Building an open source **Python application** the **right way**

Hi.

I'm Kiran.

Hacker

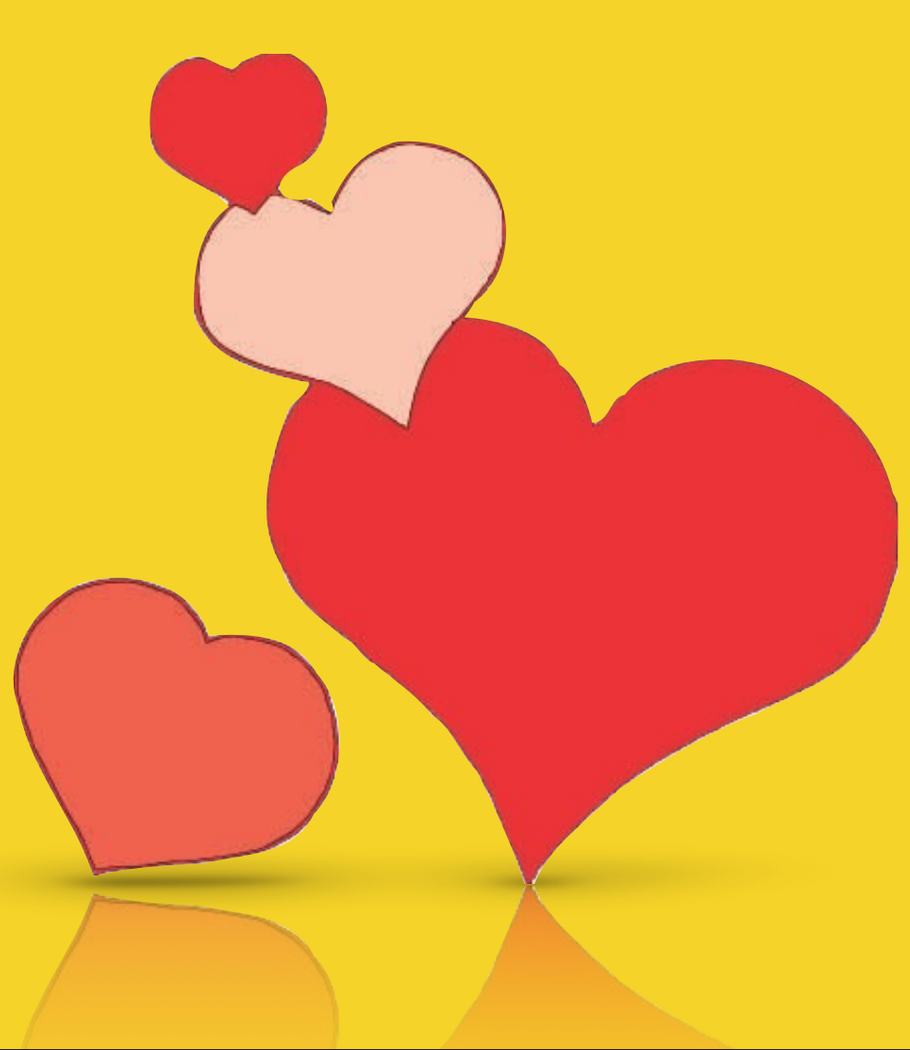
Programmer

Open Source enthusiast

Computer Science fanboy

Currently, having an affair with

Python



Why follow

Guidelines

Conventions

Best Practices

Your code is

Open Source !

What does that mean

You ask ?

Code available to **everyone**

People will **start using it**

People will(might) **try to help**

So, what do I do ?

Make it easier for people to use your application

How ?

Let's start with

basic **project structure**

```
[kiran@Kirans-MacBook-Pro:~/Workspace/bootstrapy on master]
% tree -I 'env|dist|*.pyc' -L 2
zsh: correct 'tree' to 'tee' [nyae]? n
.
|-- AUTHORS.rst
|-- CHANGELOG
|-- CONTRIBUTING.rst
|-- LICENSE
|-- MANIFEST.in
|-- Makefile
|-- README.rst
|-- docs
|   |-- Makefile
|   |-- _build
|   |-- _static
|   |-- _templates
|   |-- conf.py
|   |-- index.rst
|   `-- make.bat
|-- mypackage
|   |-- __init__.py
|   `-- myapp.py
|-- requirements.txt
|-- setup.py
`-- tests
    `-- test_sayhello.py
```

Let’s make it simpler

Application

Tests

Documentation

Dependency Management

Setup

Publish

Task Execution

Extras

Application

Write good code.

http://www.python.org/dev/peps/pep-0008/

Don't bloat your code with

classes

Use them when required

Use

modules

to architect your application

Tests

"Code not tested is broken by design"

– Jacob Kaplan Moss

Lots of ways to

Test

your application

Start with simple

unit tests

py.test

unittest

Nose

Most commonly used

Documentation

For small projects

Code is the documentation

http://www.python.org/dev/peps/pep-0257/

Sadly
This is not **scalable**

Use **Sphinx**

http://sphinx-doc.org/

HTML

EPUB

Theming Support

PDF

Automatic Syntax Highlighting

Man Pages

Plain Text

LaTex

Internationalisation


WHOA.

Dependency Management

Project Dependencies

requirements.txt

then

pip install -r requirements.txt

BRILLIANT!

Setup

Pro Tip:

Use

Virtualenv

to setup your environment

Why ?

- ✓ Test with **different Python versions**
- ✓ Install **dependencies** inside environment
- ✓ No unnecessary **namespace pollution**
- ✓ Different **environments** and **settings** per project

Publish

Every project's

setup.py

contains the necessary information
required for it's installation

Packages and **uploads** your application to

Python Package Index

a.k.a

PyPI

```
setup(
    name='project name',
    version=myapp.__version__,
    description='project description here',
    license=open("LICENSE").read(),
    author='author',
    author_email='email',
    url='project url',
    packages=['myapp'],
    include_package_data=True,
    classifiers=(
        'Development Status :: 5 - Production/Stable',
        'Intended Audience :: Developers',
        'Natural Language :: English',
        'License :: OSI Approved :: MIT License',
        'Programming Language :: Python',
        'Programming Language :: Python :: 2.6',
        'Programming Language :: Python :: 2.7',
        'Programming Language :: Python :: 3.3',
    ),
    test_suite='tests',
)
```

Register at PyPI

```
python setup.py register
```

Package and Upload application

```
python setup.py sdist upload
```

Install application on any platform

```
pip install myapp
```

Task Execution

Use a

Makefile

to automate multiple tasks
and commands

http://mrbook.org/tutorials/make/

Extras

(a.k.a enhancements)

AUTHORS

CHANGELOG

CONTRIBUTING

License

is

very important

for open source projects

http://www.tldrlegal.com/

Recap

Application ✓

Tests ✓

Documentation ✓

Dependency Management ✓

Setup ✓

Publish ✓

Task Execution ✓

Extras ✓

Questions ?

Thank you !

@kirang89

http://kirang.in