The anatomy of a python repository

Bsc. Electrical Engineering

William Jamir Silva

Software Developer

ESSS

github.com/williamjamir

hynek Add Python 2 statement

- .github
- changelog.d
- docs
- src/attr
- tests
- .coveragerc
- .gitignore
- .pre-commit-config.yaml
- .readthedocs.yml
- .travis.yml
- AUTHORS.rst
- CHANGELOG.rst
- LICENSE
- MANIFEST.in
- README.rst
- codecov.yml
- conftest.py
- pyproject.toml
- setup.cfg
- setup.py
- tox.ini

nicoddemus Merge pull request #4764

.github	Add
bench	App
changelog	Mer
doc/en	Clai
extra	Fix f
scripts	[skip
src	Mer
testing	Mer
.coveragerc	cove
.gitattributes	Add
.gitignore	Igno
.pre-commit-config.yaml	pre-
.travis.yml	Add
AUTHORS	Mer
CHANGELOG.rst	Fix t
CITATION	Fix l
CONTRIBUTING.rst	CI: u
HOWTORELEASE.rst	doc:
LICENSE	Upd
README.rst	Upd
appveyor.yml	App
pyproject.toml	Use
setup.cfg	Fix u
setup.py	setu
tox.ini	Mer

njsmith Merge pull request #919 fr

- ci
- docs
- logo
- newsfragments
- notes-to-self
- trio
- .coveragerc
- .gitignore
- .readthedocs.yml
- .style.yapf
- .travis.yml
- CODE_OF_CONDUCT.md
- CONTRIBUTING.md
- LICENSE
- LICENSE.APACHE2
- LICENSE.MIT
- MANIFEST.in
- README.rst
- azure-pipelines.yml
- check.sh
- mypy.ini
- pyproject.toml
- setup.py
- test-requirements.in
- test-requirements.txt

WHY ALL THESE FILES
TO JUST CREATE A PACKAGE

Building a package

🐍 main.py

```python
def main():
    return "Hello"
```

🐍 setup.py

```python
from setuptools import setup

setup(
    name = "python-floripa",
    py_modules = ["python-floripa"]
)
```

```
william  in /opt/python-floripa ((HEAD))
(python-floripa) >> python setup.py bdist_wheel
running bdist_wheel
running build
running build_py
...
running install_egg_info
running egg_info
creating python_floripa.egg-info
...
Copying python_floripa.egg-info to build/bdist.linux-x86_64/wheel/python_floripa-0.0.0-
py3.7.egg-info
running install_scripts
creating build/bdist.linux-x86_64/wheel/python_floripa-0.0.0.dist-info/WHEEL
creating 'dist/python_floripa-0.0.0-py3-none-any.whl' and adding 'build/bdist.linux-
x86_64/wheel' to it
adding 'python_floripa-0.0.0.dist-info/METADATA'
adding 'python_floripa-0.0.0.dist-info/WHEEL'
adding 'python_floripa-0.0.0.dist-info/top_level.txt'
adding 'python_floripa-0.0.0.dist-info/RECORD'
removing build/bdist.linux-x86_64/wheel
```

```
(python-floripa) >> tree
.
├── build
│   └── bdist.linux-x86_64
├── dist
│   └── python_floripa-0.0.0-py3-none-any.whl
├── main.py
├── python_floripa.egg-info
│   ├── dependency_links.txt
│   ├── PKG-INFO
│   ├── SOURCES.txt
│   └── top_level.txt
└── setup.py

4 directories, 7 files
```

```
(python-floripa) >> pip install dist/python_floripa-0.0.0-py3-none-any.whl
Processing ./dist/python_floripa-0.0.0-py3-none-any.whl
Installing collected packages: python-floripa
Successfully installed python-floripa-0.0.0
```

williamjamir Update CONTRIBUTING file

..

📁 docs	Update CONTRIBUTING file	
📁 src/{{cookiecutter.package_name}}	Add end of lines on files	
📁 test	Add end of lines on files	
.codecov.yml	Add initial skeleton	
.editorconfig	Add end of lines on files	
.gitattributes	Add initial skeleton	
.gitignore	Update gitignore	
.isort.cfg	Add initial skeleton	
.pre-commit-config.yaml	Update .pre-commit-config.yaml	
.readthedocs.yml	Add initial skeleton	
.travis.yml	Add initial skeleton	
CHANGELOG.RST	Update the Get Started section on CONTRIBUTIN	
CONTRIBUTING.rst	Update CONTRIBUTING file	
LICENSE	Cokiecutter folder structure	
README.rst	Update CONTRIBUTING file	
appveyor.yml	Add end of lines on files	
setup.cfg	Add end of lines on files	
setup.py	Update CONTRIBUTING file	
tox.ini	Use extras instead of deps on tox.ini	

IF IT WAS THAT EASY

WHY SO MANY FILES??

imgflip.com
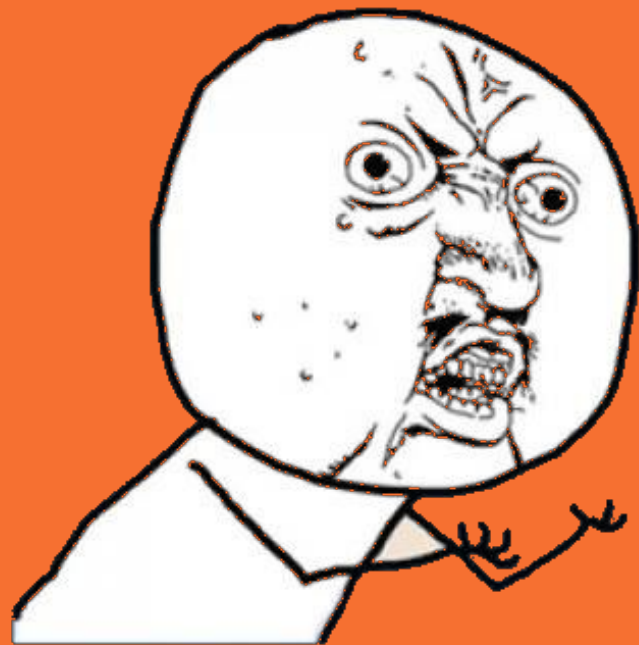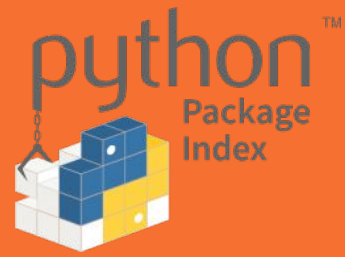
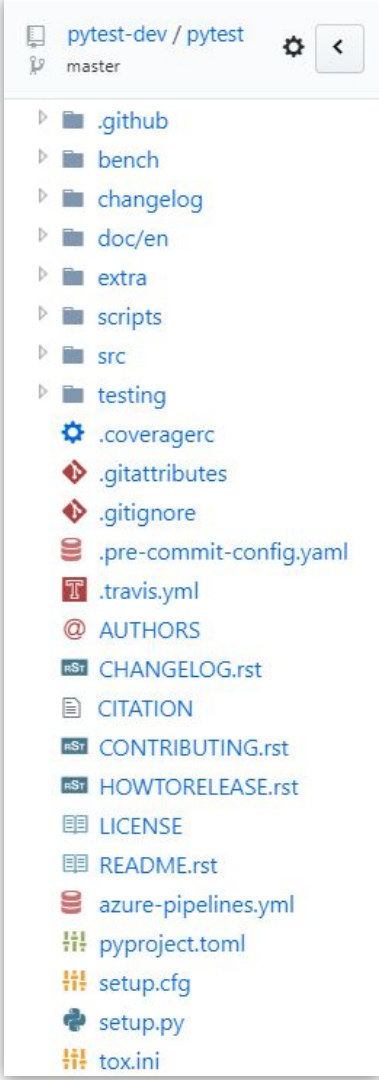# The anatomy of a python repository

~~For humans~~

~~From zero to hero~~

<u>How the cool kids are doing it</u>

Package

"Packaging is all about target environment and deployment experience"

https://packaging.python.org/

This presentation focus on:

Packaging Python libraries and tools

pytest-dev / pytest

master

- .github
- bench
- changelog
- doc/en
- extra
- scripts
- src
- testing
- .coveragerc
- .gitattributes
- .gitignore
- .pre-commit-config.yaml
- .travis.yml
- AUTHORS
- CHANGELOG.rst
- CITATION
- CONTRIBUTING.rst
- HOWTORELEASE.rst
- LICENSE
- README.rst
- azure-pipelines.yml
- pyproject.toml
- setup.cfg
- setup.py
- tox.ini

Categories

Repository

Collaboration

Packaging

Linters

Tests

Formatter

Docs

pytest

Repository

Collaboration

Packaging

Tests

Linters

Formatter

CI

Docs

---

pytest-dev / pytest
master

- .github
- bench
- changelog
- doc/en
- extra
- scripts
- src
- testing
- .coveragerc
- .gitattributes
- .gitignore
- .pre-commit-config.yaml
- .travis.yml
- @ AUTHORS
- CHANGELOG.rst
- CITATION
- CONTRIBUTING.rst
- HOWTORELEASE.rst
- LICENSE
- README.rst
- azure-pipelines.yml
- pyproject.toml
- setup.cfg
- setup.py
- tox.ini

Repository

- .gitattributes
- .gitignore
- .mailmap

Collaboration

- @ AUTHORS
- CHANGELOG.rst
- CONTRIBUTING.rst
- HOWTORELEASE.rst
- LICENSE
- README.rst
- .editorconfig

python-attrs / attrs

master

- ▷ .github
- ▷ changelog.d
- ▷ docs
- ▷ src/attr
- ▷ tests
- .coveragerc
- .gitignore
- .pre-commit-config.yaml
- .readthedocs.yml
- .travis.yml
- AUTHORS.rst
- CHANGELOG.rst
- LICENSE
- MANIFEST.in
- README.rst
- codecov.yml
- conftest.py
- pyproject.toml
- setup.cfg
- setup.py
- tox.ini

Repository

Collaboration

Packaging

Tests

Linters

Formatter

CI

Docs

Packaging

- pyproject.toml
- setup.cfg
- setup.py
- MANIFEST.in

Package

Configs

setup.py

```python
import setuptools

with open("README.md", "r") as fh:
    long_description = fh.read()

setuptool.setup(
    name="python-floripa",
    version="0.0.1",
    author="",author_email="",
    description="",
    long_description=long_description,
    packages=setuptool.find_packages(),
    install_requires=["attrs >= 18.2.0"],
    python_requires=">=3.6",
)
```

Configs

setup.cfg

```
[metadata]
name = my_package
version = attr: src.VERSION
description = A small example package
long_description = file: README.md
license = BSD 3-Clause License
classifiers =
    Programming Language :: Python :: 3

[options]
packages = find:
install_requires =
  attrs >= "18.2.0"
python_requires = 3.5
```

Configs

manifest.in

```
include LICENSE *.rst *.toml *.yml *.yaml
graft .github

# Stubs
include src/attr/py.typed
recursive-include src *.pyi

# Tests
include tox.ini .coveragerc conftest.py
recursive-include tests *.py

# Just to keep check-manifest happy; on releases those files are gone.
# Last rule wins!
exclude changelog.d/*.rst
include changelog.d/towncrier_template.rst
```

hynek Just put everything into the sdist so everyone is happy

Handling dependencies

tools and libraries

install_requires

VS

applications

requirements.txt

- Automatically analyzed by pip during an install

- "Abstract": not associated with any particular index,

- List of pip install arguments placed into a file

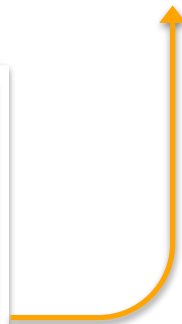- "Concrete": associated with a particular index or directory of packages.

Configs

FLIT   Poetry

pyproject.toml

```
[build-system]
requires = ["setuptools>=40.6.0", "wheel"]
build-backend = "setuptools.build_meta"



[tool.black]
line-length = 79



[tool.towncrier]
```

python-attrs / attrs

master

▷ 📁 .github
▷ 📁 changelog.d
▷ 📁 docs
▷ 📁 src/attr
▷ 📁 tests
⚙️ .coveragerc
◈ .gitignore
🗄️ .pre-commit-config.yaml
🗄️ .readthedocs.yml
🅣 .travis.yml
AUTHORS.rst
CHANGELOG.rst
LICENSE
MANIFEST.in
README.rst
codecov.yml
conftest.py
pyproject.toml
setup.cfg
setup.py
tox.ini

Repository

Collaboration

Packaging

Tests

Linters

Formatter

CI

Docs

Tests

⚙️ .coveragerc

🐍 conftest.py

🎚️ pytest.ini

☂️ codecov.yml

🎚️ tox.ini

Tests

Tests

```
coverage report -m
Name                        Stmts   Miss  Cover   Missing
-----------------------------------------------------------
my_program.py                  20      4    80%   33-35, 39
my_other_module.py             56      6    89%   17-23
-----------------------------------------------------------
TOTAL                          76     10    87%
```
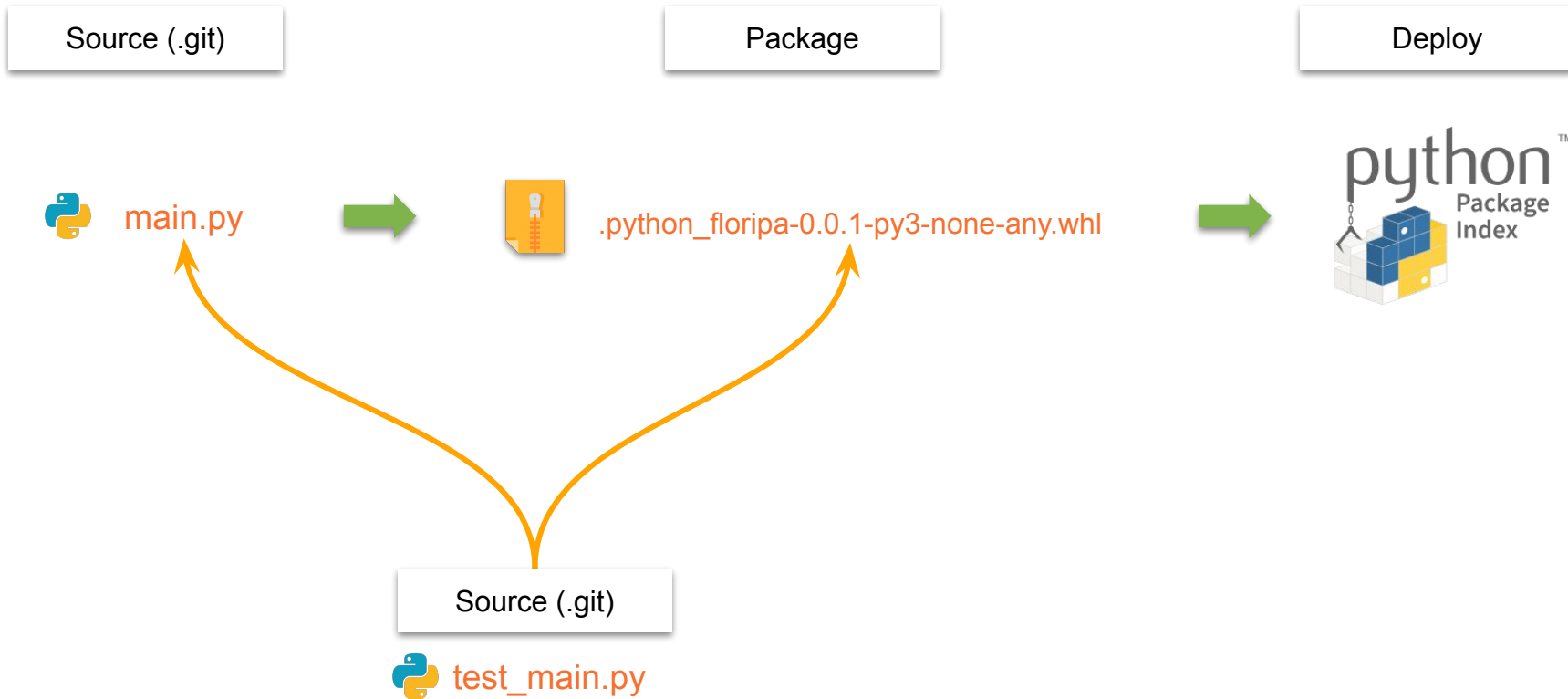
Coverage.py

codecov.yml

.coveragerc

.coveralls.yml

```python
@total_ordering
class Fraction:
    """
    Numerator, denominator similar to standard fraction with support to
    float values in numerator and denominator which will be automatically
    reduced to a simpler fraction. Class has also an special handling for
    rounding and representation issues for numbers.
    """

    def __init__(self, a, b=None):
        if a == float("inf") or a == float("-inf"):
            raise ValueError("Numerator cannot be an infinite value")

        if b == float("inf") or b == float("-inf"):
            raise ValueError("Denominator cannot be an infinite value")
```
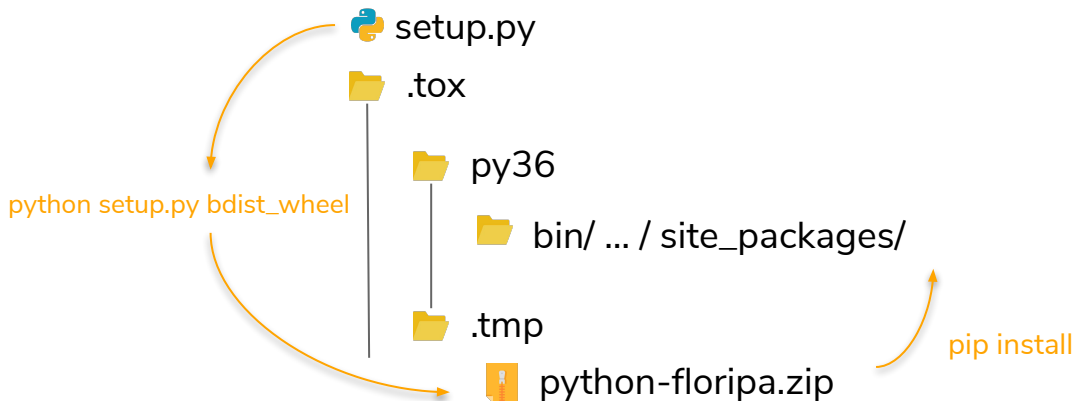
Tests

tox

- Virtualenv management
- Test command line tool

setup.py
.tox
  py36
    bin/ ... / site_packages/
  .tmp
    python-floripa.zip

python setup.py bdist_wheel

pip install

```
[tox]
envlist = py36, py37

[testenv]
ccommands =
    pytest
    codecov
```

tox

```
>> tox
GLOB sdist-make: /opt/python-floripa/setup.py

py36 create: /opt/python-floripa/.tox/py36
py36 inst: /opt/python-floripa/.tox/.tmp/package/1/python-floripa-0.0.1.zip
py36 installed: python-floripa==0.0.1
py36 run-test-pre: PYTHONHASHSEED='1977512429'
py36 run-test: commands[0] | python --version
Python 3.6.7

py37 create: /opt/python-floripa/.tox/py37
py37 inst: /opt/python-floripa/.tox/.tmp/package/1/python-floripa-0.0.1.zip
py37 installed: python-floripa==0.0.1
py37 run-test-pre: PYTHONHASHSEED='1977512429'
py37 run-test: commands[0] | python --version
Python 3.7.1
```

```
>> tree -aL 3
.
├── conftest.py
├── main.py
├── setup.py
├── test_main.py
├── .tox
│   ├── py36
│   │   ├── bin
│   │   ├── include
│   │   ├── lib
│   │   ├── log
│   │   ├── tmp
│   │   └── .tox-config1
│   ├── py37
│   │   ├── bin
│   │   ├── include
│   │   ├── lib
│   │   ├── log
│   │   ├── tmp
│   │   └── .tox-config1
├── tox.ini
```

python-trio / trio

master

▷ 📁 ci
▷ 📁 docs
▷ 📁 logo
▷ 📁 newsfragments
▷ 📁 notes-to-self
▷ 📁 trio
⚙ .coveragerc
◆ .gitignore
🗄 .readthedocs.yml
⚙ .style.yapf
🝙 .travis.yml
🔠 CODE_OF_CONDUCT.md
🔠 CONTRIBUTING.md
🈯 LICENSE
🈯 LICENSE.APACHE2
🈯 LICENSE.MIT
📄 MANIFEST.in
🈯 README.rst
🗄 azure-pipelines.yml
▣ check.sh
🎚 mypy.ini
🎚 pyproject.toml
🐍 setup.py
📄 test-requirements.in
📄 test-requirements.txt

Repository

Collaboration

Packaging

Tests

Linters

Formatter

CI

Docs
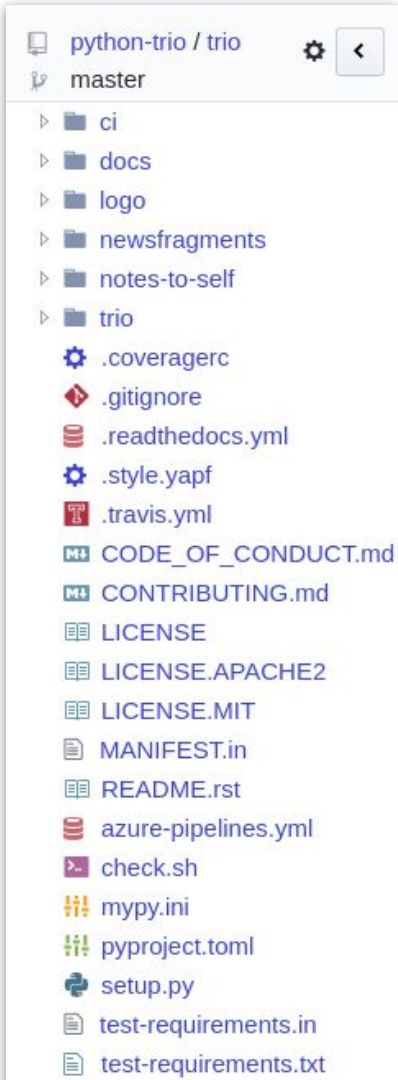
Linters

🎚 mypy.ini

⚙ .flake8

📄 pylintrc

🗄 bandit.json

Linters

Pylint
✦✦✦✦◇ Star your Python code!

🐍 main.py

```python
def test():
    import sys
    type = 1

    variable = type + 1
```

```
(python37) λ pylint main.py
************* Module main
main.py:3:4: W0622: Redefining built-in 'type' (redefined-builtin)
main.py:2:4: W0611: Unused import sys (unused-import)
main.py:5:4: W0612: Unused variable 'variable' (unused-variable)
```

Linters

🐍: my[py]

🐍 main.py

```python
1 def test(parameter: str)-> str:
2     import sys
3     type = 1
4     variable = parameter + 1
5
6     return type
```

➡️

```
(python37) mypy main.py
main.py:4: error: Unsupported operand types for + ("str" and "int")
main.py:6: error: Incompatible return value type (got "int", expected "str")
```

Linters

Pylint   pylintrc

: my[py]   mypy.ini

Flake8   .flake8
         setup.cfg

Pycodestyle
Pyflakes
Mccave
+plugins

Bandit   bandit.json

Linters

pre-commit

```
$ git commit -m "awesome feature"
black......................Passed
Flake8.....................Passed
```

```yaml
repos:
-   repo: https://github.com/ambv/black
    rev: 18.9b0
    hooks:
    -   id: black
        args: [--safe, --quiet]
        language_version: python3
-   repo: https://gitlab.com/pycqa/flake8
    rev: 3.7.0
    hooks:
    -   id: flake8
        language_version: python3
```

python-trio / trio
master

- ▷ 📁 ci
- ▷ 📁 docs
- ▷ 📁 logo
- ▷ 📁 newsfragments
- ▷ 📁 notes-to-self
- ▷ 📁 trio
- ⚙ .coveragerc
- ◈ .gitignore
- .readthedocs.yml
- ⚙ .style.yapf
- .travis.yml
- CODE_OF_CONDUCT.md
- CONTRIBUTING.md
- LICENSE
- LICENSE.APACHE2
- LICENSE.MIT
- MANIFEST.in
- README.rst
- azure-pipelines.yml
- check.sh
- mypy.ini
- pyproject.toml
- setup.py
- test-requirements.in
- test-requirements.txt

Repository

Collaboration

Packaging

Tests

Linters

Formatter

CI

Docs

Code Formatters

⚙ .style.yapf

∥∥ .isort.cfg

Code Formatter

isort

.isort.cfg

Black

pyproject.toml

[tool.black]

YAPF

.style.yapf

```python
from my_lib import Object3

from my_lib import Object2

import sys

from third_party import lib2, lib1

from __future__ import absolute_import

from third_party import lib3

class foo  (       object   ):
  def f    (self    ):
    return       37*-+2
  def g(self, x,y=42):
      return y
def f  (    a  ) :
  return      37+-+a[42-x :  y**3]
```

```python
from __future__ import absolute_import

import sys

from my_lib import Object2, Object3
from third_party import lib1, lib2, lib3


class foo(object):
    def f(self):
        return 37 * -+2

    def g(self, x, y=42):
        return y


def f(a):
    return 37 + -+a[42 - x : y ** 3]
```

attrs

Repository	
Collaboration	
Packaging	
Tests	
Linters	
Formatter	
CI	
Docs	

master

▷ 📁 .github
▷ 📁 changelog.d
▷ 📁 docs
▷ 📁 src/attr
▷ 📁 tests
⚙ .coveragerc
◈ .gitignore
🗄 .pre-commit-config.yaml
🗄 .readthedocs.yml
T .travis.yml
rST AUTHORS.rst
rST CHANGELOG.rst
📖 LICENSE
📄 MANIFEST.in
📖 README.rst
☂ codecov.yml
🐍 conftest.py
▦ pyproject.toml
▦ setup.cfg
🐍 setup.py
▦ tox.ini

CI

▲ 📁 .circleci
   🗄 config.yml
T .travis.yml
🗄 azure-pipelines.yml
◉ appveyor.yml

CI

Repository

Fetch

Continuous Integration Server

Notify
(Success or Failure)

Build   Test   Deploy

Push
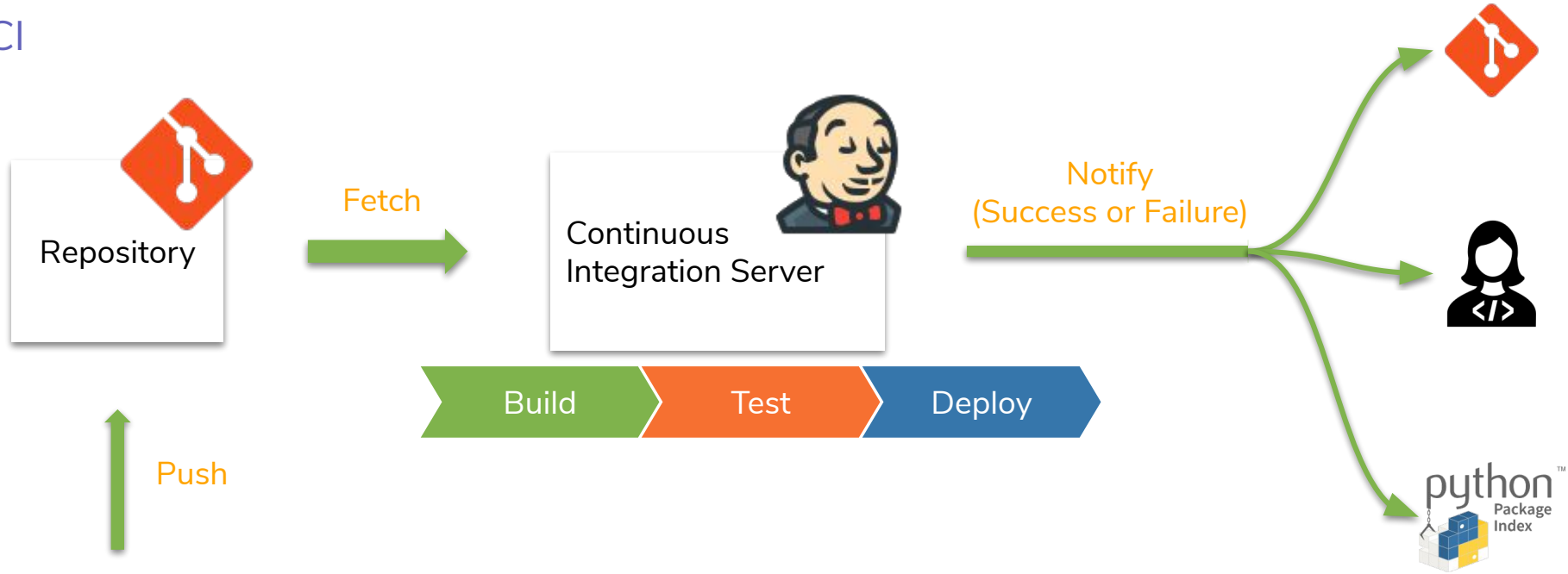
"Continuous Integration (CI) is a development practice that requires to integrate code several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early."

python-trio / trio

master

- ▷ 📁 ci
- ▷ 📁 docs
- ▷ 📁 logo
- ▷ 📁 newsfragments
- ▷ 📁 notes-to-self
- ▷ 📁 trio
- ⚙ .coveragerc
- ◆ .gitignore
- 🗄 .readthedocs.yml
- ⚙ .style.yapf
- 🔧 .travis.yml
- 📝 CODE_OF_CONDUCT.md
- 📝 CONTRIBUTING.md
- 📖 LICENSE
- 📖 LICENSE.APACHE2
- 📖 LICENSE.MIT
- 📄 MANIFEST.in
- 📖 README.rst
- 🗄 azure-pipelines.yml
- ⌨ check.sh
- 🎚 mypy.ini
- 🎚 pyproject.toml
- 🐍 setup.py
- 📄 test-requirements.in
- 📄 test-requirements.txt

Repository

Collaboration

Packaging

Tests

Linters

Formatter

CI

Docs

Docs

- ▷ 📁 docs
- 🗄 .readthedocs.yml
- 📖 README.md
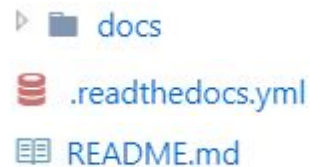
Docs

Read *the* **Docs**

📖 README.rst ➡️ 🔲 index.html

SPHINX
Python Documentation Generator

MkDocs

📖 README.md ➡️ 🔲 index.html

"Markdown syntax is intended for one purpose: to be used as a format for writing for the web."
- John Gruber

Cookiecutter


```
(python37) λ cookiecutter https://github.com/audreyr/cookiecutter-pypackage.git

full_name: William
email: williamjamir@gmail.com
github_username: williamjamir

project_name: python-floripa
version: 0.1.0

project_short_description: Awesome repo
pypi_username: williamjamir

use_pytest: y
use_pypi_deployment_with_travis: y

Select open_source_license:
1 - MIT license
2 - BSD license
3 - ISC license
4 - Apache Software License 2.0
5 - GNU General Public License v3
6 - Not open source
Choose from 1, 2, 3, 4, 5, 6: 1
```

That's all Folks!

Tests


```
(python37) >> tree
.
├── conftest.py
├── main.py
└── test_main.py
```

```
import pytest
@pytest.fixture
def constat():
    return 1
```

```
def test_answer(constat):
    assert constat == 5
```

```
(python37) >> pytest -v test.py
==============================
test session starts
==============================
collected 1 item
test.py::test_answer FAILED [100%]
============FAILURES ============
_____ test_answer _____
    def test_answer(some_constat):
>       assert some_constat == 5
E       assert 1 == 5
E         -1
E         +5

test.py:2: AssertionError
==== 1 failed in 0.04 seconds ====
```

Test Layout

Tests outside the code

```
● ● ●

pythonFloripa/
    app.py
tests/
    foo/
        test_app.py
```

Tests as part of code

```
● ● ●

pythonFloripa/
    _test/
        test_app.py
    app.py
```

pytest --pyargs python_floripa

CI

.circleci
config.yml

circleci

.travis.yml

Travis CI

appveyor.yml

AppVeyor

azure-pipelines.yml

Azure Pipelines

```yaml
language: python

stages:
- name: test
  if: repo = ESSS/hookman AND tag IS NOT present
- name: deploy
  if: repo = ESSS/hookman AND tag IS present

jobs:
  include:
    - python: '3.6'
      env: TOXENV=py36
    - python: '3.7'
      env: TOXENV=py37

    - stage: deploy
      python: '3.6'
      env:
      install: pip install -U setuptools
      script: skip
      deploy:
        provider: pypi
        distributions: sdist bdist_wheel
        user: williamjamir
        password:
          secure:
        on:
          tags: true
```

Build / publish

Explicar Twine ou mostrar -> travis-ci

Testing the build -> tox

Pip and conda-forge

PYRIGHT
pyrightconfig.json

PXRE
.pyre_configuration

: my[py]