>> help(venv.EnvBuilder.post_setup)
Help on function post_setup in module venv:

post_setup(self, context)
    Hook for post-setup modification of the
    venv. Subclasses may install additional
    packages or scripts here, add activation
    shell scripts, etc.

    :param context: The information for the
                    environment creation
                    request being processed.
```

Python 3.3 venv Module

```
def post_setup(self, context):
    """
    Hook for post-setup modification of the
    venv. Subclasses may install additional
    packages or scripts here, add activation
    shell scripts, etc.

    :param context: The information for the
                    environment creation
                    request being processed.
    """
    pass
```

Python 3.3 venv Module

- http://docs.python.org/3/library/venv.html
- An example of extending EnvBuilder
- class DistributeEnvBuilder(venv.EnvBuilder)
- Overrides post_setup to install:
  - distribute
  - pip

Python 3.3 venv Module

- https://gist.github.com/vsajip/4673395
- pyvenvex.py
- Install in base Python bin directory

```
> /opt/python330/bin/python3 /opt/python330/bin/pyvenvex.py myenv
```

- Or install in base Python site-packages

```
> /opt/python330/bin/python3 -m pyvenvex myenv
```

Python 3.3 venv Module

```
> /opt/python330/bin/python3 \
  /opt/python330/bin/pyvenvex.py myenv
Installing distribute .........................
...............................................
........................................done.
Installing pip ................done.
> . myenv/bin/activate
(myenv) > pip freeze
distribute==0.6.35
(myenv) >
```

Python 3.3 venv Module

```
(myenv) > pip install haversine
Downloading/unpacking haversine
  Downloading haversine-0.1.tar.gz
  Running setup.py egg_info for package haversine
Installing collected packages: haversine
  Running setup.py install for haversine
Successfully installed haversine
Cleaning up...
(myenv) > python
Python 3.3.0 (default, Oct 19 2012, 15:13:38)
>>> from haversine import haversine
>>> help(haversine)
```

I have a question for you:

Should pyenvex.py be packaged and placed on Pypi?

Questions?

cbc@chriscalloway.org

http://drunkenpython.org/pyvenv.pdf