2024-03-13_PyLadies-Paris

March 14, 2024

0.1 How do YOU start contributing to Open Source?

0.1.1 Abstract:

In this talk, we'll explore what it needs to start contributing to open source projects as a beginner. We'll discuss the mindset required for successful participation, focusing on qualities like collaboration, curiosity, and a willingness to learn. We'll cover essential pre-requisites such as understanding version control systems, issue tracking, and the basics of code review. Finally, we'll provide practical advice on where to begin your open source journey, including tips for finding beginner-friendly projects and issues and leveraging online resources and communities. If you're looking to take your first steps into the world of open source contribution, don't miss out on this.

0.1.2 About me

Stefanie Sabine Senger

- Wikipedia: Writer, Mentor, WikiWoman, 2011-2014
- Historian: PhD on Solidarity Movements, 2021
- Data Science Bootcamp: LeWagon along with a crazy amount of courses from 2021
- scikit-learn internship, 2023
- muffintech: Data Scientist, 2023
- probabl: Open Source Developer working mainly on scikit-learn, since 2024

contact https://github.com/StefanieSenger

https://www.linkedin.com/in/StefanieSenger

stefanie@probabl.ai

0.1.3 About :probabl.

- company founded on behalf of many of the scikit-learn core contributors in late 2023
- develop, sustain and maintain open source libraries in data science
- provide solutions to problems in the data field while running, inspecting, validating and tracking models
- services in the AI field (training, certification, expertise)

0.1.4 What mindset do you come with?

You come as a learner/explorer.

Be prepared to do a lot of research and invest time. - You try to find out what this issue is about. - You can and should ask questions, but don't expect other contributors to guide you through the process end to end.

Your contribution is a proposal. It will be discussed. - Other than to work it also needs to fit some other design goals, that you might not be aware of when you start. - You might be asked to make changes to your contribution before it's merged.

0.1.5 What mindset do you come with?

Expectation management

- finding a project and an issue to work on: a few hours to a few days
- setting up the project on your computer: a few hours to a few days
- contributing your first simple issue: a few hours to a day
- getting your first issue reviewed and merged: several days to weeks

0.1.6 What to know before you start?

Nothing, really.

0.1.7 What to know before you start?

Things you want to get into

- GitHub account
  - GitHub's tutorial on creating an account
- virtual environments
  - conda
  - pipenv
  - pyenv
- git
  - great game to learn git: OH MY GIT!
- testing
  - pytest
- packaging
  - great LinkedIn Learning course you can access for free: Create an Open Source Project in Python

0.1.8 What to know before you start?

Prepare your local repo

1. create a fork of your open source project on your GitHub
2. clone the fork to have it locally

source: https://happygitwithr.com/fork-and-clone.html

Sketch of general workflow

3. make a new branch for your contribution
4. edit on your branch
5. commit to your branch
6. push your commits to your remote branch in origin

source: https://asoplata.com/publications/talks/20171117-git-intro/slides#/fork-and-pr-tutorial-2

Sketch of general workflow

7. make a Pull Request (PR)
    - push more changes to your branch during review process
    - update your branch with the current status of the dev repo by pulling their "main" into your branch (`git pull`)
    - CI will run some more checks
8. be merged

source: https://asoplata.com/publications/talks/20171117-git-intro/slides#/fork-and-pull-request-model-again

Sketch of general workflow

- before creating a new local branch, you update your local main with the upstream main: `git fetch upstream` `git rebase upstream/main`

source: https://asoplata.com/publications/talks/20171117-git-intro/slides#/fork-and-pull-request-model-again

0.1.9 What to know before you start?

Setting up the project

- install the package in editable/development mode `pip install -e .`
- build and re-build project (if it uses compiled languages) `setup.py build`
- build documentation `sphinx-build -b html docs doc/_build/html`
- these tasks are often automated in a Makefile or Meson script
- refer to the contributing guide of your project for more details

0.1.10 How to actually start?

Find a project

- a project you have worked with before
    - libraries for Python, written mostly in Python: Django, matplotlib, pandas, FastAPI, TensorFlow, Keras, PyTorch, scikit-learn, Flask, AirFlow, scipy, numpy …
- look out for projects that have a published contribution guide, such as:
    - Pandas Contributing Guide
    - Jupyter Contributing Guide

    - scikit-learn's Contributing Guide

- look out for sprints

0.1.11 How to actually start?

Open an issue

- you make an issue
    - (for beginners:) you find the documentation confusing and want to improve it
    - (for pros:) you find an inconsistency when using a tool this library provides
- you follow the discussion
    - wait for a maintainer's assessment (triage)
    - you might work on the issue yourself or closely watch the person that does

0.1.12 How to actually start?

Find an issue

- find issues by label: `good first issue`, `easy`, `meta-issue`, `sprint`
- sort by "recently updated", look for many replies to find untagged meta-issues
- work on a stalled issue (but heat up the discussion before)
- concrete suggestions:
    - scikit-learn example links
    - scipy examples in docstrings

0.1.13 How to actually start?

Contribute something nobody has asked you for

- contribute documentation to docstrings
    - many projects use the same protocol: numpydoc extension for Sphinx: numpydoc Style Guide
    - pandas API
    - Flask API
    - jupyter Widgets list
    - scikit-learn API
- write your own jupyter widget
    - creatively invent something useful
    - get something to visually show off

0.1.14 How to actually start?

Contribute something nobody has asked you for

- improve examples and user guide
    - scikit-learn examples
    - scikit-learn user guide
    - pandas user guide

- contribute tests
- solve #TODOs
  - serve as reminders among contributors to revisit this part of the code

0.1.15 What to do when you are stuck?

- reach out to the community
  - slack, discord, meetings
- describe your problem by showing your code and error messages
- if in doubt: make a draft PR

0.1.16 Thank you!

Questions?

You can also ask me later: https://github.com/StefanieSenger

https://www.linkedin.com/in/StefanieSenger

stefanie@probabl.ai