

Baby steps towards a better software development

Reinaldo
BaNaNa Talk

Let's check some Python's project on github

- [Scipy](#).
- [Nutils](#).
- [KryPy](#).

The content of this talk is based on these sources:

- *Enterprise Software with Python Architecture and Best Practices*, by Mahmoud Hashemi. Publisher: O'Reilly Media.
- *The Hitchhiker's Guide to Python: Best Practices for Development*, by Kenneth Reitz and Tanya Schlusser. Publisher: O'Reilly.

Outline

- 1 - Creating a project.
- 2 - Structuring a project.
- 3 - Adding quality.

1 - Creating a project.

- Make a directory.
- Create a README file.
- Create a .gitignore file.
- Start git.
- Add files to git.
- Create and activate virtualenv.
- Install dependencies.
- Freeze environment.
- Commit and push.

1 - Creating a project.

- Make a directory. *\$ mkdir myProject/*
- Create a README file. *\$ vim README*
- Create a .gitignore file. *\$ vim .gitignore* or copy it
- Start git. *\$ git init*
- Add files to git. *\$ git add *.py README .gitignore*
- Create and activate virtualenv. (live demo)
- Install dependencies. (live demo)
- Freeze environment. (live demo)
- Commit and push. (*\$ git remote add ... and \$git push ..*)

2 - Structuring a project.

- Project directory.
- Package directory and `__init__.py`.
- Docs and tools.
- Tests.
- `setup.py`
- MANIFEST.in, README, CHANGELOG, LICENSE.

Adding quality.

- Pyflakes (live demo)
- [Pep8](#) (live demo)
- Unit tests (live demo)
- Continuous integration (travis-ci live demo)

Unit test.

“sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.”

([Wikipedia](#)) (live demo)

Continuous integration (travis-ci).

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

([thoughtworks](#)) (live demo with [travis-ci](#))

Example of [.travis.yml](#)

CI



Interesting

- Coverage, coverage all (Python).
- Assertions for [floating point](#) (Python-numpy).
- Google style guide, c++, Java, Python.
- Google unit test, c++.
- [Matlab](#)

The takeaway

- Structure your Python code in packages and submodules.
- Improve the quality of your code using PEP8, pyflakes.
- Make your code maintainable adding unit tests.
- Automate your testing.