

# WriteSpeakCode/ PyLadies joint meetup 2015-10-22: Tales of Open Source: rough notes

PyLadies: international mentorship program for female python coders

- meetup.com, NYC PyLadies
- Lisa moderating, Panelists: Maia McCormick, Anna Herlihy, Julian Berman, Ben Darnell, David Turner
- Intros:
  - Maia: worked on Outreachy (formerly OPW) - gives stipends to women and minorities to work on OS code; currently at Spring
  - Anna: works at MongoDB, does a lot of Mongo OS work.
  - Julian: works at Magnetic (ad company); worked on Twisted, started OS project (schema for validating Json projects)
  - Ben: Tornado maintainer, working on OS distributed database on Go.
  - David: ex FSF, OpenPlans, now at Twitter, "making git faster".
- Q: how to find OS projects, how to get started?
  - D: started contributing to Xchat... someone said "wish chat had the following feature"... silence... recently, whatever the company is working on. Advice: find the right project, see if they're interested, then write the feature.
  - B: started on python interpreter, was using game library, needed bindings for library
  - J: looked at OpenHatch OS projects. Found Twisted - told that if want to get code in there, there's a review process. Found feature/bug, wrote patch, waited for response - that got him in... vehicle for other people to read and respond to code.
  - A: first OS commit was to Mongoddb - interned there after college. Couldn't get feature to work on her mac, fixed it 'til it ran, then someone asked "are you going to put in a core request"...was first experience of request politics. Hard to find projects that both need help, and want help. Best to contact first, e.g. "are you interested in a fix for OSX". Most people's experience of OS has been rejection or a negatively tinged experience.
  - M: first pull request got landed... top-down approach, "how do I get work experience on a big codebase - obvious answer is OS... applied to outreachy, who have a list of orgs who want donations"... found Gnome music on the list... iTunes for Gnome... looked at list of beginner-friendly bugs, built that ("approx a million years") on own machine. Gnome are particularly newbie-friendly.
  - Outreach deadline is Nov 2nd.
- Q: how do you find a project that wants your contribution? (or tips for what to avoid)
  - D: avoid people who are loudly mean (e.g. Linux kernalists). Responsiveness beyond everything... e.. friendly community who took a month to fix their instructions... sat on patch for 1-2 months. Good: active community, can see closed pull requests (but linux/git have mailing lists, but that's active)

- B: has a list of newbie-friendly bugs.
- J: gauge on whether want to use that software or not.
- A: bugs are best place to start. Filing a bug report tells you a lot about the maintainers, e.g. on it immediately, starting a conversation about it, you can follow the progress of the bug - see the conversations between the contributors, reminds you that there are humans behind it... "any kind of form of life".
- M: probably would have started on bpython (shinier ipython), because peer was really excited about it... peer recommendation, people excited about a project = project probably doesn't suck.
- Q: suggestions for good places to find lists of welcoming OS projects
  - OpenHatch
  - Hacktoberfest (organised by digital ocean) - everyone submitting 4 projects from the list gets a free t-shirt
  - Look at the projects that OS projects include... those tools are also interesting projects.
  - Go to your bosses and ask if you can release the company software as OS.
- Q: about your projects, features, bugs - something you'd like to share
  - M: dev environment - hard to build these. Long slog through virtual machine (e.g. fedora 2.1 was still in alpha)... lots of patience, and a new computer. Taking notes - wrote everything down, error messages etc so can do on next install, take to project maintainer as suggestions for things to go into instructions.
  - A: pymongo sometimes gets a bug that spirals out of control, and ends up being a python bug (that's already been reported)... e.g. multiprocessing bug that took time to figure out. Getting a copy of the project is a big step towards actually contributing.
  - J: like perfect storm types of bugs, e.g. json schema had a bug... likes semantic versioning, maintaining backwards compatibility... a release was broken and put out, got bug report 6 hours after release from people in big orgs (e.g. openstack, mediawiki)... tiny detail - pip environment markers - broke the release; lots of people; doesn't like fixing bugs until have regression test in place = pressure is on... did in 24 hours...
  - B: asynch in Tornado Async is iterator returning awaitable objects, python library asyncio had different interpretation- trying to mix them, got stack overflows endlessly trying to convert objects. Still an open issue- did a workaround, but other code will have similar problems with it.
  - D: rewrote hash table function in git, git merge started crashing... because of the fix... git index is also called cache and staging area, depending of which part of code you're in... created nightmares on macs... weirdass pointer being pulled out from under code whilst still in use - only happened on a mac on certain large merges... but patch not accepted the first way was written, so rewrote a different way
- Q: anything your OS projects want help with now?
  - M: has a bug list - look for Gnome Music getting started page. "Gnome Love Bugs" <https://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=gnome-music;dist=unstable>

- A: lot of mongodb driver things to work on... any time release, looking for people to test for bugs - finding one = starting a conversation. Drivers have various levels of accessible bugs... mongodb is too big a place to start in. And Mongodb is hiring.
- J: Twisted has tedious but beginner-friendly work; J has proof of concept projects that wrote parts he needed (e.g. docker python bindings are literal translations of command line commands - can jump in and extend out that library), etc. code lives on github <https://github.com/Julian...> not heavily organised.
- B: cockroachdb has well-organised bug list. <https://github.com/cockroachdb/cockroach> - can talk to B about stuff that's not well-organised.
- D: git doesn't have a public bug list, but can look at unit tests and see known failures... need to ask git if they're things that people care about. Also e.g. "git rm" removes entire account? (is not filed yet). (all panelists are hiring!)
- Audience questions:
  - Q: Dropbox might be a good starter project.
  - Q: Setting aside time to work on OS? A: motivated by other people - find someone interested in working on a project. Take advantage of frustration - immediately after frustration, try to work something out.
  - Q: How do you deal with ownership in companies based on OS? Ordinary employee = work for hire. Contract employee = 20-point test, but can override that in the contract. Ownership matters if you want to enforce the license - need copyrights to do this.
  - Q: licenses? Apache vs MIT vs GPL? Prefer for most things copyleft (e.g. GPL), otherwise adoption. More permissive, e.g. Apache, MIT. But use FSF-approved license, e.g. Apache, MIT or GPL.