Distributing Python Packages Using Setuptools

Baiju Muthukadan

ZeOmega, Bangalore

Feb 27, 2014

About Me

- Founded the SMC project in 2001 while studying at REC Calicut
- Employed by FSF India in 2002-2003
- Contributor to Zope project (2006-2010)
- Book Author: A Comprehensive Guide to Zope Component Architecture
- During PyCon India 2013, received the first Kenneth Gonsalves Award

Agenda

- Installing Python
- Setting up development environment
- Distribution using Setuptools
- MANIFEST.in
- Console Scripts

Installing Python

- Download
- Extract
- ./configure --prefix=/path/to/install
- make
- make install

Setting up development environment

- http://www.pip-installer.org/en/latest/ installing.html
- pip install virtualenv
- virtualenv venv1
- source venv1/bin/activate

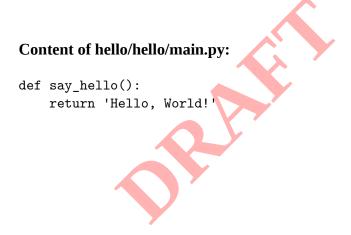
Distribution using Setuptools



Distribution using Setuptools - Continued

```
from setuptools import setup, find packages
setup(name='hello',
      version='0.1.0',
      description='Hello World!',
      author='Baiju Muthukadan',
      packages=find_packages(),
      include package data=True,
      zip_safe=False,
      install_requires=[
          'setuptools',
```

Distribution using Setuptools - Continued



Distribution using Setuptools - Continued

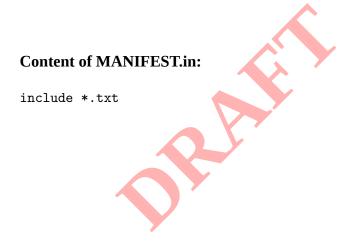
Creating source distributions:

cd hello
python setup.py sdist

Installing distributions:

pip install hello-0.1.0.tar.gz

MANIFEST.in



Console Scripts

from setuptools import setup, find_packages setup(name='hello', version='0.1.0', description='Hello World!', author='Baiju Muthukadan', packages=find packages(), include package data=True, entry points={ 'console scripts': ['hello=hello.main:run',], zip_safe=False, install requires=['setuptools',

Console Scripts - Continued

```
Content of hello/hello/main.py:
def say_hello():
    return 'Hello, World!'
def run():
    hello = say_hello()
    print(hello)
```

Thanks!

http://muthukadan.net

Command line parsing

Content of hello/hello/main.py:

```
import argparse
def say hello(name):
    return 'Hello, %s!' % name
def run():
    parser = argparse.ArgumentParser()
    parser.add_argument("name", help="Your name")
    args = parser.parse_args()
    hello = say hello(args.name)
    print(hello)
```

Command line parsing - Continued

Content of hello/hello/main.py:

```
import argparse
def say_hello(name, count):
    hello list = []
    for i in range(count):
        hello list.append('Hello, %s!' % name)
    return hello list
def run():
    parser = argparse.ArgumentParser()
    parser.add argument("name", help="Your name")
    parser.add argument("--count", type=int)
    args = parser.parse_args()
    for hello in say_hello(args.name, args.count):
        print(hello)
```